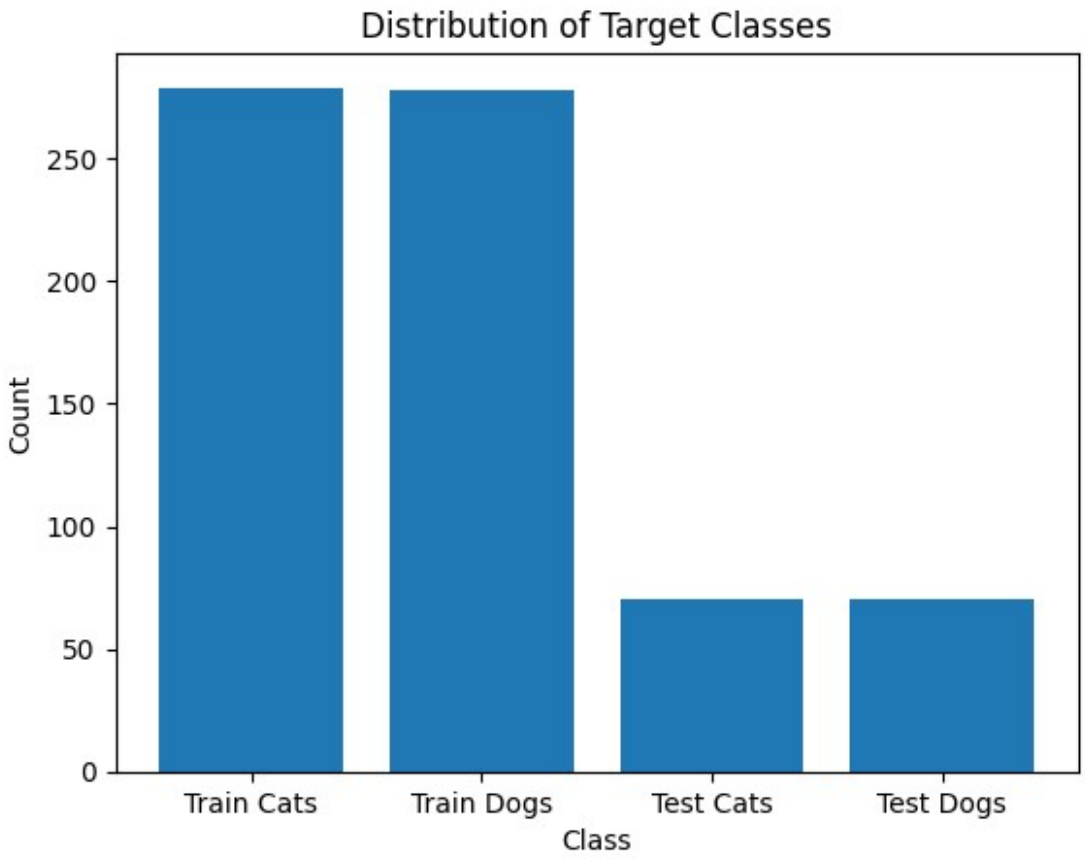
```
import os
os.environ['KAGGLE_CONFIG_DIR'] = './content'

!kaggle datasets download -d samuelcortinhas/cats-and-dogs-image-
classification

Warning: Your Kaggle API key is readable by other users on this
system! To fix this, you can run 'chmod 600 ./content/kaggle.json'
Downloading cats-and-dogs-image-classification.zip to /content
 95% 61.0M/64.4M [00:02<00:00, 35.4MB/s]
100% 64.4M/64.4M [00:02<00:00, 25.0MB/s]

!unzip cats-and-dogs-image-classification.zip
```

An 80/20 split was used between train and test.

```
import matplotlib.pyplot as plt

train_dir = "/content/train"
test_dir = "/content/test"

train_cats = len(os.listdir(os.path.join(train_dir, "cats")))
train_dogs = len(os.listdir(os.path.join(train_dir, "dogs")))

test_cats = len(os.listdir(os.path.join(test_dir, "cats")))
test_dogs = len(os.listdir(os.path.join(test_dir, "dogs")))

classes = ["Train Cats", "Train Dogs", "Test Cats", "Test Dogs"]
counts = [train_cats, train_dogs, test_cats, test_dogs]
plt.bar(classes, counts)
plt.title("Distribution of Target Classes")
plt.xlabel("Class")
plt.ylabel("Count")
plt.show()
```



Create a sequential model and evaluate on the test data

```
import matplotlib.pyplot as plt
import numpy as np

from keras import layers
from keras.models import Sequential

batchSize = 32
img_h = 224
img_w = 224
train_ds = keras.utils.image_dataset_from_directory(
    train_dir,
    validation_split=0.2,
    subset="training",
    seed=1234,
    image_size=(img_h,img_w),
    batch_size = batchSize
)
test_ds = keras.utils.image_dataset_from_directory(
    test_dir,
    validation_split=0.2,
    subset="validation",
    seed=1234,
```

```
    image_size=(img_h,img_w),
    batch_size = batchSize
)
class_names = train_ds.class_names
print(class_names)

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")

for image_batch, labels_batch in train_ds:
  print(image_batch.shape)
  print(labels_batch.shape)
  break


num_classes = len(class_names)

model = Sequential([
  layers.Rescaling(1./255, input_shape=(img_h, img_w, 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)
])

model.compile(optimizer='adam',
loss=keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])
model.summary()
history = model.fit(train_ds, validation_data=test_ds, epochs = 10)

epochs_range = range(10)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, history.history['accuracy'], label='Accuracy')
plt.plot(epochs_range, history.history['val_accuracy'],
label='Val_accuracy')
```

```
plt.legend(loc='lower right')
plt.title('Training and Val Accuracy')
```

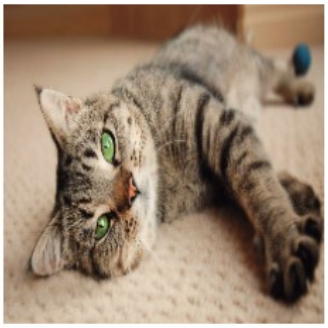
```
Found 557 files belonging to 2 classes.
Using 446 files for training.
Found 140 files belonging to 2 classes.
Using 28 files for validation.
['cats', 'dogs']
(32, 224, 224, 3)
(32,)
Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_1 (Rescaling)     (None, 224, 224, 3)       0

 conv2d_3 (Conv2D)           (None, 224, 224, 16)      448

 max_pooling2d_3 (MaxPooling  (None, 112, 112, 16)     0
 2D)

 conv2d_4 (Conv2D)           (None, 112, 112, 32)      4640

 max_pooling2d_4 (MaxPooling  (None, 56, 56, 32)       0
 2D)

 conv2d_5 (Conv2D)           (None, 56, 56, 64)        18496

 max_pooling2d_5 (MaxPooling  (None, 28, 28, 64)       0
 2D)

 flatten_1 (Flatten)         (None, 50176)             0

 dense_8 (Dense)             (None, 128)               6422656

 dense_9 (Dense)             (None, 2)                 258

=================================================================
Total params: 6,446,498
Trainable params: 6,446,498
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
14/14 [==============================] - 26s 2s/step - loss: 1.3296 -
accuracy: 0.4865 - val_loss: 0.6814 - val_accuracy: 0.5357
Epoch 2/10
14/14 [==============================] - 25s 2s/step - loss: 0.6887 -
accuracy: 0.5404 - val_loss: 0.6732 - val_accuracy: 0.6429
Epoch 3/10
14/14 [==============================] - 25s 2s/step - loss: 0.6631 -
```

```
accuracy: 0.6368 - val_loss: 0.6446 - val_accuracy: 0.6429
Epoch 4/10
14/14 [==============================] - 25s 2s/step - loss: 0.6703 -
accuracy: 0.6278 - val_loss: 0.6626 - val_accuracy: 0.5714
Epoch 5/10
14/14 [==============================] - 25s 2s/step - loss: 0.6229 -
accuracy: 0.6771 - val_loss: 0.5832 - val_accuracy: 0.7857
Epoch 6/10
14/14 [==============================] - 25s 2s/step - loss: 0.5415 -
accuracy: 0.7489 - val_loss: 0.6367 - val_accuracy: 0.5714
Epoch 7/10
14/14 [==============================] - 25s 2s/step - loss: 0.4958 -
accuracy: 0.7937 - val_loss: 0.5582 - val_accuracy: 0.7500
Epoch 8/10
14/14 [==============================] - 25s 2s/step - loss: 0.3754 -
accuracy: 0.8408 - val_loss: 0.5252 - val_accuracy: 0.7143
Epoch 9/10
14/14 [==============================] - 25s 2s/step - loss: 0.2717 -
accuracy: 0.8857 - val_loss: 0.5465 - val_accuracy: 0.6429
Epoch 10/10
14/14 [==============================] - 25s 2s/step - loss: 0.1963 -
accuracy: 0.9395 - val_loss: 0.4752 - val_accuracy: 0.8214

Text(0.5, 1.0, 'Training and Val Accuracy')
```

cats

cats

cats



cats

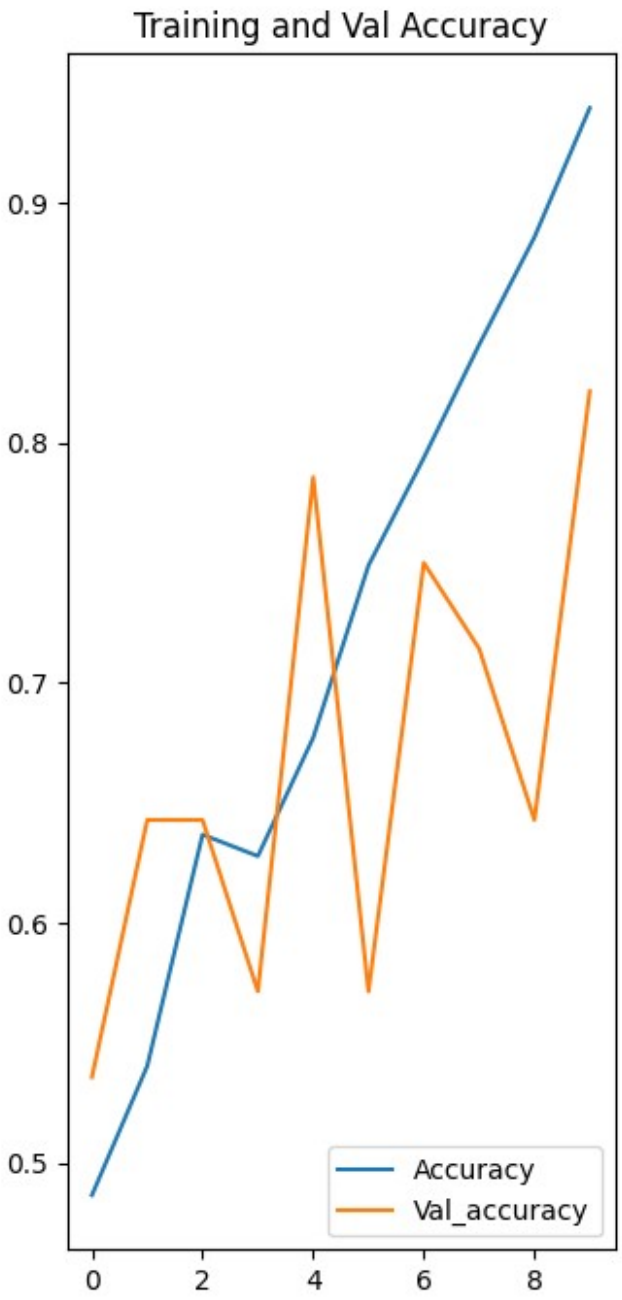
dogs

dogs

dogs

dogs

cats



CNN evaluates on the test data

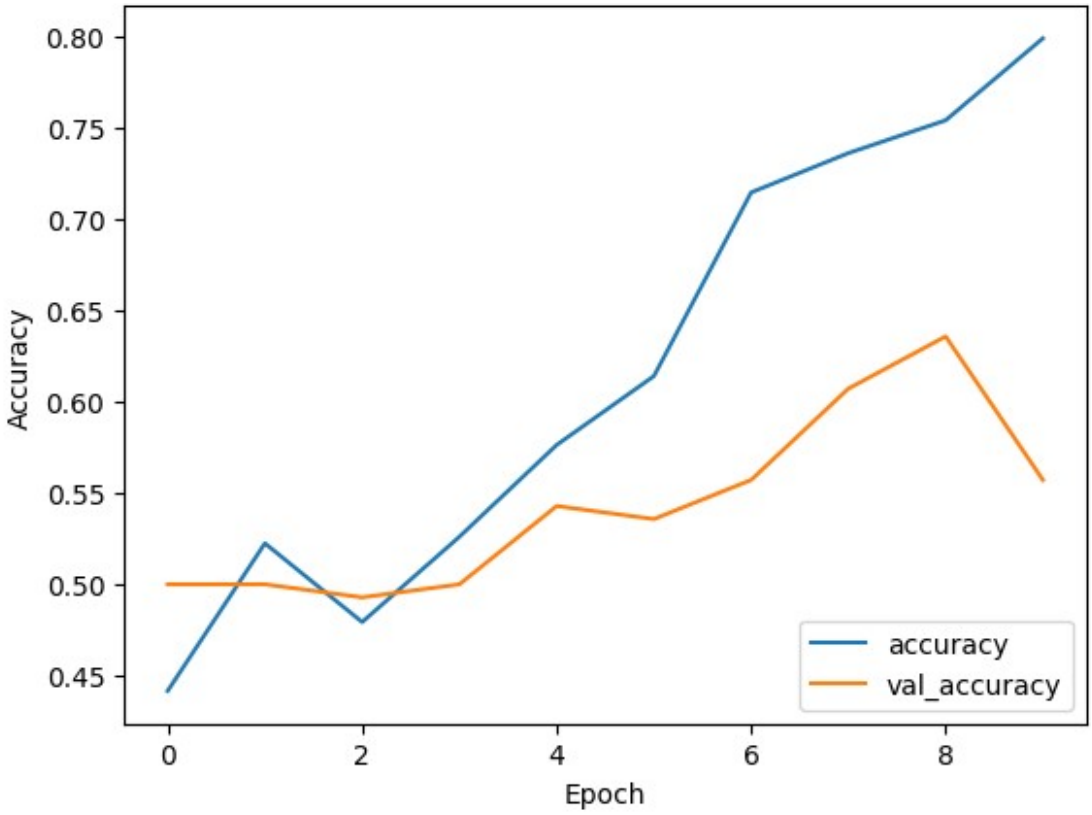
```
import pandas as pd
import numpy as np
import tensorflow as tf
import keras
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Flatten, Activation, Dense, Rescaling,
Softmax, Conv2D, MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
```

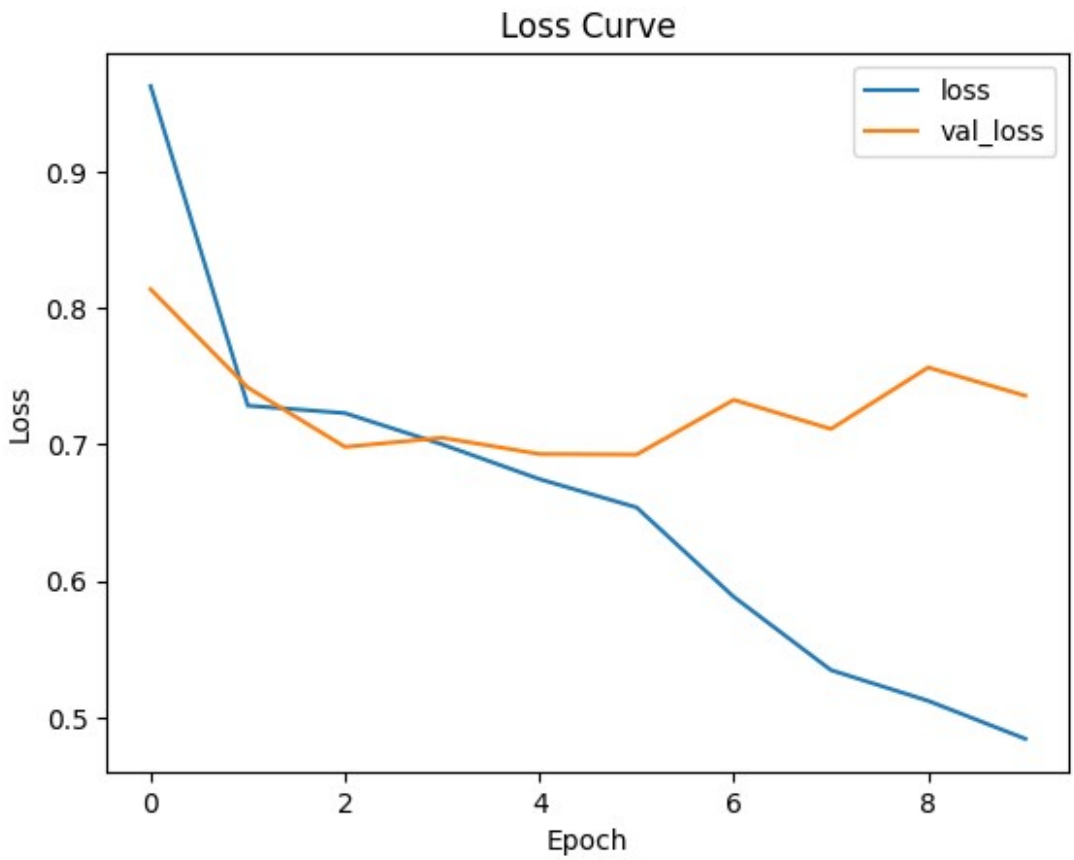
```
from keras.optimizers import adam


test_dir = "/content/test"
train_dir = "/content/train"

Train = ImageDataGenerator(rescale=1/255.)
Test = ImageDataGenerator(rescale=1/255.)

Train_data = Train.flow_from_directory(directory=train_dir,
target_size=(64,64), class_mode="sparse", batch_size=32)
Test_data = Test.flow_from_directory(directory=test_dir,
target_size=(64,64), class_mode="sparse", batch_size=32)

model1 = Sequential()

model1.add(Conv2D(32, (3, 3),activation='relu',input_shape=(64, 64,
3)))
model1.add(MaxPooling2D((2,2)))
model1.add(Conv2D(64, (3, 3),activation='relu'))
model1.add(MaxPooling2D((2,2)))
model1.add(Conv2D(64, (3, 3), activation='relu'))

model1.summary()
model1.add(Flatten())
model1.add(Dense(64,activation="relu"))
model1.add(Dense(10))

model1.compile(loss=keras.losses.SparseCategoricalCrossentropy(from_lo
gits=True), optimizer='adam', metrics=['accuracy'])

history = model1.fit(Train_data, epochs=10, validation_data=Test_data)


plt.plot(history.history['accuracy'], label='accuracy')
plt.plot(history.history['val_accuracy'], label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.show()

plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.title('Loss Curve')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
test_loss,test_acc = model1.evaluate(Test_data)
```

```
Found 557 images belonging to 2 classes.
Found 140 images belonging to 2 classes.
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_6 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_7 (Conv2D)           (None, 29, 29, 64)        18496

 max_pooling2d_7 (MaxPooling  (None, 14, 14, 64)       0
 2D)

 conv2d_8 (Conv2D)           (None, 12, 12, 64)        36928

=================================================================
Total params: 56,320
Trainable params: 56,320
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
18/18 [==============================] - 14s 701ms/step - loss: 0.9625
- accuracy: 0.4417 - val_loss: 0.8136 - val_accuracy: 0.5000
Epoch 2/10
18/18 [==============================] - 12s 689ms/step - loss: 0.7284
- accuracy: 0.5224 - val_loss: 0.7416 - val_accuracy: 0.5000
Epoch 3/10
18/18 [==============================] - 12s 674ms/step - loss: 0.7229
- accuracy: 0.4794 - val_loss: 0.6981 - val_accuracy: 0.4929
Epoch 4/10
18/18 [==============================] - 15s 816ms/step - loss: 0.7000
- accuracy: 0.5260 - val_loss: 0.7048 - val_accuracy: 0.5000
Epoch 5/10
18/18 [==============================] - 13s 727ms/step - loss: 0.6746
- accuracy: 0.5763 - val_loss: 0.6930 - val_accuracy: 0.5429
Epoch 6/10
18/18 [==============================] - 13s 674ms/step - loss: 0.6538
- accuracy: 0.6140 - val_loss: 0.6926 - val_accuracy: 0.5357
Epoch 7/10
18/18 [==============================] - 12s 679ms/step - loss: 0.5883
- accuracy: 0.7145 - val_loss: 0.7326 - val_accuracy: 0.5571
Epoch 8/10
18/18 [==============================] - 13s 723ms/step - loss: 0.5347
- accuracy: 0.7361 - val_loss: 0.7113 - val_accuracy: 0.6071
```

```
Epoch 9/10
18/18 [==============================] - 15s 820ms/step - loss: 0.5122
- accuracy: 0.7540 - val_loss: 0.7563 - val_accuracy: 0.6357
Epoch 10/10
18/18 [==============================] - 13s 738ms/step - loss: 0.4843
- accuracy: 0.7989 - val_loss: 0.7356 - val_accuracy: 0.5571
```






```
5/5 [==============================] - 2s 347ms/step - loss: 0.7356 -
accuracy: 0.5571
```

Pretrained model and transfer learning

```
from keras.api._v2.keras.applications.mobilenet_v2 import
preprocess_input
import matplotlib.pyplot as plt
import numpy as np
import keras

from keras.applications.mobilenet_v2 import MobileNetV2

train_data = keras.utils.image_dataset_from_directory(train_dir,
shuffle=True, batch_size=32, image_size=(224,224))
test_data = keras.utils.image_dataset_from_directory(test_dir,
shuffle=True, batch_size=32, image_size=(224,224))

augmentation = keras.Sequential([
    keras.layers.RandomFlip('horizontal'),
    keras.layers.RandomRotation(0.2)
])

preprocess_input = keras.applications.mobilenet_v2.preprocess_input
```

```
image_shape = (224,224) + (3,)
baseModel = keras.applications.MobileNetV2(input_shape=image_shape,
include_top=False, weights='imagenet')
imageBatch, labelBatch = next(iter(train_data))
Batch = baseModel(imageBatch)
print(Batch.shape)

baseModel.trainable = False

baseModel.summary()


Global_average = keras.layers.GlobalAveragePooling2D()
Batch_average = Global_average(Batch)
print(Batch_average.shape)

Prediction = keras.layers.Dense(1)
Prediction_batch = Prediction(Batch_average)
print(Prediction_batch.shape)


inputs = keras.Input(shape=(224,224,3))
x = augmentation(inputs)
x = preprocess_input(x)
x = baseModel(x,training=False)
x = Global_average(x)
x = keras.layers.Dropout(0.2)(x)
outputs = Prediction(x)
model = keras.Model(inputs, outputs)

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

history = model.fit(train_data, epochs=10, validation_data=test_data)

plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(history.history['accuracy'], label='Accuracy')
plt.plot(history.history['val_accuracy'], label='Val_accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.title('Accuracy')
```

```
Found 557 files belonging to 2 classes.
Found 140 files belonging to 2 classes.
(32, 7, 7, 1280)
```

```
Model: "mobilenetv2_1.00_224"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_14 (InputLayer)          [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864         ['input_14[0][0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0           ['bn_Conv1[0][0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288        ['Conv1_relu[0][0]']
 wiseConv2D)                    )

 expanded_conv_depthwise_BN (Ba  (None, 112, 112, 32  128        ['expanded_conv_depthwise[0][0]']
 tchNormalization)              )

 expanded_conv_depthwise_relu (  (None, 112, 112, 32  0          ['expanded_conv_depthwise_BN[0][0
 ReLU)                          )                                ]']
```

```
 expanded_conv_project (Conv2D)  (None, 112, 112, 16  512
['expanded_conv_depthwise_relu[0]
                                )                                 [0]']


 expanded_conv_project_BN (Batc  (None, 112, 112, 16  64
['expanded_conv_project[0][0]']
 hNormalization)                )


 block_1_expand (Conv2D)         (None, 112, 112, 96  1536
['expanded_conv_project_BN[0][0]'
                                )                                 ]


 block_1_expand_BN (BatchNormal  (None, 112, 112, 96  384
['block_1_expand[0][0]']
 ization)                       )


 block_1_expand_relu (ReLU)      (None, 112, 112, 96  0
['block_1_expand_BN[0][0]']
                                )


 block_1_pad (ZeroPadding2D)     (None, 113, 113, 96  0
['block_1_expand_relu[0][0]']
                                )


 block_1_depthwise (DepthwiseCo  (None, 56, 56, 96)  864
['block_1_pad[0][0]']
 nv2D)


 block_1_depthwise_BN (BatchNor  (None, 56, 56, 96)  384
['block_1_depthwise[0][0]']
 malization)


 block_1_depthwise_relu (ReLU)  (None, 56, 56, 96)   0
```

```
['block_1_depthwise_BN[0][0]']


 block_1_project (Conv2D)       (None, 56, 56, 24)   2304
['block_1_depthwise_relu[0][0]']


 block_1_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_1_project[0][0]']
 lization)



 block_2_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_1_project_BN[0][0]']


 block_2_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_2_expand[0][0]']
 ization)



 block_2_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_2_expand_BN[0][0]']


 block_2_depthwise (DepthwiseCo  (None, 56, 56, 144)  1296
['block_2_expand_relu[0][0]']
 nv2D)



 block_2_depthwise_BN (BatchNor  (None, 56, 56, 144)  576
['block_2_depthwise[0][0]']
 malization)



 block_2_depthwise_relu (ReLU)  (None, 56, 56, 144)  0
['block_2_depthwise_BN[0][0]']


 block_2_project (Conv2D)       (None, 56, 56, 24)   3456
['block_2_depthwise_relu[0][0]']


 block_2_project_BN (BatchNorma  (None, 56, 56, 24)  96
['block_2_project[0][0]']
```

```
 lization)



 block_2_add (Add)              (None, 56, 56, 24)   0
['block_1_project_BN[0][0]',

'block_2_project_BN[0][0]']


 block_3_expand (Conv2D)        (None, 56, 56, 144)  3456
['block_2_add[0][0]']


 block_3_expand_BN (BatchNormal  (None, 56, 56, 144)  576
['block_3_expand[0][0]']
 ization)



 block_3_expand_relu (ReLU)     (None, 56, 56, 144)  0
['block_3_expand_BN[0][0]']


 block_3_pad (ZeroPadding2D)    (None, 57, 57, 144)  0
['block_3_expand_relu[0][0]']


 block_3_depthwise (DepthwiseCo  (None, 28, 28, 144)  1296
['block_3_pad[0][0]']
 nv2D)



 block_3_depthwise_BN (BatchNor  (None, 28, 28, 144)  576
['block_3_depthwise[0][0]']
 malization)



 block_3_depthwise_relu (ReLU)  (None, 28, 28, 144)  0
['block_3_depthwise_BN[0][0]']


 block_3_project (Conv2D)       (None, 28, 28, 32)   4608
['block_3_depthwise_relu[0][0]']


 block_3_project_BN (BatchNorma  (None, 28, 28, 32)  128
```

```
['block_3_project[0][0]']
 lization)



 block_4_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_3_project_BN[0][0]']


 block_4_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_4_expand[0][0]']
 ization)



 block_4_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_4_expand_BN[0][0]']


 block_4_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_4_expand_relu[0][0]']
 nv2D)



 block_4_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_4_depthwise[0][0]']
 malization)



 block_4_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_4_depthwise_BN[0][0]']


 block_4_project (Conv2D)       (None, 28, 28, 32)   6144
['block_4_depthwise_relu[0][0]']


 block_4_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_4_project[0][0]']
 lization)



 block_4_add (Add)              (None, 28, 28, 32)   0
['block_3_project_BN[0][0]',

'block_4_project_BN[0][0]']
```

```
 block_5_expand (Conv2D)        (None, 28, 28, 192)  6144
['block_4_add[0][0]']


 block_5_expand_BN (BatchNormal  (None, 28, 28, 192)  768
['block_5_expand[0][0]']
 ization)



 block_5_expand_relu (ReLU)     (None, 28, 28, 192)  0
['block_5_expand_BN[0][0]']


 block_5_depthwise (DepthwiseCo  (None, 28, 28, 192)  1728
['block_5_expand_relu[0][0]']
 nv2D)



 block_5_depthwise_BN (BatchNor  (None, 28, 28, 192)  768
['block_5_depthwise[0][0]']
 malization)



 block_5_depthwise_relu (ReLU)  (None, 28, 28, 192)  0
['block_5_depthwise_BN[0][0]']


 block_5_project (Conv2D)       (None, 28, 28, 32)   6144
['block_5_depthwise_relu[0][0]']


 block_5_project_BN (BatchNorma  (None, 28, 28, 32)  128
['block_5_project[0][0]']
 lization)



 block_5_add (Add)              (None, 28, 28, 32)   0
['block_4_add[0][0]',

'block_5_project_BN[0][0]']


 block_6_expand (Conv2D)        (None, 28, 28, 192)  6144
```

['block_5_add[0][0]']

 block_6_expand_BN (BatchNormal (None, 28, 28, 192) 768
['block_6_expand[0][0]']
 ization)

 block_6_expand_relu (ReLU)   (None, 28, 28, 192) 0
['block_6_expand_BN[0][0]']

 block_6_pad (ZeroPadding2D)   (None, 29, 29, 192) 0
['block_6_expand_relu[0][0]']

 block_6_depthwise (DepthwiseCo (None, 14, 14, 192) 1728
['block_6_pad[0][0]']
 nv2D)

 block_6_depthwise_BN (BatchNor (None, 14, 14, 192) 768
['block_6_depthwise[0][0]']
 malization)

 block_6_depthwise_relu (ReLU) (None, 14, 14, 192) 0
['block_6_depthwise_BN[0][0]']

 block_6_project (Conv2D)   (None, 14, 14, 64)   12288
['block_6_depthwise_relu[0][0]']

 block_6_project_BN (BatchNorma (None, 14, 14, 64) 256
['block_6_project[0][0]']
 lization)

 block_7_expand (Conv2D)   (None, 14, 14, 384) 24576
['block_6_project_BN[0][0]']

 block_7_expand_BN (BatchNormal (None, 14, 14, 384) 1536
['block_7_expand[0][0]']

```
 ization)


 block_7_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_7_expand_BN[0][0]']


 block_7_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_7_expand_relu[0][0]']
 nv2D)


 block_7_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_7_depthwise[0][0]']
 malization)


 block_7_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_7_depthwise_BN[0][0]']


 block_7_project (Conv2D)       (None, 14, 14, 64)   24576
['block_7_depthwise_relu[0][0]']


 block_7_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_7_project[0][0]']
 lization)


 block_7_add (Add)              (None, 14, 14, 64)   0
['block_6_project_BN[0][0]',

'block_7_project_BN[0][0]']


 block_8_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_7_add[0][0]']


 block_8_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_8_expand[0][0]']
 ization)
```

```
 block_8_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_8_expand_BN[0][0]']


 block_8_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_8_expand_relu[0][0]']
 nv2D)



 block_8_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_8_depthwise[0][0]']
 malization)



 block_8_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_8_depthwise_BN[0][0]']


 block_8_project (Conv2D)       (None, 14, 14, 64)   24576
['block_8_depthwise_relu[0][0]']


 block_8_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_8_project[0][0]']
 lization)



 block_8_add (Add)              (None, 14, 14, 64)   0
['block_7_add[0][0]',

'block_8_project_BN[0][0]']


 block_9_expand (Conv2D)        (None, 14, 14, 384)  24576
['block_8_add[0][0]']


 block_9_expand_BN (BatchNormal  (None, 14, 14, 384)  1536
['block_9_expand[0][0]']
 ization)



 block_9_expand_relu (ReLU)     (None, 14, 14, 384)  0
['block_9_expand_BN[0][0]']
```

```
 block_9_depthwise (DepthwiseCo  (None, 14, 14, 384)  3456
['block_9_expand_relu[0][0]']
 nv2D)


 block_9_depthwise_BN (BatchNor  (None, 14, 14, 384)  1536
['block_9_depthwise[0][0]']
 malization)


 block_9_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_9_depthwise_BN[0][0]']


 block_9_project (Conv2D)       (None, 14, 14, 64)   24576
['block_9_depthwise_relu[0][0]']


 block_9_project_BN (BatchNorma  (None, 14, 14, 64)  256
['block_9_project[0][0]']
 lization)


 block_9_add (Add)              (None, 14, 14, 64)   0
['block_8_add[0][0]',

'block_9_project_BN[0][0]']


 block_10_expand (Conv2D)       (None, 14, 14, 384)  24576
['block_9_add[0][0]']


 block_10_expand_BN (BatchNorma  (None, 14, 14, 384)  1536
['block_10_expand[0][0]']
 lization)


 block_10_expand_relu (ReLU)    (None, 14, 14, 384)  0
['block_10_expand_BN[0][0]']


 block_10_depthwise (DepthwiseC  (None, 14, 14, 384)  3456
```

```
['block_10_expand_relu[0][0]']
 onv2D)



 block_10_depthwise_BN (BatchNo  (None, 14, 14, 384)  1536
['block_10_depthwise[0][0]']
 rmalization)



 block_10_depthwise_relu (ReLU)  (None, 14, 14, 384)  0
['block_10_depthwise_BN[0][0]']


 block_10_project (Conv2D)      (None, 14, 14, 96)   36864
['block_10_depthwise_relu[0][0]']


 block_10_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_10_project[0][0]']
 alization)



 block_11_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_10_project_BN[0][0]']


 block_11_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_11_expand[0][0]']
 lization)



 block_11_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_11_expand_BN[0][0]']


 block_11_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_11_expand_relu[0][0]']
 onv2D)



 block_11_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_11_depthwise[0][0]']
 rmalization)
```

```
 block_11_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
['block_11_depthwise_BN[0][0]']


 block_11_project (Conv2D)      (None, 14, 14, 96)   55296
['block_11_depthwise_relu[0][0]']


 block_11_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_11_project[0][0]']
 alization)



 block_11_add (Add)             (None, 14, 14, 96)   0
['block_10_project_BN[0][0]',

'block_11_project_BN[0][0]']


 block_12_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_11_add[0][0]']


 block_12_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_12_expand[0][0]']
 lization)



 block_12_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_12_expand_BN[0][0]']


 block_12_depthwise (DepthwiseC  (None, 14, 14, 576)  5184
['block_12_expand_relu[0][0]']
 onv2D)



 block_12_depthwise_BN (BatchNo  (None, 14, 14, 576)  2304
['block_12_depthwise[0][0]']
 rmalization)



 block_12_depthwise_relu (ReLU)  (None, 14, 14, 576)  0
```

```
['block_12_depthwise_BN[0][0]']


 block_12_project (Conv2D)      (None, 14, 14, 96)   55296
['block_12_depthwise_relu[0][0]']


 block_12_project_BN (BatchNorm  (None, 14, 14, 96)  384
['block_12_project[0][0]']
 alization)



 block_12_add (Add)             (None, 14, 14, 96)   0
['block_11_add[0][0]',

'block_12_project_BN[0][0]']


 block_13_expand (Conv2D)       (None, 14, 14, 576)  55296
['block_12_add[0][0]']


 block_13_expand_BN (BatchNorma  (None, 14, 14, 576)  2304
['block_13_expand[0][0]']
 lization)



 block_13_expand_relu (ReLU)    (None, 14, 14, 576)  0
['block_13_expand_BN[0][0]']


 block_13_pad (ZeroPadding2D)   (None, 15, 15, 576)  0
['block_13_expand_relu[0][0]']


 block_13_depthwise (DepthwiseC  (None, 7, 7, 576)   5184
['block_13_pad[0][0]']
 onv2D)



 block_13_depthwise_BN (BatchNo  (None, 7, 7, 576)   2304
['block_13_depthwise[0][0]']
 rmalization)
```

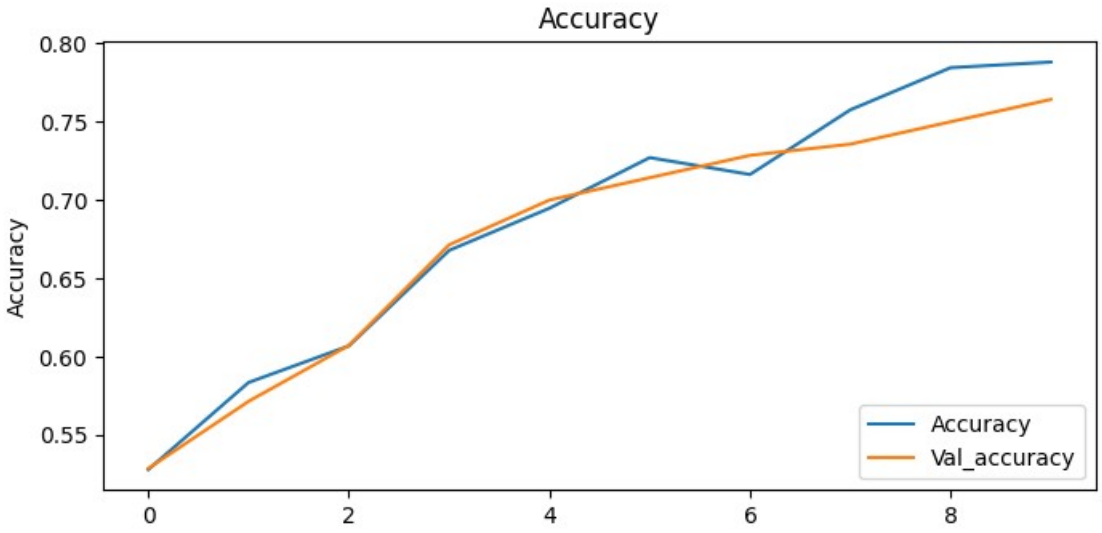
```
 block_13_depthwise_relu (ReLU)  (None, 7, 7, 576)   0
['block_13_depthwise_BN[0][0]']


 block_13_project (Conv2D)      (None, 7, 7, 160)    92160
['block_13_depthwise_relu[0][0]']


 block_13_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_13_project[0][0]']
 alization)



 block_14_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_13_project_BN[0][0]']


 block_14_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_14_expand[0][0]']
 lization)



 block_14_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_14_expand_BN[0][0]']


 block_14_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_14_expand_relu[0][0]']
 onv2D)



 block_14_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_14_depthwise[0][0]']
 rmalization)



 block_14_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_14_depthwise_BN[0][0]']


 block_14_project (Conv2D)      (None, 7, 7, 160)    153600
['block_14_depthwise_relu[0][0]']


 block_14_project_BN (BatchNorm  (None, 7, 7, 160)   640
```

```
['block_14_project[0][0]']
 alization)



 block_14_add (Add)             (None, 7, 7, 160)    0
['block_13_project_BN[0][0]',

'block_14_project_BN[0][0]']


 block_15_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_14_add[0][0]']


 block_15_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_15_expand[0][0]']
 lization)



 block_15_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_15_expand_BN[0][0]']


 block_15_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_15_expand_relu[0][0]']
 onv2D)



 block_15_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_15_depthwise[0][0]']
 rmalization)



 block_15_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_15_depthwise_BN[0][0]']


 block_15_project (Conv2D)      (None, 7, 7, 160)    153600
['block_15_depthwise_relu[0][0]']


 block_15_project_BN (BatchNorm  (None, 7, 7, 160)   640
['block_15_project[0][0]']
 alization)
```

```
 block_15_add (Add)             (None, 7, 7, 160)    0
['block_14_add[0][0]',

'block_15_project_BN[0][0]']


 block_16_expand (Conv2D)       (None, 7, 7, 960)    153600
['block_15_add[0][0]']


 block_16_expand_BN (BatchNorma  (None, 7, 7, 960)   3840
['block_16_expand[0][0]']
 lization)



 block_16_expand_relu (ReLU)    (None, 7, 7, 960)    0
['block_16_expand_BN[0][0]']


 block_16_depthwise (DepthwiseC  (None, 7, 7, 960)   8640
['block_16_expand_relu[0][0]']
 onv2D)



 block_16_depthwise_BN (BatchNo  (None, 7, 7, 960)   3840
['block_16_depthwise[0][0]']
 rmalization)



 block_16_depthwise_relu (ReLU)  (None, 7, 7, 960)   0
['block_16_depthwise_BN[0][0]']


 block_16_project (Conv2D)      (None, 7, 7, 320)    307200
['block_16_depthwise_relu[0][0]']


 block_16_project_BN (BatchNorm  (None, 7, 7, 320)   1280
['block_16_project[0][0]']
 alization)



 Conv_1 (Conv2D)                (None, 7, 7, 1280)   409600
```

```
['block_16_project_BN[0][0]']


 Conv_1_bn (BatchNormalization)  (None, 7, 7, 1280)  5120
['Conv_1[0][0]']


 out_relu (ReLU)                (None, 7, 7, 1280)   0
['Conv_1_bn[0][0]']


==================================================================================
============================
Total params: 2,257,984
Trainable params: 0
Non-trainable params: 2,257,984
__________________________________________________________________________________
_____________________________
(32, 1280)
(32, 1)
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_15 (InputLayer)       [(None, 224, 224, 3)]     0

 sequential_5 (Sequential)   (None, 224, 224, 3)       0

 tf.math.truediv_4 (TFOpLamb  (None, 224, 224, 3)      0
 da)

 tf.math.subtract_4 (TFOpLam  (None, 224, 224, 3)      0
 bda)

 mobilenetv2_1.00_224 (Funct  (None, 7, 7, 1280)       2257984
 ional)

 global_average_pooling2d_5   (None, 1280)             0
 (GlobalAveragePooling2D)

 dropout_4 (Dropout)         (None, 1280)              0

 dense_5 (Dense)             (None, 1)                 1281

=================================================================
Total params: 2,259,265
Trainable params: 1,281
Non-trainable params: 2,257,984
_________________________________________________________________
Epoch 1/10
```

```
18/18 [==============================] - 33s 2s/step - loss: 0.7650 -
accuracy: 0.5278 - val_loss: 0.7278 - val_accuracy: 0.5286
Epoch 2/10
18/18 [==============================] - 30s 2s/step - loss: 0.7040 -
accuracy: 0.5835 - val_loss: 0.6826 - val_accuracy: 0.5714
Epoch 3/10
18/18 [==============================] - 29s 2s/step - loss: 0.6666 -
accuracy: 0.6068 - val_loss: 0.6345 - val_accuracy: 0.6071
Epoch 4/10
18/18 [==============================] - 29s 2s/step - loss: 0.6031 -
accuracy: 0.6679 - val_loss: 0.5972 - val_accuracy: 0.6714
Epoch 5/10
18/18 [==============================] - 30s 2s/step - loss: 0.5446 -
accuracy: 0.6948 - val_loss: 0.5595 - val_accuracy: 0.7000
Epoch 6/10
18/18 [==============================] - 28s 2s/step - loss: 0.5042 -
accuracy: 0.7271 - val_loss: 0.5289 - val_accuracy: 0.7143
Epoch 7/10
18/18 [==============================] - 28s 2s/step - loss: 0.4980 -
accuracy: 0.7163 - val_loss: 0.5017 - val_accuracy: 0.7286
Epoch 8/10
18/18 [==============================] - 29s 2s/step - loss: 0.4764 -
accuracy: 0.7576 - val_loss: 0.4756 - val_accuracy: 0.7357
Epoch 9/10
18/18 [==============================] - 30s 2s/step - loss: 0.4523 -
accuracy: 0.7846 - val_loss: 0.4590 - val_accuracy: 0.7500
Epoch 10/10
18/18 [==============================] - 29s 2s/step - loss: 0.4243 -
accuracy: 0.7882 - val_loss: 0.4395 - val_accuracy: 0.7643

Text(0.5, 1.0, 'Accuracy')
```



Write up your analysis of the performance of various approaches

Convolutional Neural Networks (CNNs), Transfer Learning, and Sequential Models are three of the most powerful deep learning architectures in machine learning applications. Each architecture has unique advantages that make them suitable for different types of data analysis. CNNs are particularly effective in processing images and video frames because of their ability to capture spatial patterns and relationships between pixels. They are often used in computer vision applications, such as image classification. Sequential Models are ideal for processing sequential data such as time-series or text data because they can model the dependencies and temporal relationships between inputs. They are widely used in natural language processing and speech recognition. Additionly, transfer learning can save time and resources required for training a model from scratch, and is widely used in many applications such as computer vision. The choice of which architecture to use depends on the nature of the data and the specific task at hand.