# Cat-Dog-Image-Classifier

March 8, 2024

Data Set Downloading

[ ]:
```
!kaggle datasets download -d salader/dogs-vs-cats
```

```
dogs-vs-cats.zip: Skipping, found more recently modified local copy (use --force
to force download)
```

Importing the Nessessary libraries

[ ]:
```
import numpy as np
import os
import sys
import cv2
import matplotlib.pyplot as plt
import pickle
import random
import pandas as pd

import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,Dropout,Activation,␣
 ↪Flatten,Conv2D,MaxPooling2D
import pickle

import matplotlib.pyplot as plt
%matplotlib inline


from tensorflow import keras
from keras import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D, Flatten
```

```
WARNING:tensorflow:From
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

UnZipping the File and Loading the Data into a DataFrame

[ ]:
```
import zipfile
import os

zip_file_path = r'C:\Users\Asus\OneDrive\Desktop\Bharat Intern Data␣
 ↪Science\Cat-Dog-Image-Classifier\dogs-vs-cats.zip'
extracted_dir_path = r'C:\Users\Asus\OneDrive\Desktop\Bharat Intern Data␣
 ↪Science\Cat-Dog-Image-Classifier'

# Check if the target directory already exists
if os.path.exists(extracted_dir_path):
    print(f"The target directory '{extracted_dir_path}' already exists.␣
 ↪Terminating the process.")
else:
    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(extracted_dir_path)

    print(f"ZIP file '{zip_file_path}' has been successfully extracted to␣
 ↪'{extracted_dir_path}'.")
```

```
The target directory 'C:\Users\Asus\OneDrive\Desktop\Bharat Intern Data
Science\Cat-Dog-Image-Classifier' already exists. Terminating the process.
```

Creating Train and Test Batches and setting images Re-Size

[ ]:
```
from tensorflow import keras

# Training Data Generator
train_ds = keras.utils.image_dataset_from_directory(
    directory='C:\\Users\\Asus\\OneDrive\\Desktop\\Bharat Intern Data␣
 ↪Science\\Cat-Dog-Image-Classifier\\train',
    labels='inferred',
    label_mode='int',
    batch_size=32,
    image_size=(224, 224)
)

# Validation Data Generator
validation_ds = keras.utils.image_dataset_from_directory(
    directory='C:\\Users\\Asus\\OneDrive\\Desktop\\Bharat Intern Data␣
 ↪Science\\Cat-Dog-Image-Classifier\\test',
    labels='inferred',
    label_mode='int',
    batch_size=32,
    image_size=(224, 224)
)
```

```
Found 20000 files belonging to 2 classes.
Found 5000 files belonging to 2 classes.
```

Normalize function

[ ]:
```
import tensorflow as tf

# Normalize function
def process(image, label):
    image = tf.cast(image / 224.0, tf.float32)
    return image, label

# Apply normalization to training dataset
train_ds = train_ds.map(process)

# Apply normalization to validation dataset
validation_ds = validation_ds.map(process)
```

[ ]:
```
from tensorflow.keras import layers, models
from tensorflow.keras.callbacks import EarlyStopping

# Define a simplified CNN model with dropout
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224,␣
 ↪3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.25))  # Add dropout for regularization
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.25))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation='sigmoid'))

model.summary()
```

```
WARNING:tensorflow:From
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated.
Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool
is deprecated. Please use tf.nn.max_pool2d instead.

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
```

```
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 max_pooling2d (MaxPooling2  (None, 111, 111, 32)      0
 D)

 dropout (Dropout)           (None, 111, 111, 32)      0

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 flatten (Flatten)           (None, 186624)            0

 dense (Dense)               (None, 128)               23888000

 dropout_2 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 23907521 (91.20 MB)
Trainable params: 23907521 (91.20 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[ ]:
```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
```

```
WARNING:tensorflow:From
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer is
deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

[ ]:
```
# Implement early stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=3,␣
 ↪restore_best_weights=True)

# Train the model with augmented data and early stopping
history = model.fit(train_ds, validation_data=validation_ds, epochs=20,␣
 ↪callbacks=[early_stopping])
```

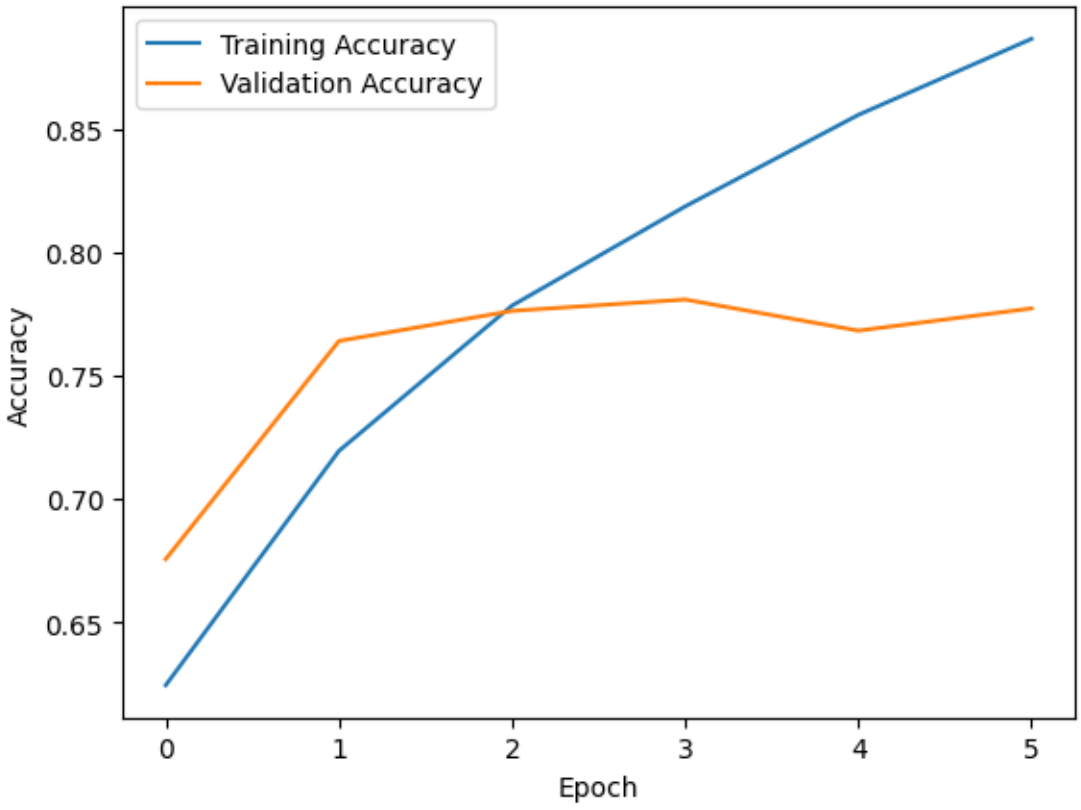
```
Epoch 1/20
WARNING:tensorflow:From
```

```
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue
is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From
c:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\engine\base_layer_utils.py:384: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.

625/625 [==============================] - 308s 491ms/step - loss: 0.7482 -
accuracy: 0.6245 - val_loss: 0.6151 - val_accuracy: 0.6756
Epoch 2/20
625/625 [==============================] - 301s 482ms/step - loss: 0.5564 -
accuracy: 0.7194 - val_loss: 0.5053 - val_accuracy: 0.7640
Epoch 3/20
625/625 [==============================] - 302s 483ms/step - loss: 0.4711 -
accuracy: 0.7783 - val_loss: 0.4873 - val_accuracy: 0.7762
Epoch 4/20
625/625 [==============================] - 302s 483ms/step - loss: 0.3973 -
accuracy: 0.8185 - val_loss: 0.4917 - val_accuracy: 0.7808
Epoch 5/20
625/625 [==============================] - 302s 483ms/step - loss: 0.3314 -
accuracy: 0.8557 - val_loss: 0.5287 - val_accuracy: 0.7682
Epoch 6/20
625/625 [==============================] - 301s 482ms/step - loss: 0.2662 -
accuracy: 0.8864 - val_loss: 0.5777 - val_accuracy: 0.7772
```

[ ]:
```python
# Plot training and validation accuracy
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```




[ ]:
```
import cv2
```

[ ]:
```
test_image = cv2.imread('C:\\Users\\Asus\\OneDrive\\Desktop\\R (1).jpeg')
```

[ ]:
```
plt.imshow(test_image)
```

[ ]:
```
<matplotlib.image.AxesImage at 0x204611b6f90>
```


[ ]:
```
test_image.shape
```

[ ]: (1802, 3000, 3)

[ ]:
```
test_image = cv2.resize(test_image,(224,224))
```

[ ]:
```
test_input = test_image.reshape(1, 224, 224, 3)
```

[ ]:
```
model.predict(test_input)

# Interpret the predictions
class_names = ['Cat', 'Dog']
predicted_class = class_names[int(predictions[0, 0] > 0.5)]

# Print the result
print(f'The model predicts that the image is a {predicted_class}.')
```

```
1/1 [==============================] - 0s 20ms/step
The model predicts that the image is a Dog.
```