

Importing Skin Cancer Data

read the data

▼ Importing all the important libraries

```
import pathlib
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
import PIL
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
```

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

→ Mounted at /content/gdrive

This assignment uses a dataset of about 2357 images of skin cancer types. The dataset contains 9 sub-directories in each train and test subdirectories. The 9 sub-directories contains the images of 9 skin cancer types respectively.

```
!unzip gdrive/MyDrive/CNN_assignment.zip
```

→ Archive: gdrive/MyDrive/CNN_assignment.zip
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0010512.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0010889.jpg

inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024468.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024470.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024511.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024646.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024654.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024707.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024763.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024771.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024800.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024913.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0024948.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0025368.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0025427.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/actinic keratosis/ISIC_0025605.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024331.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024332.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024345.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024360.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024403.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024411.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024431.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024432.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024436.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024443.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024448.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024452.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024454.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024457.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024461.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/basal cell carcinoma/ISIC_0024472.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0001114.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0001130.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0011410.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0011433.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0011478.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0011677.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0011865.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024318.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024330.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024386.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024396.jpg

```
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024553.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024845.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024973.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0024994.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/dermatofibroma/ISIC_0025154.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000002.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000004.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000013.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000022.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000026.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000029.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_0000030.jpg
```

```
# Defining the path for train and test images
data_dir_train = pathlib.Path("Skin cancer ISIC The International Skin Imaging Collaboration/Train")
data_dir_test = pathlib.Path('Skin cancer ISIC The International Skin Imaging Collaboration/Test')
```

```
!ls
```

```
→ gdrive sample_data 'Skin cancer ISIC The International Skin Imaging Collaboration'
```

```
image_count_train = len(list(data_dir_train.glob('*/*.jpg')))
print(image_count_train)
image_count_test = len(list(data_dir_test.glob('*/*.jpg')))
print(image_count_test)
```

```
→ 2239
118
```

Start coding or [generate](#) with AI.

load these images

▼ dataset creation

Define some parameters for the loader:

```
batch_size = 32
img_height = 180
img_width = 180
```

Using 80% of the images for training, and 20% for validation.

```
# train dataset here
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir_train,
    seed=123,
    validation_split = 0.2,
    subset = 'training',
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

→ Found 2239 files belonging to 9 classes.
Using 1792 files for training.

```
## Write your validation dataset here
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir_train,
    seed=123,
    validation_split = 0.2,
    subset = 'validation',
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

→ Found 2239 files belonging to 9 classes.
Using 447 files for validation.

```
# List of the classes of skin cancer and storage in a list.
```

```
class_names = train_ds.class_names
print(class_names)
```

```
→ ['actinic keratosis', 'basal cell carcinoma', 'dermatofibroma', 'melanoma', 'nevus', 'pigmented benign keratosis', 'seborrheic k
```

▼ Visualize the data

```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 10))
for images, labels in train_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
```



vascular lesion



nevus



pigmented benign keratosis



pigmented benign keratosis



melanoma



dermatofibroma



pigmented benign keratosis



actinic keratosis



basal cell carcinoma





The `image_batch` is a tensor of the shape `(32, 180, 180, 3)`. This is a batch of 32 images of shape `180x180x3`. The `label_batch` is a tensor of the shape `(32,)`, corresponds labels to the 32 images.

Double-click (or enter) to edit

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

▼ model

```
#codes
num_classes = 9

model = Sequential([
    layers.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

→ /usr/local/lib/python3.10/dist-packages/keras/src/layers/preprocessing/tf_data_layer.py:19: UserWarning: Do not pass an `input_`
super().__init__(**kwargs)

▼ Compile the model

Choose an appropriate optimiser and loss function for model training

```
### Todo, choose an appropriate optimiser and loss function
model.compile(optimizer='adam',
               loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
               metrics=['accuracy'])

# View the summary of all layers
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_1 (Conv2D)	(None, 90, 90, 32)	4,640
max_pooling2d_1 (MaxPooling2D)	(None, 45, 45, 32)	0
conv2d_2 (Conv2D)	(None, 45, 45, 64)	18,496
max_pooling2d_2 (MaxPooling2D)	(None, 22, 22, 64)	0
flatten (Flatten)	(None, 30976)	0
dense (Dense)	(None, 128)	3,965,056
dense_1 (Dense)	(None, 9)	1,161

Total params: 3,989,801 (15.22 MB)

Trainable params: 3,989,801 (15.22 MB)

Non-trainable params: 0 (0.00 MB)

▼ Train the model

```
epochs = 20
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)
```

Epoch 1/20

56/56  21s 84ms/step - accuracy: 0.2402 - loss: 2.0299 - val_accuracy: 0.3669 - val_loss: 1.6486

Epoch 2/20
56/56 2s 18ms/step - accuracy: 0.4308 - loss: 1.5825 - val_accuracy: 0.5078 - val_loss: 1.4444

Epoch 3/20
56/56 1s 19ms/step - accuracy: 0.4852 - loss: 1.4603 - val_accuracy: 0.3870 - val_loss: 1.6989

Epoch 4/20
56/56 1s 16ms/step - accuracy: 0.4843 - loss: 1.4721 - val_accuracy: 0.5526 - val_loss: 1.3125

Epoch 5/20
56/56 1s 16ms/step - accuracy: 0.5653 - loss: 1.2168 - val_accuracy: 0.5302 - val_loss: 1.3218

Epoch 6/20
56/56 1s 16ms/step - accuracy: 0.5685 - loss: 1.1639 - val_accuracy: 0.5459 - val_loss: 1.3298

Epoch 7/20
56/56 1s 16ms/step - accuracy: 0.6339 - loss: 1.0459 - val_accuracy: 0.5414 - val_loss: 1.3735

Epoch 8/20
56/56 1s 16ms/step - accuracy: 0.6565 - loss: 0.9600 - val_accuracy: 0.5145 - val_loss: 1.3799

Epoch 9/20
56/56 1s 16ms/step - accuracy: 0.6398 - loss: 0.9376 - val_accuracy: 0.5682 - val_loss: 1.3716

Epoch 10/20
56/56 1s 16ms/step - accuracy: 0.6976 - loss: 0.8402 - val_accuracy: 0.5660 - val_loss: 1.5361

Epoch 11/20
56/56 1s 16ms/step - accuracy: 0.7380 - loss: 0.7553 - val_accuracy: 0.5548 - val_loss: 1.4635

Epoch 12/20
56/56 1s 18ms/step - accuracy: 0.7763 - loss: 0.6053 - val_accuracy: 0.5682 - val_loss: 1.4689

Epoch 13/20
56/56 1s 18ms/step - accuracy: 0.7828 - loss: 0.6250 - val_accuracy: 0.5459 - val_loss: 1.9278

Epoch 14/20
56/56 1s 18ms/step - accuracy: 0.8132 - loss: 0.5384 - val_accuracy: 0.5436 - val_loss: 1.8804

Epoch 15/20
56/56 1s 16ms/step - accuracy: 0.8074 - loss: 0.4981 - val_accuracy: 0.5302 - val_loss: 1.9547

Epoch 16/20
56/56 1s 16ms/step - accuracy: 0.8620 - loss: 0.3811 - val_accuracy: 0.5481 - val_loss: 1.8196

Epoch 17/20
56/56 1s 16ms/step - accuracy: 0.8878 - loss: 0.3017 - val_accuracy: 0.5615 - val_loss: 1.8274

Epoch 18/20
56/56 1s 16ms/step - accuracy: 0.8924 - loss: 0.2933 - val_accuracy: 0.5369 - val_loss: 1.9698

Epoch 19/20
56/56 1s 16ms/step - accuracy: 0.8866 - loss: 0.3022 - val_accuracy: 0.5101 - val_loss: 2.2178

Epoch 20/20
56/56 1s 16ms/step - accuracy: 0.8946 - loss: 0.2663 - val_accuracy: 0.5503 - val_loss: 2.3706

✓ Visualizing training results

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

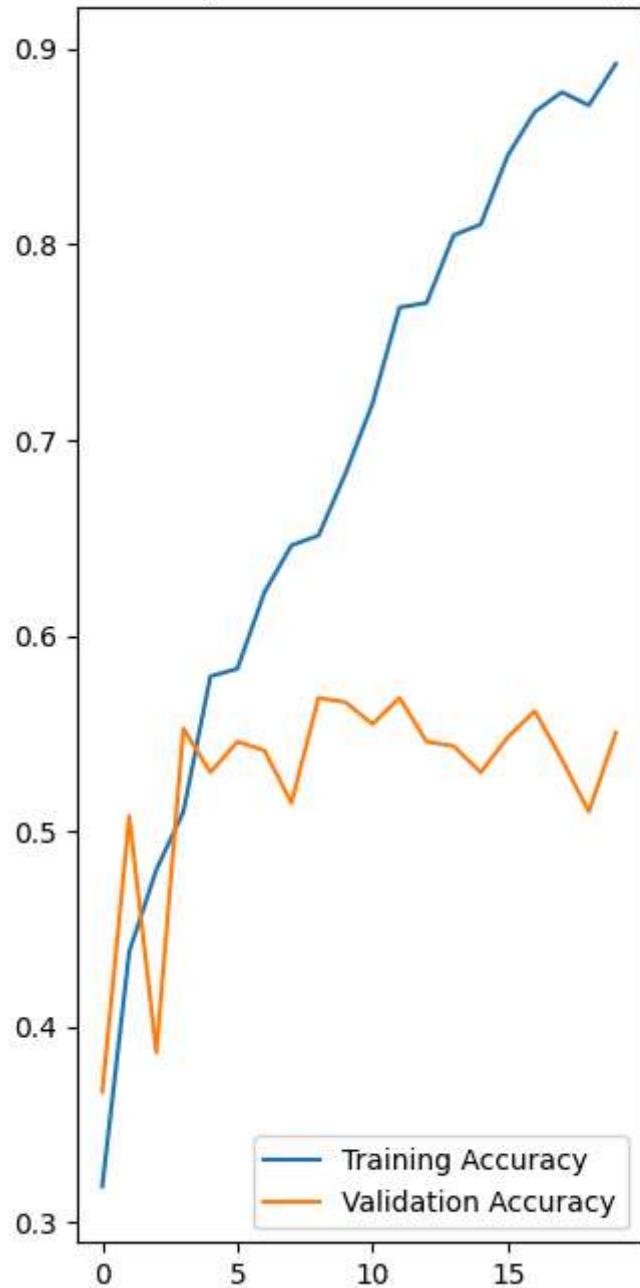
epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

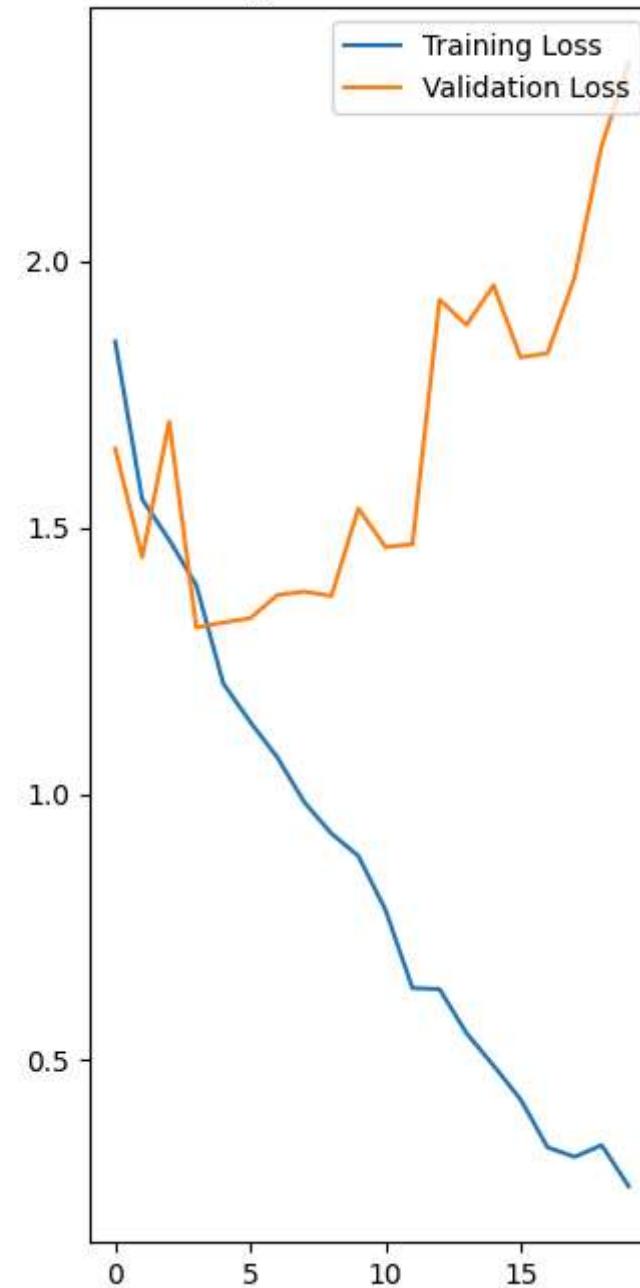
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Training and Validation Accuracy



Training and Validation Loss



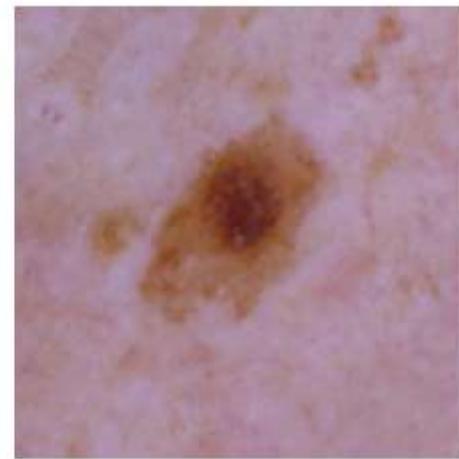
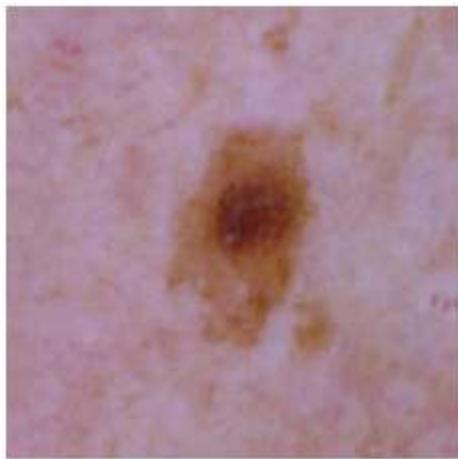
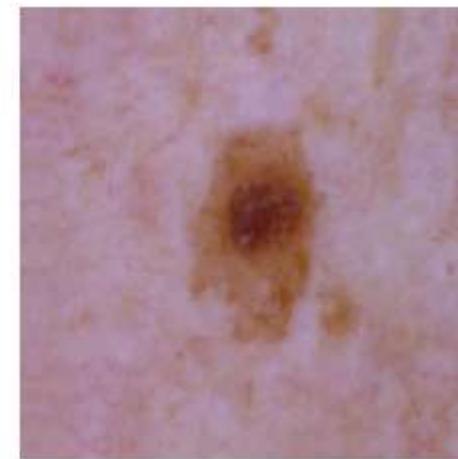
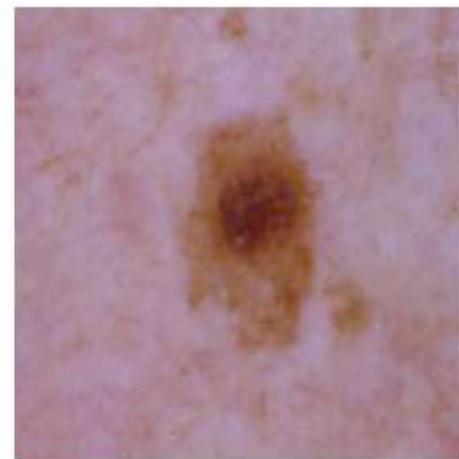
- ✓ training accuracy keeps on rising whereas validation accuracy is almost flat whereas training loss is on constant decline & validation loss is on rise

Overfitting: The increasing gap between training and validation loss, coupled with the high training accuracy and relatively low validation accuracy, is a clear sign of overfitting. The model is memorizing the training data rather than learning patterns that can generalize well.

Potential Issues: The sharp increase in validation loss towards the end suggests that the model might be excessively complex or that the training process might be too long for the dataset size or complexity

```
#data augmentation strategy.  
data_augmentation = keras.Sequential(  
[  
    layers.RandomFlip("horizontal",  
                      input_shape=(img_height,  
                                  img_width,  
                                  3)),  
    layers.RandomRotation(0.1),  
    layers.RandomZoom(0.1),  
])  
)
```

```
# visualize augmentation strategy works for one instance of training image.  
plt.figure(figsize=(10, 10))  
for images, _ in train_ds.take(1):  
    for i in range(9):  
        augmented_images = data_augmentation(images)  
        ax = plt.subplot(3, 3, i + 1)  
        plt.imshow(augmented_images[0].numpy().astype("uint8"))  
        plt.axis("off")
```





▼ model, compile and train the model

```
## Dropout layer if there is an evidence of overfitting in your findings

model = Sequential([
    data_augmentation,
    layers.Rescaling(1./255),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

▼ Compiling the model

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

▼ Training the model

```
## train model for 20 epochs
epochs = 20
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)
```

→ Epoch 1/20
56/56 6s 35ms/step - accuracy: 0.2246 - loss: 2.0795 - val_accuracy: 0.2371 - val_loss: 1.9084
Epoch 2/20
56/56 2s 32ms/step - accuracy: 0.3412 - loss: 1.7865 - val_accuracy: 0.4609 - val_loss: 1.5694
Epoch 3/20
56/56 2s 29ms/step - accuracy: 0.4429 - loss: 1.6013 - val_accuracy: 0.4966 - val_loss: 1.4914
Epoch 4/20
56/56 2s 29ms/step - accuracy: 0.4939 - loss: 1.4446 - val_accuracy: 0.4497 - val_loss: 1.5502
Epoch 5/20
56/56 2s 30ms/step - accuracy: 0.5083 - loss: 1.4470 - val_accuracy: 0.4228 - val_loss: 1.5906
Epoch 6/20
56/56 2s 29ms/step - accuracy: 0.4923 - loss: 1.4317 - val_accuracy: 0.5123 - val_loss: 1.4161
Epoch 7/20
56/56 2s 30ms/step - accuracy: 0.5107 - loss: 1.3676 - val_accuracy: 0.5168 - val_loss: 1.3464
Epoch 8/20
56/56 3s 32ms/step - accuracy: 0.5090 - loss: 1.3751 - val_accuracy: 0.5190 - val_loss: 1.4181
Epoch 9/20
56/56 2s 32ms/step - accuracy: 0.5331 - loss: 1.3018 - val_accuracy: 0.5235 - val_loss: 1.4301
Epoch 10/20
56/56 2s 30ms/step - accuracy: 0.5396 - loss: 1.2668 - val_accuracy: 0.5459 - val_loss: 1.3736
Epoch 11/20
56/56 3s 31ms/step - accuracy: 0.5698 - loss: 1.2348 - val_accuracy: 0.5481 - val_loss: 1.3027
Epoch 12/20
56/56 2s 33ms/step - accuracy: 0.5618 - loss: 1.2138 - val_accuracy: 0.5347 - val_loss: 1.3719
Epoch 13/20
56/56 2s 30ms/step - accuracy: 0.5940 - loss: 1.1462 - val_accuracy: 0.5638 - val_loss: 1.3326
Epoch 14/20
56/56 3s 31ms/step - accuracy: 0.5712 - loss: 1.1641 - val_accuracy: 0.5101 - val_loss: 1.3561

```
Epoch 15/20
56/56 2s 30ms/step - accuracy: 0.5780 - loss: 1.1416 - val_accuracy: 0.5570 - val_loss: 1.3990
Epoch 16/20
56/56 2s 29ms/step - accuracy: 0.5874 - loss: 1.1270 - val_accuracy: 0.4832 - val_loss: 1.5591
Epoch 17/20
56/56 2s 30ms/step - accuracy: 0.5961 - loss: 1.1586 - val_accuracy: 0.5705 - val_loss: 1.3189
Epoch 18/20
56/56 2s 29ms/step - accuracy: 0.6147 - loss: 1.1189 - val_accuracy: 0.5369 - val_loss: 1.3023
Epoch 19/20
56/56 3s 29ms/step - accuracy: 0.6058 - loss: 1.0784 - val_accuracy: 0.5570 - val_loss: 1.2753
Epoch 20/20
56/56 2s 30ms/step - accuracy: 0.6159 - loss: 1.0739 - val_accuracy: 0.5682 - val_loss: 1.3616
```

▼ Visualizing the results

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

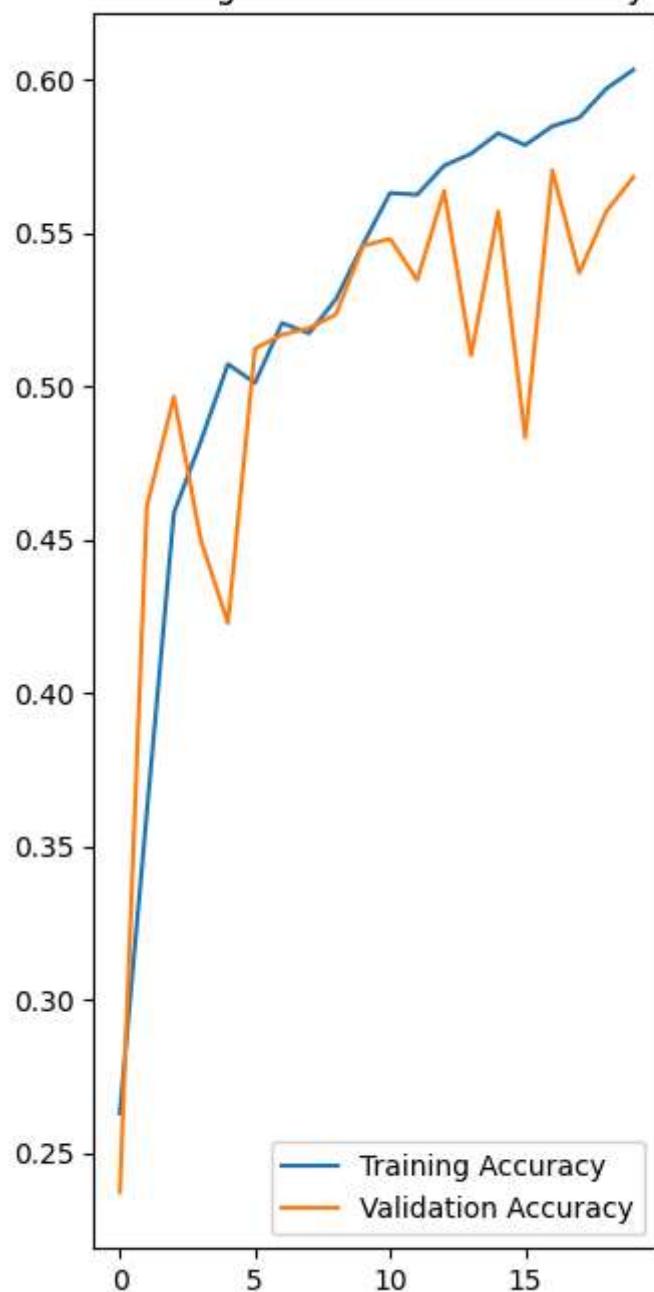
epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

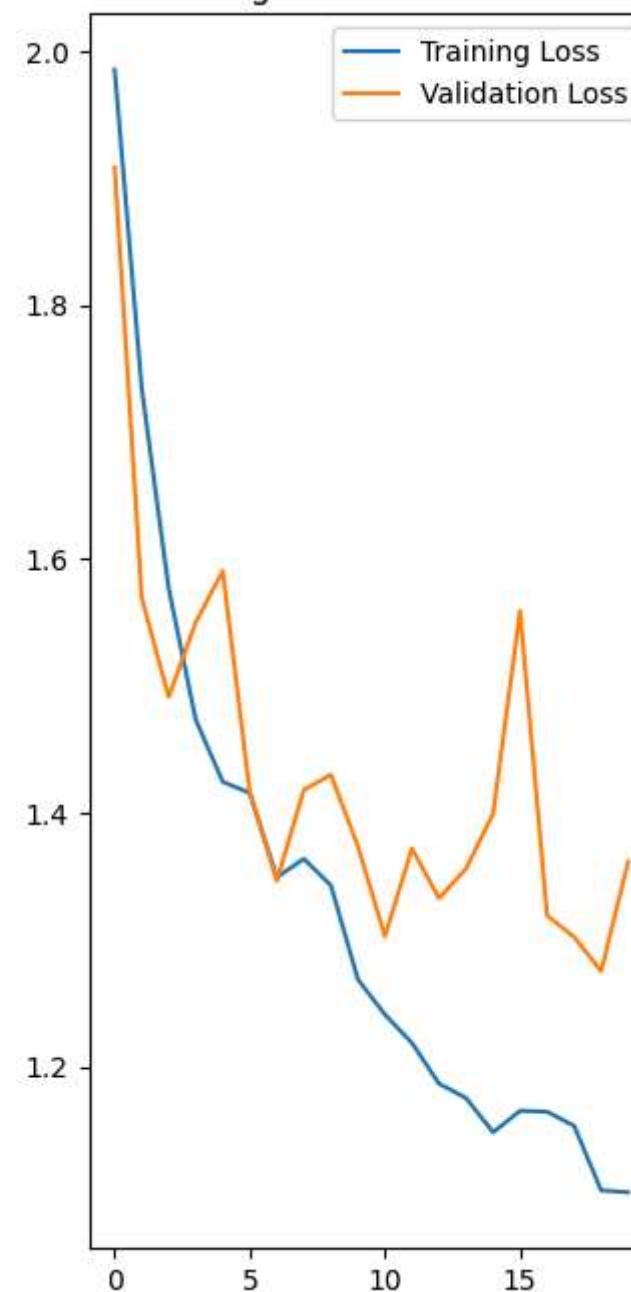
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Training and Validation Accuracy



Training and Validation Loss



Overfitting Signs: The fluctuations in validation accuracy and the occasional rise in validation loss while training loss continues to decrease suggest overfitting. The model is possibly learning noise or specific patterns in the training data that do not generalize well to unseen data.

Model Performance: The gap between training and validation performance is not extremely large, so the model is not severely overfitting, but the fluctuations indicate room for improvement, possibly through regularization techniques, early stopping, or cross-validation.

- ✓ **Todo:** Find the distribution of classes in the training dataset.

Context: Many times real life datasets can have class imbalance, one class can have proportionately higher number of samples compared to the others. Class imbalance can have a detrimental effect on the final model quality. Hence as a sanity check it becomes important to check what is the distribution of classes in the data.

```
## Your code goes here.  
from glob import glob  
path_list = [x for x in glob(os.path.join(data_dir_train, '*', '*.jpg'))]  
lesion_list = [os.path.basename(os.path.dirname(y)) for y in glob(os.path.join(data_dir_train, '*', '*.jpg'))]  
len(path_list)
```

2239

```
dataframe_dict_original = dict(zip(path_list, lesion_list))  
original_df = pd.DataFrame(list(dataframe_dict_original.items()), columns = ['Path', 'Label'])  
original_df
```



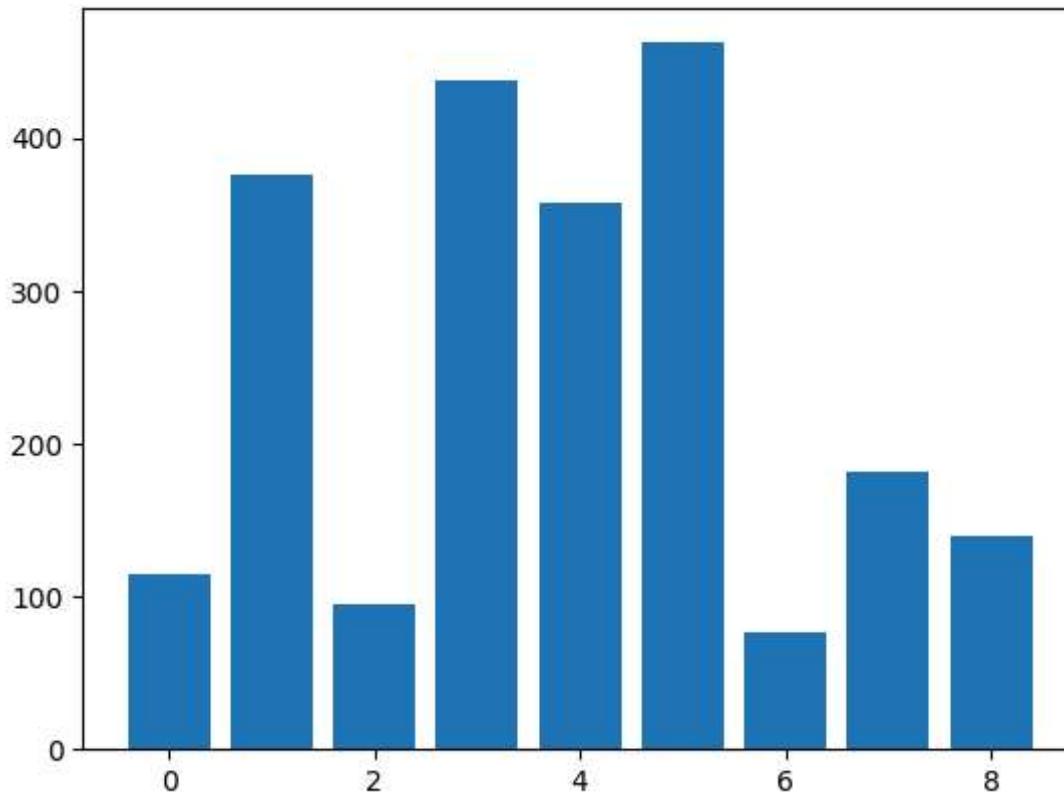
	Path	Label
--	------	-------

0	Skin cancer ISIC The International Skin Imagin...	basal cell carcinoma
1	Skin cancer ISIC The International Skin Imagin...	basal cell carcinoma
2	Skin cancer ISIC The International Skin Imagin...	basal cell carcinoma
3	Skin cancer ISIC The International Skin Imagin...	basal cell carcinoma
4	Skin cancer ISIC The International Skin Imagin...	basal cell carcinoma
...
2234	Skin cancer ISIC The International Skin Imagin...	melanoma
2235	Skin cancer ISIC The International Skin Imagin...	melanoma
2236	Skin cancer ISIC The International Skin Imagin...	melanoma
2237	Skin cancer ISIC The International Skin Imagin...	melanoma
2238	Skin cancer ISIC The International Skin Imagin...	melanoma

2239 rows × 2 columns

```
from sklearn.preprocessing import LabelEncoder
from collections import Counter
# split into input and output elements
X, y = original_df['Path'], original_df['Label']
# label encode the target variable
y = LabelEncoder().fit_transform(y)
# summarize distribution
counter = Counter(y)
for k,v in counter.items():
    per = v / len(y) * 100
    print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
plt.bar(counter.keys(), counter.values())
plt.show()
```

Class=1, n=376 (16.793%)
Class=8, n=139 (6.208%)
Class=4, n=357 (15.945%)
Class=2, n=95 (4.243%)
Class=5, n=462 (20.634%)
Class=6, n=77 (3.439%)
Class=0, n=114 (5.092%)
Class=7, n=181 (8.084%)
Class=3, n=438 (19.562%)



class has the least number of samples= class 6

classes dominate the data in terms proportionate number of samples=class 5

▼ Rectification of the class imbalance

```
!pip install Augmentor
```

```
→ Collecting Augmentor
  Downloading Augmentor-0.2.12-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: Pillow>=5.2.0 in /usr/local/lib/python3.10/dist-packages (from Augmentor) (9.4.0)
Requirement already satisfied: tqdm>=4.9.0 in /usr/local/lib/python3.10/dist-packages (from Augmentor) (4.66.5)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from Augmentor) (1.26.4)
  Downloading Augmentor-0.2.12-py2.py3-none-any.whl (38 kB)
Installing collected packages: Augmentor
Successfully installed Augmentor-0.2.12
```

To use Augmentor, the following general procedure is followed:

1. Instantiate a `Pipeline` object pointing to a directory containing your initial image data set.
2. Define a number of operations to perform on this data set using your `Pipeline` object.
3. Execute these operations by calling the `Pipeline's` `sample()` method.

```
path_to_training_dataset="Skin cancer ISIC The International Skin Imaging Collaboration/Train/"
import Augmentor
for i in class_names:
    p = Augmentor.Pipeline(path_to_training_dataset + i)
    p.rotate(probability=0.7, max_left_rotation=10, max_right_rotation=10)
    p.sample(500) ## We are adding 500 samples per class to make sure that none of the classes are sparse.
```

```
→ Initialised with 114 image(s) found.
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output.Processing
Initialised with 376 image(s) found.
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output.Process:
Initialised with 95 image(s) found.
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/dermatofibroma/output.Processing <P:
Initialised with 438 image(s) found.
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/melanoma/output.Processing <PIL.Imag
```

```
Initialised with 357 image(s) found.  
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/nevus/output.Processing <PIL.Image.:  
Initialised with 462 image(s) found.  
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/pigmented benign keratosis/output.Pi  
Initialised with 77 image(s) found.  
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/seborrheic keratosis/output.Process:  
Initialised with 181 image(s) found.  
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/squamous cell carcinoma/output.Proc  
Initialised with 139 image(s) found.  
Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/vascular lesion/output.Processing <P
```

Augmentor has stored the augmented images in the output sub-directory of each of the sub-directories of skin cancer types.. Lets take a look at total count of augmented images.

```
image_count_train = len(list(data_dir_train.glob('*/*output/*.jpg')))  
print(image_count_train)
```

→ 4500

▼ Lets see the distribution of augmented data after adding new images to the original training data.

```
path_list_new = [x for x in glob(os.path.join(data_dir_train, '*', 'output', '*.jpg'))]  
path_list_new
```

→ ['Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0030261.jpg_db863d1a-645e-4d61-8a1f-3bb4e4946b19.jpg', 'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0025031.jpg_54ee65c5-4bc8-4582-9991-5fd8a9374e64.jpg', 'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0031062.jpg_269be014-8fa5-491b-bc5e-cb780efb3304.jpg', 'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0031095.jpg_8c883bb3-ca51-405e-b23f-77951c2fe8d8.jpg', 'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0026350.jpg_da80169a-2e81-4cb7-8d4a-12e918b1ec98.jpg',

```
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0030915.jpg_f52f91d2-89cf-42a8-94c3-f6b558545206.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0030594.jpg_e4c6fb74-00d2-4c08-9343-556b3988dc7e.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0027531.jpg_9f2a1095-fd88-46f0-b40b-20ecc863e92b.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0031489.jpg_86d239c0-3385-4e03-ab1a-a5a283e37253.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0030767.jpg_e3e73e18-e061-4a9d-9d0a-b8a873a33c9a.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0027722.jpg_b5fae678-86bf-4b56-8a36-3893be3bb474.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0024743.jpg_b525fddc-729c-422b-a6b9-fa9dcbba5043.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0029680.jpg_18dc7d4e-29c1-4803-9096-aa9e1e448d29.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0026845.jpg_bbd32934-f039-4cc0-9764-0de21aada0d3.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0026091.jpg_243d8332-15ab-4579-ae2e-c1ae7ebdc027.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0028677.jpg_1b01b043-a42b-4753-a415-264124058824.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0029341.jpg_bce169ad-e78a-4e6c-b566-1252689c4529.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0025700.jpg_cf3b5474-f2f2-4838-9742-8df0eb728134.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0027544.jpg_15cd65d0-d064-49ec-a942-da079244060a.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0028719.jpg_9cd592bd-7444-4a6a-9ace-4afc99f9d9a3.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0029020.jpg_51053043-2a00-491a-a4dd-2cf3e5e7ef75.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0028928.jpg_be21a4cf-26d6-4054-bddd-23c1cff2c650.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0029899.jpg_6edaf01c-25ef-4d90-a941-6a894d2d1b7d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0029372.jpg_c390ec99-8ed6-41ac-a9b9-482342e67753.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell carcinoma_original_ISIC_0030915.jpg_978ae9f7-4af6-439f-b23f-2aeaf0b3e2c0.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell
```

```
carcinoma_original_ISIC_0027058.jpg_b311bf97-84fe-4f4e-8391-e41db99e0179.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell
carcinoma_original_ISIC_0029193.jpg_41040100-df4f-4eed-a17a-6235ee962734.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell
carcinoma_original_ISIC_0031513.jpg_6bb1bbbb3-3b67-42b9-93e1-f61a9ed4642d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output/basal cell
```

```
lesion_list_new = [os.path.basename(os.path.dirname(os.path.dirname(y))) for y in glob(os.path.join(data_dir_train, '*', 'output', '*'))]
lesion_list_new
```

```
dataframe_dict_new = dict(zip(path_list_new, lesion_list_new))
```

```
df2 = pd.DataFrame(list(dataframe_dict_new.items()), columns = ['Path', 'Label'])
```

```
new_df = pd.concat([original_df, df2], ignore_index=True)
```

```
new_df['Label'].value_counts()
```

```
→ pigmented benign keratosis    962
    melanoma                    938
    basal cell carcinoma        876
    nevus                       857
    squamous cell carcinoma    681
    vascular lesion              639
    actinic keratosis            614
    dermatofibroma                595
    seborrheic keratosis          577
Name: Label, dtype: int64
```

now we have added 500 images to all the classes to maintain some class balance. and can be added more to improve training process.

▼ model on the data created using Augmentor

```
batch_size = 32
img_height = 180
img_width = 180
```

▼ Create a training dataset

```
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir_train,
    seed=123,
    validation_split = 0.2,
    subset = 'training', ## Todo choose the correct parameter value, so that only training data is referred to
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

```
→ Found 6739 files belonging to 9 classes.
Using 5392 files for training.
```

▼ Todo: Create a validation dataset

```
val_ds = tf.keras.preprocessing.image_dataset_from_directory(  
    data_dir_train,  
    seed=123,  
    validation_split = 0.2,  
    subset = 'validation',## Todo choose the correct parameter value, so that only validation data is referred to,  
    image_size=(img_height, img_width),  
    batch_size=batch_size)
```

→ Found 6739 files belonging to 9 classes.
Using 1347 files for validation.

▼ Creation model

```
AUTOTUNE = tf.data.AUTOTUNE  
  
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)  
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)  
  
model = Sequential([  
    layers.Rescaling(1./255),  
    layers.Conv2D(16, 3, padding='same', activation='relu'),  
    layers.MaxPooling2D(),  
    layers.Conv2D(32, 3, padding='same', activation='relu'),  
    layers.MaxPooling2D(),  
    layers.Conv2D(64, 3, padding='same', activation='relu'),  
    layers.MaxPooling2D(),  
    layers.Dropout(0.2),  
    layers.Flatten(),  
    layers.Dense(128, activation='relu'),  
    layers.Dense(num_classes)  
])
```

▼ *Compiling model

```
## your code goes here

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

▼ Train your model

```
epochs = 20
history = model.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)
```

```
→ Epoch 1/20
169/169 36s 83ms/step - accuracy: 0.1936 - loss: 2.3524 - val_accuracy: 0.4350 - val_loss: 1.4697
Epoch 2/20
169/169 10s 17ms/step - accuracy: 0.4310 - loss: 1.5373 - val_accuracy: 0.5323 - val_loss: 1.2371
Epoch 3/20
169/169 6s 23ms/step - accuracy: 0.5542 - loss: 1.2088 - val_accuracy: 0.5679 - val_loss: 1.1562
Epoch 4/20
169/169 4s 21ms/step - accuracy: 0.6216 - loss: 1.0434 - val_accuracy: 0.5939 - val_loss: 1.0769
Epoch 5/20
169/169 3s 17ms/step - accuracy: 0.6876 - loss: 0.8894 - val_accuracy: 0.6696 - val_loss: 0.9257
Epoch 6/20
169/169 5s 17ms/step - accuracy: 0.7410 - loss: 0.7205 - val_accuracy: 0.6303 - val_loss: 0.9707
Epoch 7/20
169/169 5s 17ms/step - accuracy: 0.7880 - loss: 0.5887 - val_accuracy: 0.7342 - val_loss: 0.7731
Epoch 8/20
169/169 5s 17ms/step - accuracy: 0.8242 - loss: 0.4977 - val_accuracy: 0.7416 - val_loss: 0.7713
Epoch 9/20
169/169 5s 18ms/step - accuracy: 0.8455 - loss: 0.4343 - val_accuracy: 0.7639 - val_loss: 0.6472
```

```
Epoch 10/20  
169/169 5s 18ms/step - accuracy: 0.8863 - loss: 0.3338 - val_accuracy: 0.8010 - val_loss: 0.6769  
Epoch 11/20  
169/169 3s 17ms/step - accuracy: 0.9014 - loss: 0.2918 - val_accuracy: 0.7929 - val_loss: 0.6333  
Epoch 12/20  
169/169 3s 18ms/step - accuracy: 0.9166 - loss: 0.2335 - val_accuracy: 0.8018 - val_loss: 0.6895  
Epoch 13/20  
169/169 3s 18ms/step - accuracy: 0.9176 - loss: 0.2319 - val_accuracy: 0.8181 - val_loss: 0.6359  
Epoch 14/20  
169/169 5s 18ms/step - accuracy: 0.9263 - loss: 0.2011 - val_accuracy: 0.8159 - val_loss: 0.6004  
Epoch 15/20  
169/169 5s 18ms/step - accuracy: 0.9356 - loss: 0.1804 - val_accuracy: 0.8226 - val_loss: 0.6350  
Epoch 16/20  
169/169 3s 18ms/step - accuracy: 0.9403 - loss: 0.1572 - val_accuracy: 0.8137 - val_loss: 0.6267  
Epoch 17/20  
169/169 3s 17ms/step - accuracy: 0.9380 - loss: 0.1688 - val_accuracy: 0.8107 - val_loss: 0.6615  
Epoch 18/20  
169/169 3s 17ms/step - accuracy: 0.9400 - loss: 0.1478 - val_accuracy: 0.8203 - val_loss: 0.6652  
Epoch 19/20  
169/169 3s 17ms/step - accuracy: 0.9509 - loss: 0.1308 - val_accuracy: 0.7892 - val_loss: 0.9013  
Epoch 20/20  
169/169 5s 17ms/step - accuracy: 0.9508 - loss: 0.1210 - val_accuracy: 0.8322 - val_loss: 0.7136
```

▼ Visualize the model results

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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

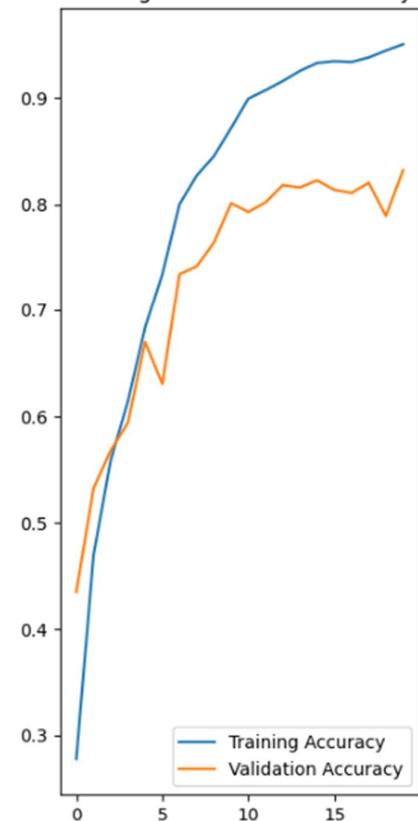
epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

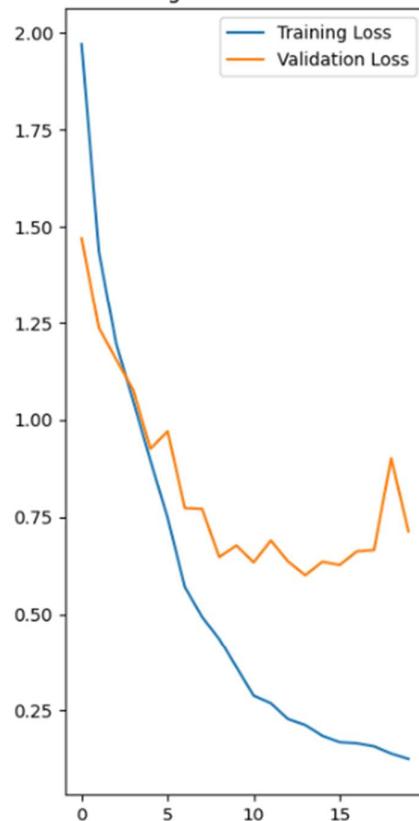
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



Training and Validation Accuracy



Training and Validation Loss



Overfitting Reduced: Compared to the previous graphs, the overfitting issue seems to have been mitigated to some extent,

- ✓ as indicated by the closer alignment of validation and training curves. However, the significant gap between the two still suggests that overfitting may be present.

Improved Generalization: The improvement in validation accuracy and the more stable validation loss are positive signs.

[] Start coding or generate with AI.