

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_000002.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000004.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000013.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000022.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000026.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000029.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000030.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000031.jpg
inflating: Skin cancer ISIC The International Skin Imaging Collaboration/Test/melanoma/ISIC_000035.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 keratosis', 'squamous cell carcinoma', 'vascular lesion']

▼ 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_shape`/`input_dim` argument to a layer. When using Sequence

▼ 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)

Train the model

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

Epoch 1/20
56/56 28s 204ms/step - accuracy: 0.2038 - loss: 2.2001 - val_accuracy: 0.3937 - val_loss: 1.7866
Epoch 2/20
56/56 1s 17ms/step - accuracy: 0.3731 - loss: 1.7054 - val_accuracy: 0.4676 - val_loss: 1.5504
Epoch 3/20
56/56 1s 16ms/step - accuracy: 0.4423 - loss: 1.5423 - val_accuracy: 0.4362 - val_loss: 1.5823
Epoch 4/20
56/56 1s 16ms/step - accuracy: 0.4809 - loss: 1.4625 - val_accuracy: 0.4497 - val_loss: 1.5377
Epoch 5/20
56/56 1s 16ms/step - accuracy: 0.5384 - loss: 1.2975 - val_accuracy: 0.5257 - val_loss: 1.3213
Epoch 6/20
56/56 1s 18ms/step - accuracy: 0.5717 - loss: 1.2008 - val_accuracy: 0.5391 - val_loss: 1.4066
Epoch 7/20
56/56 1s 18ms/step - accuracy: 0.6227 - loss: 1.0704 - val_accuracy: 0.5347 - val_loss: 1.3467
Epoch 8/20
56/56 1s 17ms/step - accuracy: 0.6523 - loss: 1.0287 - val_accuracy: 0.5123 - val_loss: 1.3984
Epoch 9/20
56/56 1s 16ms/step - accuracy: 0.6837 - loss: 0.8876 - val_accuracy: 0.5414 - val_loss: 1.3686
Epoch 10/20
56/56 1s 16ms/step - accuracy: 0.6738 - loss: 0.8279 - val_accuracy: 0.5391 - val_loss: 1.4763
Epoch 11/20
56/56 1s 16ms/step - accuracy: 0.7424 - loss: 0.7427 - val_accuracy: 0.5123 - val_loss: 1.5707
Epoch 12/20
56/56 1s 16ms/step - accuracy: 0.7273 - loss: 0.7427 - val_accuracy: 0.5481 - val_loss: 1.6262
```

```
Epoch 13/20
56/56 1s 16ms/step - accuracy: 0.8134 - loss: 0.5611 - val_accuracy: 0.5369 - val_loss: 1.7353
Epoch 14/20
56/56 1s 16ms/step - accuracy: 0.8200 - loss: 0.5076 - val_accuracy: 0.5302 - val_loss: 1.8014
Epoch 15/20
56/56 1s 16ms/step - accuracy: 0.8670 - loss: 0.4217 - val_accuracy: 0.5257 - val_loss: 1.7949
Epoch 16/20
56/56 1s 16ms/step - accuracy: 0.8553 - loss: 0.3966 - val_accuracy: 0.5213 - val_loss: 1.7783
Epoch 17/20
56/56 1s 16ms/step - accuracy: 0.8623 - loss: 0.3941 - val_accuracy: 0.5101 - val_loss: 1.9450
Epoch 18/20
56/56 1s 16ms/step - accuracy: 0.8995 - loss: 0.2732 - val_accuracy: 0.5414 - val_loss: 1.9770
Epoch 19/20
56/56 1s 19ms/step - accuracy: 0.9018 - loss: 0.2414 - val_accuracy: 0.5101 - val_loss: 2.3880
Epoch 20/20
56/56 1s 19ms/step - accuracy: 0.8906 - loss: 0.2518 - val_accuracy: 0.5302 - val_loss: 2.1048
```

▼ Visualizing training results

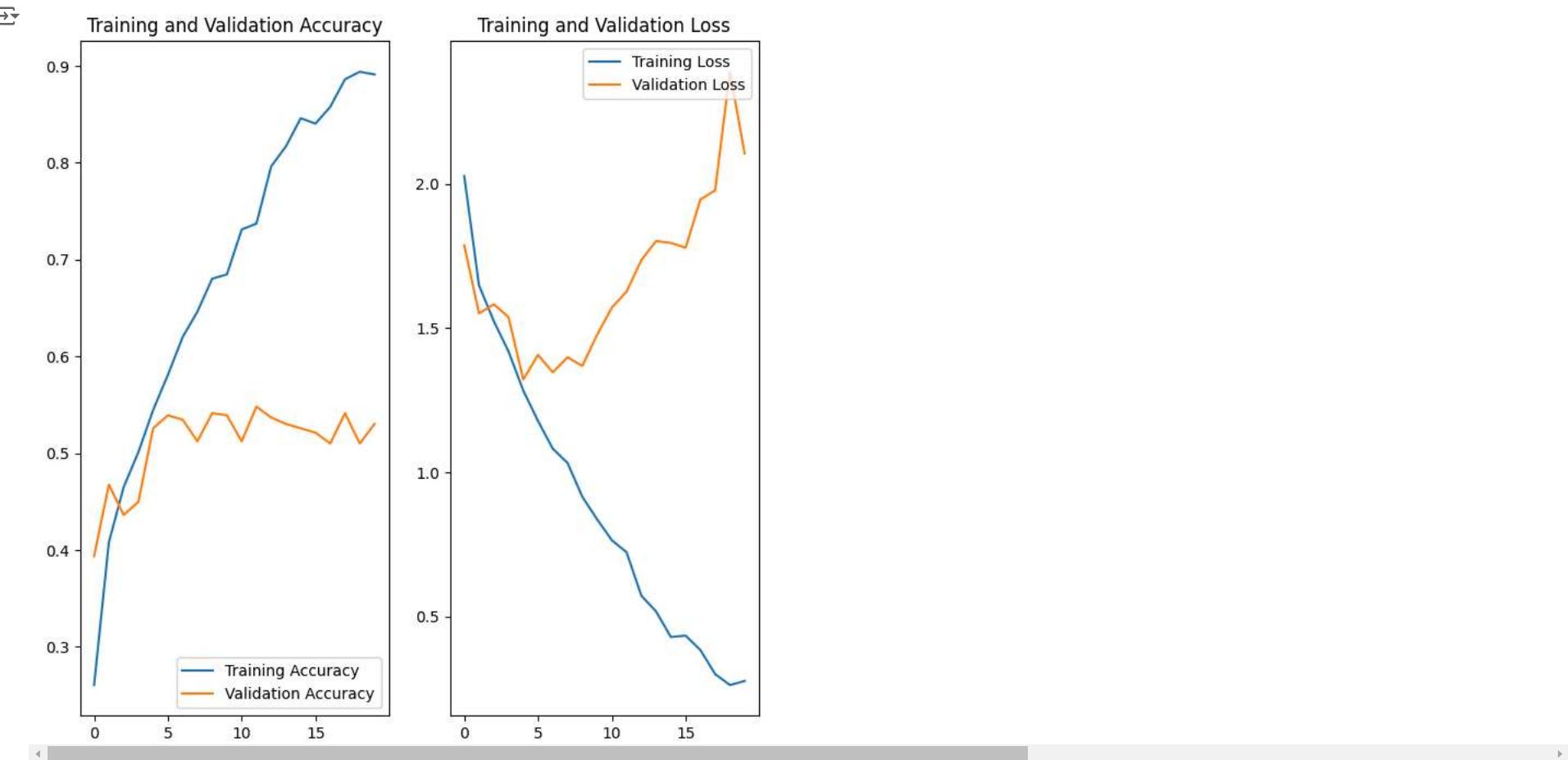
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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()
```



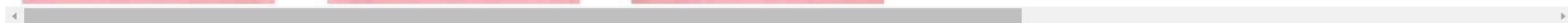
training accuracy keeps on rising whereas validation accuracy is almost flat whereas training loss is on constant decline & validation loss is on rise

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

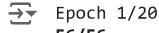
model = Sequential([
    data_augmentation,
    layers.Rescaling(1./255),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    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 34ms/step - accuracy: 0.2272 - loss: 2.1496 - val_accuracy: 0.3848 - val_loss: 1.7569  

Epoch 2/20  

56/56 ━━━━━━━━━━ 2s 30ms/step - accuracy: 0.4000 - loss: 1.6980 - val_accuracy: 0.4541 - val_loss: 1.5673  

Epoch 3/20  

56/56 ━━━━━━━━━━ 3s 30ms/step - accuracy: 0.4522 - loss: 1.5452 - val_accuracy: 0.4787 - val_loss: 1.4826  

Epoch 4/20  

56/56 ━━━━━━━━━━ 3s 30ms/step - accuracy: 0.4690 - loss: 1.4923 - val_accuracy: 0.5190 - val_loss: 1.3842  

Epoch 5/20  

56/56 ━━━━━━━━━━ 3s 30ms/step - accuracy: 0.4944 - loss: 1.4224 - val_accuracy: 0.5190 - val_loss: 1.3961  

Epoch 6/20  

56/56 ━━━━━━━━━━ 2s 30ms/step - accuracy: 0.5134 - loss: 1.3768 - val_accuracy: 0.5145 - val_loss: 1.4052  

Epoch 7/20  

56/56 ━━━━━━━━━━ 2s 29ms/step - accuracy: 0.5555 - loss: 1.3078 - val_accuracy: 0.5436 - val_loss: 1.3687  

Epoch 8/20  

56/56 ━━━━━━━━━━ 3s 32ms/step - accuracy: 0.5621 - loss: 1.2422 - val_accuracy: 0.5503 - val_loss: 1.3324  

Epoch 9/20  

56/56 ━━━━━━━━━━ 2s 30ms/step - accuracy: 0.5626 - loss: 1.2485 - val_accuracy: 0.5705 - val_loss: 1.3201  

Epoch 10/20
```

```
56/56 ━━━━━━━━━━ 2s 29ms/step - accuracy: 0.5680 - loss: 1.2077 - val_accuracy: 0.5660 - val_loss: 1.3430
Epoch 11/20
56/56 ━━━━━━━━ 2s 30ms/step - accuracy: 0.5462 - loss: 1.1939 - val_accuracy: 0.5526 - val_loss: 1.3227
Epoch 12/20
56/56 ━━━━ 2s 29ms/step - accuracy: 0.5914 - loss: 1.1528 - val_accuracy: 0.5749 - val_loss: 1.3606
Epoch 13/20
56/56 ━━━━ 3s 29ms/step - accuracy: 0.6125 - loss: 1.0842 - val_accuracy: 0.5817 - val_loss: 1.2650
Epoch 14/20
56/56 ━━━━ 3s 31ms/step - accuracy: 0.5792 - loss: 1.1692 - val_accuracy: 0.5548 - val_loss: 1.3218
Epoch 15/20
56/56 ━━━━ 2s 30ms/step - accuracy: 0.5958 - loss: 1.1455 - val_accuracy: 0.5861 - val_loss: 1.2455
Epoch 16/20
56/56 ━━━━ 2s 30ms/step - accuracy: 0.5968 - loss: 1.1037 - val_accuracy: 0.5615 - val_loss: 1.2739
Epoch 17/20
56/56 ━━━━ 2s 30ms/step - accuracy: 0.6174 - loss: 1.0566 - val_accuracy: 0.5593 - val_loss: 1.2964
Epoch 18/20
56/56 ━━━━ 3s 30ms/step - accuracy: 0.6157 - loss: 1.0433 - val_accuracy: 0.5705 - val_loss: 1.2340
Epoch 19/20
56/56 ━━━━ 2s 30ms/step - accuracy: 0.6397 - loss: 0.9752 - val_accuracy: 0.5548 - val_loss: 1.4017
Epoch 20/20
56/56 ━━━━ 3s 30ms/step - accuracy: 0.6135 - loss: 1.0390 - val_accuracy: 0.5794 - val_loss: 1.3172
```

▼ Visualizing the results

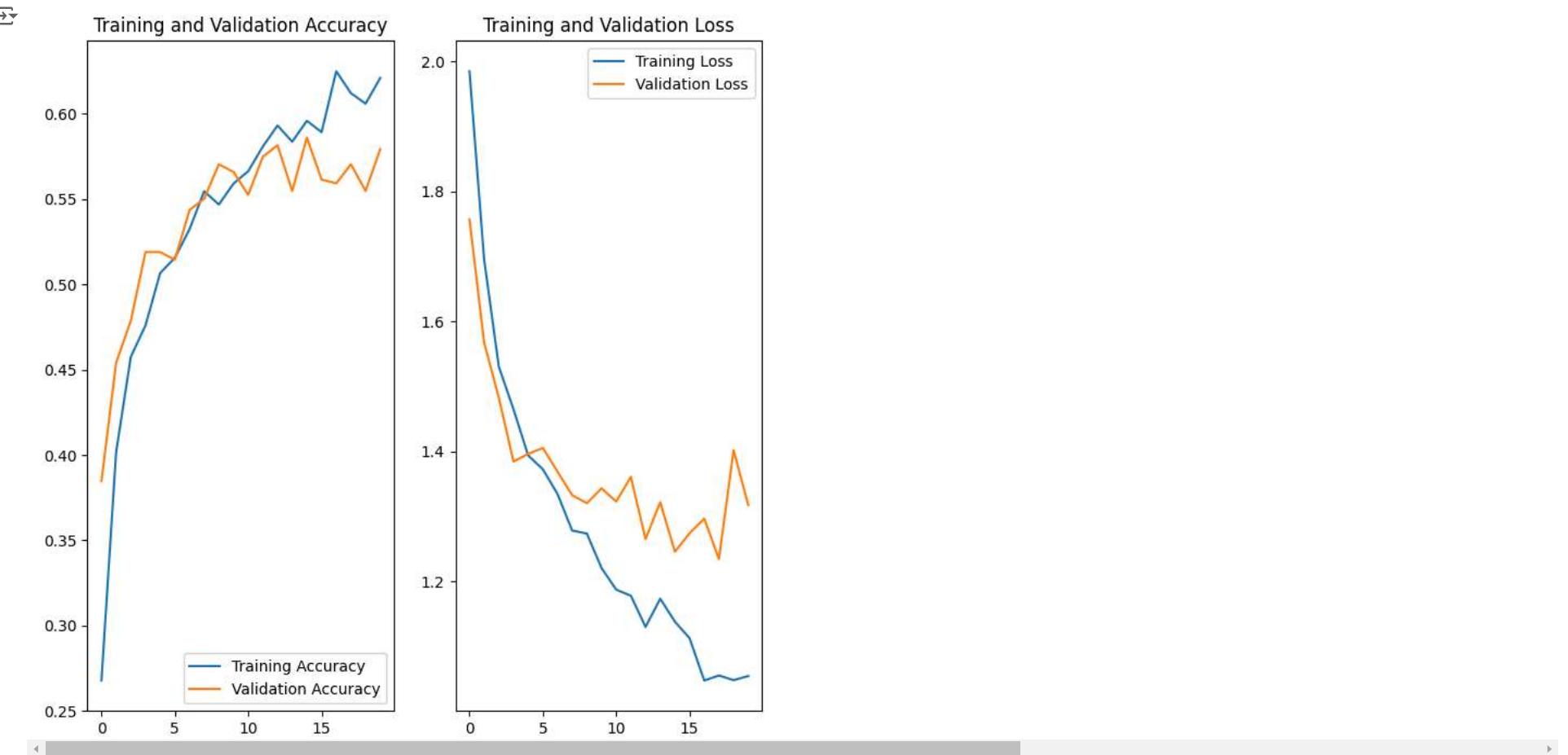
```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

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()
```



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
```

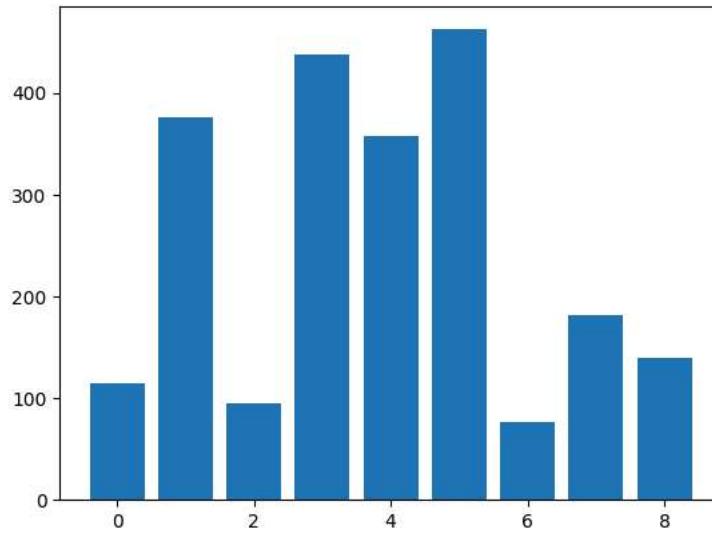
	Path	Label	grid icon
0	Skin cancer ISIC The International Skin Imagin...	nevus	grid icon
1	Skin cancer ISIC The International Skin Imagin...	nevus	grid icon
2	Skin cancer ISIC The International Skin Imagin...	nevus	grid icon
3	Skin cancer ISIC The International Skin Imagin...	nevus	grid icon
4	Skin cancer ISIC The International Skin Imagin...	nevus	grid icon
...	grid icon
2234	Skin cancer ISIC The International Skin Imagin...	vascular lesion	grid icon
2235	Skin cancer ISIC The International Skin Imagin...	vascular lesion	grid icon
2236	Skin cancer ISIC The International Skin Imagin...	vascular lesion	grid icon
2237	Skin cancer ISIC The International Skin Imagin...	vascular lesion	grid icon
2238	Skin cancer ISIC The International Skin Imagin...	vascular lesion	grid icon

2239 rows × 2 columns

Next steps: [Generate code with original_df](#) [View recommended plots](#) [New interactive sheet](#)

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



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 <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=600x450 at 0x781BE47:
 Initialised with 376 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/basal cell carcinoma/output. Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x781BE47:
 Initialised with 95 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/dermatofibroma/output. Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x781BE47:
 Initialised with 438 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/melanoma/output. Processing <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=3072x2304 at 0x781BE47:
 Initialised with 357 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/nevus/output. Processing <PIL.Image.Image image mode=RGB size=2048x1536 at 0x781BE4714730:
 Initialised with 462 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/pigmented benign keratosis/output. Processing <PIL.JpegImagePlugin.JpegImageFile image mode=RGB size=600x450 at 0x781BE47:
 Initialised with 77 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/seborrheic keratosis/output. Processing <PIL.Image.Image image mode=RGB size=1024x768 at 0x781BE47:
 Initialised with 181 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/squamous cell carcinoma/output. Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x781BE47:
 Initialised with 139 image(s) found.
 Output directory set to Skin cancer ISIC The International Skin Imaging Collaboration/Train/vascular lesion/output. Processing <PIL.Image.Image image mode=RGB size=600x450 at 0x781BE47:

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
```

→

```
0f73d2b3374c.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0028558.jpg_7ad1e117-52c7-4d98-9f92-49d13f2d35ec.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026171.jpg_c2def3c5-42aa-4db5-87e9-dd072a3598c7.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0030142.jpg_1cd00337-7618-453f-8d5f-fb4077eef5da.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0032404.jpg_829237c2-a8b6-47eb-837f-99cdb623ecae.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0029133.jpg_08c5c8bf-0b1b-46d0-a537-0ff5e8853d94.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026626.jpg_ba27419c-bd7d-45ca-aea0-026b9a337f3f.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026729.jpg_8592e6f0-ca49-4812-9394-c9e59a46ded.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0028314.jpg_36aa1d5f-4219-40d9-9d42-07c9b2979048.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0029460.jpg_e40621d0-d21c-4d65-8027-4fbfb1181f5d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026984.jpg_fe84a942-d8bf-4039-99ea-6ca5bbb9f377.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0031292.jpg_494d3ffc-d0a9-4002-a4da-f41b533a9a57.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0029930.jpg_d421c943-2c57-42fa-862f-aea998e99921.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026857.jpg_b38f76b1-00c9-4095-b593-175ab1f8572d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0030586.jpg_83807d60-4801-400c-a73a-49f83aaeb64d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0032404.jpg_1a5094c2-6be2-4394-90b1-09fd19c59492.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0032206.jpg_d07ac203-4303-40db-831a-caa84f96a61d.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0031993.jpg_fe25f7c3-6c4f-443d-baac-96bfe1ca3268.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0027650.jpg_cacd899a-ce28-4b10-beff-bad8af81f6c8.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0027829.jpg_bf6bd89e-dacc-4a55-b1b2-97cfe1634d3a.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0026709.jpg_ea85e664-e18d-468a-91dc-5c265ebf2ce5.jpg',
'Skin cancer ISIC The International Skin Imaging Collaboration/Train/actinic keratosis/output/actinic keratosis_original_ISIC_0029781.jpg_8e6ac4b9-218c-4a7e-9ebe-e1a18d14db7e.jpg',
```

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



```
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()
```



Label	count
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

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 ━━━━━━━━━━━━━━━━ 48s 83ms/step - accuracy: 0.1827 - loss: 2.2277 - val_accuracy: 0.3957 - val_loss: 1.6598
Epoch 2/20
```

```
. 169/169 ━━━━━━━━━━ 3s 18ms/step - accuracy: 0.4344 - loss: 1.5475 - val_accuracy: 0.5108 - val_loss: 1.3378
Epoch 3/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.5334 - loss: 1.2788 - val_accuracy: 0.5924 - val_loss: 1.1641
Epoch 4/20
169/169 ━━━━━━━━━━ 3s 18ms/step - accuracy: 0.6193 - loss: 1.0736 - val_accuracy: 0.5872 - val_loss: 1.0671
Epoch 5/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.6674 - loss: 0.9412 - val_accuracy: 0.6459 - val_loss: 1.0006
Epoch 6/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.7070 - loss: 0.8130 - val_accuracy: 0.6548 - val_loss: 0.9766
Epoch 7/20
169/169 ━━━━━━━━━━ 3s 18ms/step - accuracy: 0.7685 - loss: 0.6661 - val_accuracy: 0.6615 - val_loss: 0.9658
Epoch 8/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.7933 - loss: 0.5865 - val_accuracy: 0.6852 - val_loss: 0.9263
Epoch 9/20
169/169 ━━━━━━━━━━ 5s 18ms/step - accuracy: 0.8316 - loss: 0.5068 - val_accuracy: 0.7290 - val_loss: 0.7945
Epoch 10/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.8604 - loss: 0.3963 - val_accuracy: 0.7372 - val_loss: 0.7888
Epoch 11/20
169/169 ━━━━━━━━━━ 5s 18ms/step - accuracy: 0.8785 - loss: 0.3513 - val_accuracy: 0.7691 - val_loss: 0.7025
Epoch 12/20
169/169 ━━━━━━━━━━ 3s 20ms/step - accuracy: 0.8856 - loss: 0.3226 - val_accuracy: 0.7958 - val_loss: 0.6441
Epoch 13/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9119 - loss: 0.2429 - val_accuracy: 0.7639 - val_loss: 0.8100
Epoch 14/20
169/169 ━━━━━━━━━━ 3s 18ms/step - accuracy: 0.9043 - loss: 0.2492 - val_accuracy: 0.7944 - val_loss: 0.6460
Epoch 15/20
169/169 ━━━━━━━━━━ 5s 17ms/step - accuracy: 0.9239 - loss: 0.2157 - val_accuracy: 0.7854 - val_loss: 0.6923
Epoch 16/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9192 - loss: 0.2338 - val_accuracy: 0.7996 - val_loss: 0.6667
Epoch 17/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9291 - loss: 0.1885 - val_accuracy: 0.7899 - val_loss: 0.6974
Epoch 18/20
169/169 ━━━━━━━━━━ 5s 18ms/step - accuracy: 0.9233 - loss: 0.1956 - val_accuracy: 0.8226 - val_loss: 0.6568
Epoch 19/20
169/169 ━━━━━━━━━━ 3s 17ms/step - accuracy: 0.9401 - loss: 0.1503 - val_accuracy: 0.8285 - val_loss: 0.6314
Epoch 20/20
169/169 ━━━━━━━━━━ 5s 18ms/step - accuracy: 0.9417 - loss: 0.1535 - val_accuracy: 0.8174 - val_loss: 0.6327
```

Visualize the model results

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

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

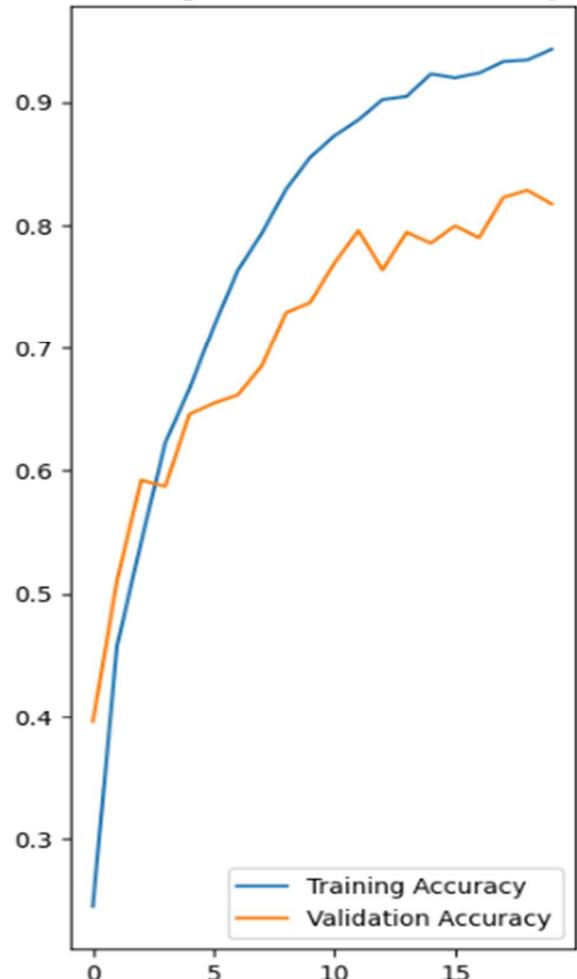
epochs_range = range(epochs)

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')

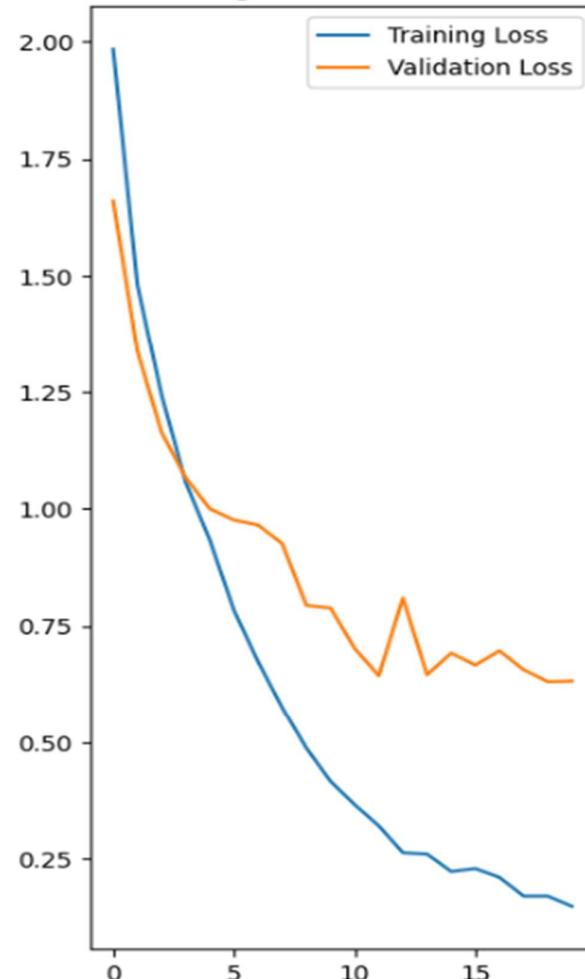
plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```



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.