Importing the zip file

```
from zipfile import ZipFile
file_name='meat_meat.zip'
with ZipFile(file_name,'r')as zip:
 zip.extractall()
print('Finished')
```

```
Finished
```

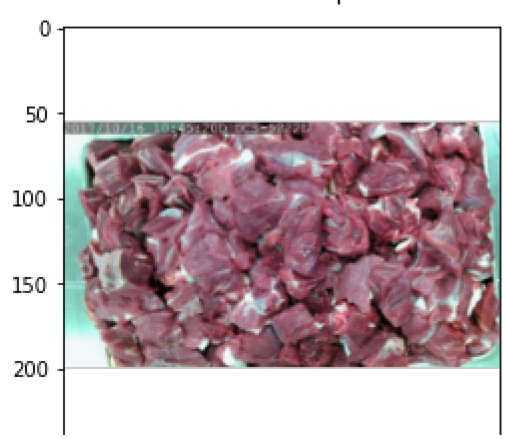
Importing the libraries

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

Fresh Meat

```
print(f"       Fresh meat example")
path = '/content/meat_last/Test/Fresh/test_20171016_104521D.png'
img=mpimg.imread(path)
imgplot=plt.imshow(img)
```

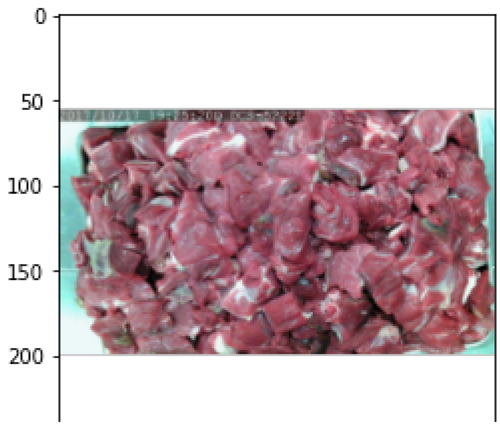
```
       Fresh meat example
```



Spoiled Meat

```
print(f"       Spoiled Meat Example")
path='/content/meat_last/Test/Spoiled/test_20171017_192521D.png'
img=mpimg.imread(path)
imgplot=plt.imshow(img)
```

Spoiled Meat Example



```
path_to_train = "/content/meat_last/Train"
path_to_test = "/content/meat_last/Test"
```

```
Generator = ImageDataGenerator()
train_data = Generator.flow_from_directory(path_to_train, (256, 256), batch_size=32)
test_data = Generator.flow_from_directory(path_to_test, (256, 256), batch_size=32)
```

```
Found 1106 images belonging to 2 classes.
Found 790 images belonging to 2 classes.
```

2.KISIM: Kendinize özgü bir CNN oluşturunuz.

Creating our Model

```
model=Sequential()
model.add(Conv2D(32, (4, 4), activation='relu', input_shape=(256,256,3)))
model.add(MaxPooling2D(pool_size=(2, 2),strides=2))
model.add(Dropout(0.3))
model.add(Conv2D(64, (4, 4), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
```

```
model.add(Dense(128, activation='relu'))
model.add(Dense(2,activation='sigmoid'))
```

Compiling the model

```
from tensorflow import keras
model.compile(loss=keras.losses.categorical_crossentropy,optimizer=keras.optimizers.Adam(),me
```

4.KISIM: Modelinizi eğitim verisi ile eğitirken, doğrulama verisi ile performansını gözleyecek şekilde history öğesine kaydediniz.

Training the Model

```
history= model.fit_generator(train_data,steps_per_epoch=1000//30,epochs=30,
verbose=1,
validation_data=test_data,validation_steps = 3)
```

```
33/33 [==============================] - 5s 151ms/step - loss: 0.1915 - accuracy: 0.9
Epoch 2/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1962 - accuracy: 0.9
Epoch 3/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1604 - accuracy: 0.9
Epoch 4/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1732 - accuracy: 0.9
Epoch 5/30
33/33 [==============================] - 5s 151ms/step - loss: 0.1511 - accuracy: 0.9
Epoch 6/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1522 - accuracy: 0.9
Epoch 7/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1702 - accuracy: 0.9
Epoch 8/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1564 - accuracy: 0.9
Epoch 9/30
33/33 [==============================] - 5s 153ms/step - loss: 0.1274 - accuracy: 0.9
Epoch 10/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1092 - accuracy: 0.9
Epoch 11/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1141 - accuracy: 0.9
Epoch 12/30
33/33 [==============================] - 5s 150ms/step - loss: 0.0901 - accuracy: 0.9
Epoch 13/30

33/33 [==============================] - 5s 152ms/step - loss: 0.1027 - accuracy: 0.9
Epoch 14/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0862 - accuracy: 0.9
Epoch 15/30
```

```
33/33 [==============================] - 5s 152ms/step - loss: 0.1002 - accuracy: 0.9
Epoch 16/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0857 - accuracy: 0.9
Epoch 17/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0923 - accuracy: 0.9
Epoch 18/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0777 - accuracy: 0.9
Epoch 19/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0680 - accuracy: 0.9
Epoch 20/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0939 - accuracy: 0.9
Epoch 21/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0831 - accuracy: 0.9
Epoch 22/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1218 - accuracy: 0.9
Epoch 23/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0907 - accuracy: 0.9
Epoch 24/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0653 - accuracy: 0.9
Epoch 25/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0729 - accuracy: 0.9
Epoch 26/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0593 - accuracy: 0.9
Epoch 27/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0681 - accuracy: 0.9
Epoch 28/30
33/33 [==============================] - 5s 160ms/step - loss: 0.0537 - accuracy: 0.9
Epoch 29/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0487 - accuracy: 0.9
```

3. KISIM: CNN modelinizin özetini çıktı olarak alınız (summary).

```
model.summary()
```

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_25 (Conv2D)          (None, 253, 253, 32)      1568

 max_pooling2d_25 (MaxPoolin  (None, 126, 126, 32)     0
 g2D)

 dropout_24 (Dropout)        (None, 126, 126, 32)      0

 conv2d_26 (Conv2D)          (None, 123, 123, 64)      32832

 max_pooling2d_26 (MaxPoolin  (None, 61, 61, 64)       0
 g2D)
```

```
 dropout_25 (Dropout)        (None, 61, 61, 64)        0

 flatten_13 (Flatten)        (None, 238144)            0

 dense_33 (Dense)            (None, 64)                15241280

 dense_34 (Dense)            (None, 128)               8320

 dense_35 (Dense)            (None, 2)                 258

=================================================================
Total params: 15,284,258
Trainable params: 15,284,258
Non-trainable params: 0
_________________________________________________________________
```

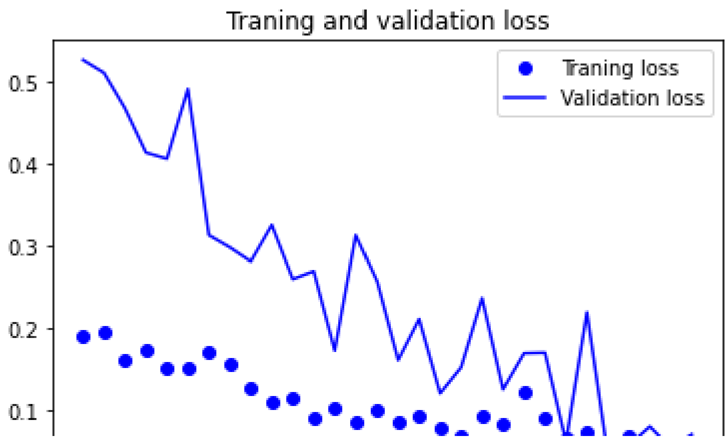
5.KISIM: History’e kaydettiğiniz Eğitim ve doğrulama kaybı (training and validation loss) ve eğitim ve doğrulama başarısının (training and validation accuracy) değişimini grafik olarak çizdiriniz

Ploting the Graph for training and validation accurancy

```
import matplotlib.pyplot as plt
acc=history.history['accuracy']
val_acc=history.history['val_accuracy']
loss=history.history['loss']
val_loss=history.history['val_loss']
epochs=range(1,len(acc)+1)
plt.plot(epochs,acc,'bo',label='Traning acc')
plt.plot(epochs,val_acc,'b',label='Validation acc')
plt.title('Traning and validation accuracy')
plt.legend()
plt.figure()



plt.plot(epochs,loss,'bo',label='Traning loss')
plt.plot(epochs,val_loss,'b',label='Validation loss')
plt.title('Traning and validation loss')
plt.legend()
plt.show()
```



6.KISIM: Overfitting görüyorsanız dropout ve augmentation ekleyerek bunu gidermeye çalışınız.

7.KISIM: Overfitting yok olduysa modelinizin kapasitesini (katman sayısı ve katmanlardaki nöron sayısı) artırınız. Başarınızın önünüzdeki tek engel overfitting kalıncaya kadar 6. KISIM ve 7. KISIM'ı tekrar ediniz.

IMPROVING THE MODEL

```
model=Sequential()
model.add(Conv2D(32, (5, 5), activation='relu', input_shape=(256,256,3)))
model.add(MaxPooling2D(pool_size=(2, 2),strides=2))
model.add(Dropout(0.3))
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2),strides=2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(256, activation='relu'))
model.add(Dense(2,activation='sigmoid'))
```

```
from tensorflow import keras
model.compile(loss=keras.losses.categorical_crossentropy,optimizer=keras.optimizers.Adam(),me
```

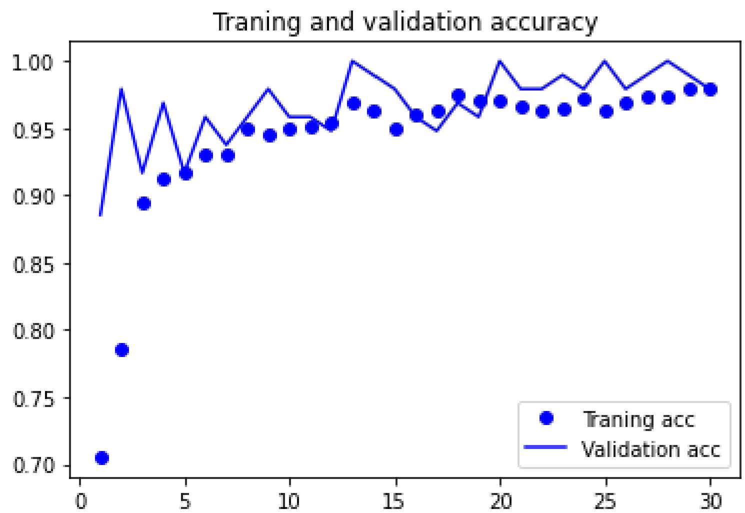
```
history= model.fit_generator(train_data,steps_per_epoch=1000//30,epochs=30,
verbose=1,
validation_data=test_data,validation_steps = 3)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: `Model.f
```

```
  This is separate from the ipykernel package so we can avoid doing imports until
Epoch 1/30
33/33 [==============================] - 5s 163ms/step - loss: 0.6215 - accuracy: 0.7
Epoch 2/30
33/33 [==============================] - 5s 155ms/step - loss: 0.4630 - accuracy: 0.7
Epoch 3/30
33/33 [==============================] - 5s 157ms/step - loss: 0.2790 - accuracy: 0.8
Epoch 4/30
33/33 [==============================] - 5s 155ms/step - loss: 0.2176 - accuracy: 0.9
Epoch 5/30
33/33 [==============================] - 5s 153ms/step - loss: 0.2369 - accuracy: 0.9
Epoch 6/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1655 - accuracy: 0.9
Epoch 7/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1557 - accuracy: 0.9
Epoch 8/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1135 - accuracy: 0.9
Epoch 9/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1378 - accuracy: 0.9
Epoch 10/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1064 - accuracy: 0.9
Epoch 11/30
33/33 [==============================] - 5s 155ms/step - loss: 0.1038 - accuracy: 0.9
Epoch 12/30
33/33 [==============================] - 5s 159ms/step - loss: 0.1103 - accuracy: 0.9
Epoch 13/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0933 - accuracy: 0.9
Epoch 14/30
33/33 [==============================] - 5s 156ms/step - loss: 0.0852 - accuracy: 0.9
Epoch 15/30
33/33 [==============================] - 5s 155ms/step - loss: 0.1104 - accuracy: 0.9
Epoch 16/30
33/33 [==============================] - 5s 156ms/step - loss: 0.0986 - accuracy: 0.9
Epoch 17/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0925 - accuracy: 0.9
Epoch 18/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0623 - accuracy: 0.9
Epoch 19/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0800 - accuracy: 0.9
Epoch 20/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0707 - accuracy: 0.9
Epoch 21/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0864 - accuracy: 0.9
Epoch 22/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0933 - accuracy: 0.9
Epoch 23/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0685 - accuracy: 0.9
Epoch 24/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0703 - accuracy: 0.9
Epoch 25/30
33/33 [==============================] - 5s 154ms/step - loss: 0.1058 - accuracy: 0.9
Epoch 26/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0659 - accuracy: 0.9
Epoch 27/30
33/33 [==============================] - 5s 158ms/step - loss: 0.0627 - accuracy: 0.9
Epoch 28/30
```

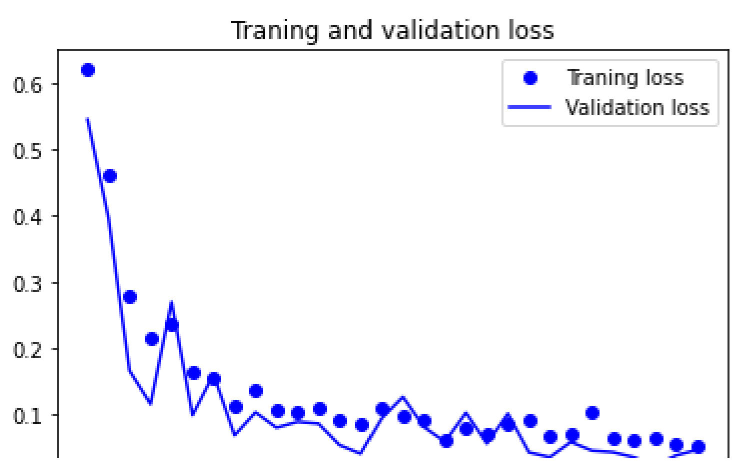
```
import matplotlib.pyplot as plt
acc=history.history['accuracy']
val_acc=history.history['val_accuracy']
loss=history.history['loss']
val_loss=history.history['val_loss']
epochs=range(1,len(acc)+1)
plt.plot(epochs,acc,'bo',label='Traning acc')
plt.plot(epochs,val_acc,'b',label='Validation acc')
plt.title('Traning and validation accuracy')
plt.legend()
plt.figure()
```

<Figure size 432x288 with 0 Axes>



<Figure size 432x288 with 0 Axes>

```
plt.plot(epochs,loss,'bo',label='Traning loss')
plt.plot(epochs,val_loss,'b',label='Validation loss')
plt.title('Traning and validation loss')
plt.legend()
plt.show()
```



8. KISIM: Overfitting’in başladığı noktayı tespit edip buna göre kaç epoch eğitmeniz gerektiğine karar veriniz

The best result I had when I was using 30 epoachs.

9-10-11 Kisimlar

```
model.evaluate(test_data)
```

```
35/35 [==============================] - 50s 1s/step - loss: 2.3722 - accuracy: 0.5000
[2.3721671104431152, 0.5]
```

Firstly I uploaded the zip file where we have all our meat datas.The meat photos were very similar to each other.After I imported the libraries and created a model I started training that model.After training the results were not too much satisfying,this can be seen very easily from the graphs.To improve the model I have changed the number of filters and also the size of them.At the same time I increased the number of neurons and because I had just to classes I used Sigmoid output activation.If I will write about epoach number I tried many times different types of epoachs from 5 to 50 (my Pc capasity)and the best result was when I was using 30 epoachs. .After retraining the model as it can be seen from the graphs our model had a great impovement.

50 sn. tamamlanma zamanı: 16:19

```
33/33 [==============================] - 5s 152ms/step - loss: 0.1962 - accuracy: 0.9175 - val_loss: 0.5113 - val_accuracy: 0.7812
Epoch 3/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1604 - accuracy: 0.9347 - val_loss: 0.4680 - val_accuracy: 0.6979
Epoch 4/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1732 - accuracy: 0.9290 - val_loss: 0.4139 - val_accuracy: 0.9479
Epoch 5/30
33/33 [==============================] - 5s 151ms/step - loss: 0.1511 - accuracy: 0.9451 - val_loss: 0.4064 - val_accuracy: 0.9375
Epoch 6/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1522 - accuracy: 0.9376 - val_loss: 0.4915 - val_accuracy: 0.7083
Epoch 7/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1702 - accuracy: 0.9405 - val_loss: 0.3133 - val_accuracy: 0.9896
Epoch 8/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1564 - accuracy: 0.9405 - val_loss: 0.2984 - val_accuracy: 0.9479
Epoch 9/30
33/33 [==============================] - 5s 153ms/step - loss: 0.1274 - accuracy: 0.9441 - val_loss: 0.2814 - val_accuracy: 0.8854
Epoch 10/30
33/33 [==============================] - 5s 150ms/step - loss: 0.1092 - accuracy: 0.9511 - val_loss: 0.3256 - val_accuracy: 0.9479
Epoch 11/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1141 - accuracy: 0.9530 - val_loss: 0.2598 - val_accuracy: 0.9271
Epoch 12/30
33/33 [==============================] - 5s 150ms/step - loss: 0.0901 - accuracy: 0.9655 - val_loss: 0.2695 - val_accuracy: 0.9375
Epoch 13/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1027 - accuracy: 0.9539 - val_loss: 0.1728 - val_accuracy: 0.9375
Epoch 14/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0862 - accuracy: 0.9655 - val_loss: 0.3131 - val_accuracy: 0.8333
Epoch 15/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1002 - accuracy: 0.9655 - val_loss: 0.2571 - val_accuracy: 0.8958
Epoch 16/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0857 - accuracy: 0.9612 - val_loss: 0.1611 - val_accuracy: 0.9688
Epoch 17/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0923 - accuracy: 0.9616 - val_loss: 0.2112 - val_accuracy: 0.9271
Epoch 18/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0777 - accuracy: 0.9664 - val_loss: 0.1206 - val_accuracy: 0.9792
Epoch 19/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0680 - accuracy: 0.9645 - val_loss: 0.1523 - val_accuracy: 0.9479
Epoch 20/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0939 - accuracy: 0.9655 - val_loss: 0.2368 - val_accuracy: 0.9688
Epoch 21/30
33/33 [==============================] - 5s 152ms/step - loss: 0.0831 - accuracy: 0.9655 - val_loss: 0.1254 - val_accuracy: 0.9688
Epoch 22/30
33/33 [==============================] - 5s 152ms/step - loss: 0.1218 - accuracy: 0.9568 - val_loss: 0.1696 - val_accuracy: 0.9479
Epoch 23/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0907 - accuracy: 0.9674 - val_loss: 0.1702 - val_accuracy: 0.9688
Epoch 24/30
33/33 [==============================] - 5s 154ms/step - loss: 0.0653 - accuracy: 0.9731 - val_loss: 0.0615 - val_accuracy: 0.9896
Epoch 25/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0729 - accuracy: 0.9773 - val_loss: 0.2196 - val_accuracy: 0.9792
Epoch 26/30
33/33 [==============================] - 5s 153ms/step - loss: 0.0593 - accuracy: 0.9770 - val_loss: 0.0443 - val_accuracy: 1.0000
```

```
Epoch 7/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1557 - accuracy: 0.9299 - val_loss: 0.1615 - val_accuracy: 0.9375
Epoch 8/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1135 - accuracy: 0.9501 - val_loss: 0.0689 - val_accuracy: 0.9583
Epoch 9/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1378 - accuracy: 0.9453 - val_loss: 0.1038 - val_accuracy: 0.9792
Epoch 10/30
33/33 [==============================] - 5s 156ms/step - loss: 0.1064 - accuracy: 0.9498 - val_loss: 0.0805 - val_accuracy: 0.9583
Epoch 11/30
33/33 [==============================] - 5s 155ms/step - loss: 0.1038 - accuracy: 0.9511 - val_loss: 0.0895 - val_accuracy: 0.9583
Epoch 12/30
33/33 [==============================] - 5s 159ms/step - loss: 0.1103 - accuracy: 0.9539 - val_loss: 0.0867 - val_accuracy: 0.9479
Epoch 13/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0933 - accuracy: 0.9683 - val_loss: 0.0542 - val_accuracy: 1.0000
Epoch 14/30
33/33 [==============================] - 5s 156ms/step - loss: 0.0852 - accuracy: 0.9635 - val_loss: 0.0412 - val_accuracy: 0.9896
Epoch 15/30
33/33 [==============================] - 5s 155ms/step - loss: 0.1104 - accuracy: 0.9491 - val_loss: 0.0952 - val_accuracy: 0.9792
Epoch 16/30
33/33 [==============================] - 5s 156ms/step - loss: 0.0986 - accuracy: 0.9607 - val_loss: 0.1271 - val_accuracy: 0.9583
Epoch 17/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0925 - accuracy: 0.9631 - val_loss: 0.0815 - val_accuracy: 0.9479
Epoch 18/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0623 - accuracy: 0.9750 - val_loss: 0.0592 - val_accuracy: 0.9688
Epoch 19/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0800 - accuracy: 0.9712 - val_loss: 0.1032 - val_accuracy: 0.9583
Epoch 20/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0707 - accuracy: 0.9712 - val_loss: 0.0564 - val_accuracy: 1.0000
Epoch 21/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0864 - accuracy: 0.9655 - val_loss: 0.1019 - val_accuracy: 0.9792
Epoch 22/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0933 - accuracy: 0.9626 - val_loss: 0.0432 - val_accuracy: 0.9792
Epoch 23/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0685 - accuracy: 0.9645 - val_loss: 0.0359 - val_accuracy: 0.9896
Epoch 24/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0703 - accuracy: 0.9722 - val_loss: 0.0589 - val_accuracy: 0.9792
Epoch 25/30
33/33 [==============================] - 5s 154ms/step - loss: 0.1058 - accuracy: 0.9626 - val_loss: 0.0458 - val_accuracy: 1.0000
Epoch 26/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0659 - accuracy: 0.9683 - val_loss: 0.0434 - val_accuracy: 0.9792
Epoch 27/30
33/33 [==============================] - 5s 158ms/step - loss: 0.0627 - accuracy: 0.9731 - val_loss: 0.0358 - val_accuracy: 0.9896
Epoch 28/30
33/33 [==============================] - 5s 155ms/step - loss: 0.0636 - accuracy: 0.9741 - val_loss: 0.0227 - val_accuracy: 1.0000
Epoch 29/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0553 - accuracy: 0.9789 - val_loss: 0.0384 - val_accuracy: 0.9896
Epoch 30/30
33/33 [==============================] - 5s 157ms/step - loss: 0.0541 - accuracy: 0.9798 - val_loss: 0.0469 - val_accuracy: 0.9792
```