Bangalore Institute of Technology
Department of Computer Science and Engineering
K R Road, V V Pura, Bengaluru-560004

KIDNEY STONE DETECTION USING DIP

Submitted as the **Content Beyond the Syllabus** for the subject Computer Graphics and Visualization (18CS62)

Submitted by

Aayush	**1BI20CS002**
Aryaman Jalali	**1BI20CS032**
Akshay P Bhagwat	**1BI20CS017**
Benaka Aditya	**1BI20CS039**

For academic year 2022-23

Under the guidance of
Prof/Dr. Girija . J

1. INTRODUCTION :

A kidney stone, also known as renal calculi, is a hard mass or crystal-like deposit that forms in the kidneys. It is composed of various substances, such as calcium, oxalate, uric acid, or cystine, that are present in urine. Kidney stones can vary in size, ranging from as small as a grain of sand to as large as a golf ball. The exact cause of kidney stone formation is not always clear, but several factors can contribute to their development, including:

- Dehydration: Insufficient fluid intake can lead to concentrated urine, increasing the risk of stone formation.
- Diet: Consuming foods high in oxalate, such as spinach, rhubarb, chocolate, and certain nuts, can contribute to calcium oxalate stones. A high-sodium or high-protein diet may also increase the risk.
- Family history: A person with a family history of kidney stones is more likely to develop them.
- Certain medical conditions: Conditions like hyperparathyroidism, gout, urinary tract infections, and certain digestive disorders can increase the risk of kidney stones.

1.1 KIDNEY STONE DETECTION USING DIP

Here are some general steps that can be considered:

1. Image Acquisition: Obtain images of the kidneys or urinary tract using an appropriate imaging modality such as ultrasound or CT scan. These images will serve as input for the subsequent DIP analysis.
2. Preprocessing: Apply preprocessing techniques to enhance the quality of the acquired images. This may involve noise reduction, contrast enhancement, and image filtering to improve the clarity of the kidney stone regions.
3. Segmentation: Segment the kidney stones from the background and surrounding tissues in the acquired images. Various segmentation algorithms can be employed, such as thresholding, edge detection, region-growing, or clustering methods, to separate the stone regions of interest.
4. Feature Extraction: Extract relevant features from the segmented kidney stone regions. These features could include shape descriptors, texture features, or statistical properties that capture the characteristics of the stones.
5. Classification: Utilize a classification algorithm or machine learning technique to classify the extracted features as kidney stones or non-stone regions. This could involve training a classifier using a labeled dataset to distinguish between normal kidney tissue and different types of stones.

2.Concept/Algorithm :

2.1 VGG-16 Model

- VGG16 is a type of CNN (Convolutional Neural Network) that is considered to be one of the best computer vision models to date.
- VGG16 is object detection and classification algorithm which is able to classify 1000 images of 1000 different categories with 92.7% accuracy. It is one of the popular algorithms for image classification and is easy to use with transfer learning.
- The 16 in VGG16 refers to 16 layers that have weights. In VGG16 there are thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers but it has only sixteen weight layers i.e., learnable parameters layer.
- VGG16 takes input tensor size as 224, 244 with 3 RGB channel
- Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with stride 1 and always used the same padding and maxpool layer of 2x2 filter of stride 2
- The convolution and max pool layers are consistently arranged throughout the whole architecture
- Conv-1 Layer has 64 number of filters, Conv-2 has 128 filters, Conv-3 has 256 filters, Conv 4 and Conv 5 has 512 filters.
- Three Fully-Connected (FC) layers follow a stack of convolutional layers: the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer.

3. CODE

```
# Import libraries, packages, modules, functions, etc...
import numpy as np
import cv2
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Activation, Dense, Flatten,
BatchNormalization, Conv2D, MaxPool2D, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix
import itertools
import os
import shutil
import random
import glob
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import warnings
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
import skimage
from skimage.restoration import denoise_nl_means, estimate_sigma,
denoise_tv_chambolle, denoise_bilateral,denoise_wavelet, unsupervised_wiener
from scipy.signal import convolve2d as conv2
from skimage.filters import median
import copy
from tqdm import tqdm
from scipy import ndimage as nd
import multiprocessing
from skimage.metrics import peak_signal_noise_ratio
from skimage.metrics import mean_squared_error
from skimage.metrics import structural_similarity as ssim
from itertools import repeat
import zipfile
import os
from IPython.display import FileLink
from sklearn.svm import SVC
from keras.callbacks import EarlyStopping
import pandas as pd
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.cluster import MiniBatchKMeans
from sklearn.metrics import accuracy_score
from tensorflow.keras.preprocessing import image
from skimage.morphology import disk
from tensorflow.keras.optimizers import SGD
warnings.simplefilter(action='ignore', category=FutureWarning)
!pip install natsort
!pip install MedPy
import natsort
from medpy.filter.smoothing import anisotropic_diffusion
```

```
%matplotlib inline
```

```
Requirement already satisfied: natsort in c:\users\prajwal
mr\anaconda3\lib\site-packages (8.4.0)
Requirement already satisfied: MedPy in c:\users\prajwal mr\anaconda3\lib\site-
packages (0.4.0)
Requirement already satisfied: numpy>=1.11.0 in c:\users\prajwal
mr\anaconda3\lib\site-packages (from MedPy) (1.22.4)
Requirement already satisfied: scipy>=1.1.0 in c:\users\prajwal
mr\anaconda3\lib\site-packages (from MedPy) (1.7.3)
Requirement already satisfied: SimpleITK>=1.1.0 in c:\users\prajwal
mr\anaconda3\lib\site-packages (from MedPy) (2.2.1)
```

In [2]:

```
#pip install opencv-python
```

In [3]:

```
#pip install tensorflow
```

In [4]:

```
# Get the complete data of kidney stone images
import os
def Get_data(dir, catagories, data):
    for category in catagories:
        path = os.path.join(dir, category)
        class_number = catagories.index(category)
        for img in os.listdir(path):
            try:
                img_array = cv2.imread(os.path.join(path , img))
                new_image = cv2.resize(img_array,(128,128))
                data.append([new_image , class_number])
            except Exception as e:
                pass

data = []
# Augmented Dataset
# Get_data("/kaggle/input/combined-aug-ks/KS_Detection", ["Combined_N",
"Combined_KS"], data)

# Original Dataset
Get_data("C:/Users/prajwal mr/Downloads/archive/CT_SCAN", ["Kidney_stone",
"Normal"], data)
```

In [5]:

```
catagories = ["Kidney_stone", "Normal"]
print(catagories.index("Kidney_stone"))
print(catagories.index("Normal"))
```

```
0
1
```

In [6]:

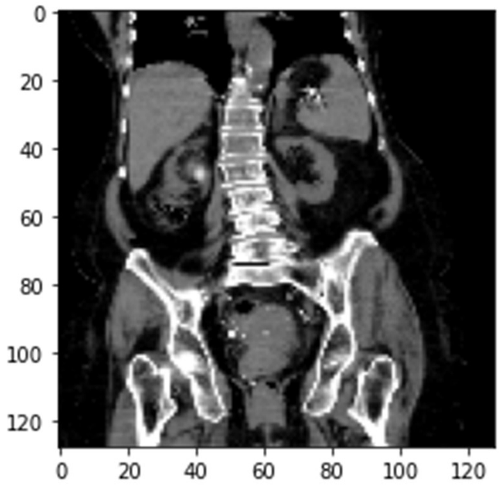
```
print('Number of kidney stone images',len(os.listdir('C:/Users/prajwal
mr/Downloads/archive/CT_SCAN/Kidney_stone')))
print('Number of Normal kidney images',len(os.listdir('C:/Users/prajwal
mr/Downloads/archive/CT_SCAN/Normal')))
```

```
Number of kidney stone images 781
Number of Normal kidney images 828
```

In [7]:

```
print(data[143][0].shape)
plt.imshow(data[143][0])
plt.show()
```

```
(128, 128, 3)
```



In [8]:

```
# Extract Features and Labels from the data
Features = []
Labels = []
for features , labels in data:
    Features.append(features)
    Labels.append(labels)
Features = np.array(Features)
Labels = np.array(Labels)
```

In [9]:

```
Features[5][1]
```

Out[9]:

```
array([[  0,   0,   0],
       [ 88,  88,  88],
       [  0,   0,   0],
       [  0,   0,   0],
       [  2,   2,   2],
       [  6,   6,   6],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  1,   1,   1],
       [  4,   4,   4],
       [ 15,  15,  15],
       [ 90,  90,  90],
       [  1,   1,   1],
       [  0,   0,   0],
       [  4,   4,   4],
       [ 37,  37,  37],
       [ 75,  75,  75],
       [113, 113, 113],
       [118, 118, 118],
       [121, 121, 121],
       [106, 106, 106],
       [135, 135, 135],
       [ 33,  33,  33],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [ 81,  81,  81],
       [105, 105, 105],
       [115, 115, 115],
       [107, 107, 107],
       [107, 107, 107],
       [103, 103, 103],
       [109, 109, 109],
       [104, 104, 104],
       [110, 110, 110],
       [109, 109, 109],
       [125, 125, 125],
       [116, 116, 116],
       [116, 116, 116],
       [120, 120, 120],
       [118, 118, 118],
       [125, 125, 125],
       [115, 115, 115],
       [115, 115, 115],
       [109, 109, 109],
       [ 59,  59,  59],
       [ 77,  77,  77],
       [100, 100, 100],
       [108, 108, 108],
       [107, 107, 107],
       [109, 109, 109],
       [112, 112, 112],
       [106, 106, 106],
       [ 82,  82,  82],
       [ 40,  40,  40],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [106, 106, 106],
       [113, 113, 113],
       [125, 125, 125],
       [118, 118, 118],
       [112, 112, 112],
       [111, 111, 111],
       [ 36,  36,  36],
       [  8,   8,   8],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  2,   2,   2],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  6,   6,   6],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0],
       [  0,   0,   0]], dtype=uint8)
```

In [10]:

```
# Shape of Features and Labels
print(Features.shape)
print(Labels.shape)
```

```
(1609, 128, 128, 3)
(1609,)
```

PREPROCESSING

In [11]:

```
def Processed(Processed_images):
    img_per_row = 4
    fig, ax = plt.subplots(nrows = 2, ncols = img_per_row, figsize=(10,10), 
subplot_kw = dict(xticks = [], yticks = []))
    for row in [0, 1]:
        for col in range(img_per_row):
            if(row * img_per_row + col == 7):
                break
            if(Processed_images[row * img_per_row + col][1] == 
"Bilateral_Denoised" or Processed_images[row * img_per_row + col][1] == 
"Wavelet_Denoised" or Processed_images[row * img_per_row + col][1] == 
"Non_Local_Means"):
                ax[row, col].imshow(Processed_images[row * img_per_row + 
col][0].astype('float64'), cmap = "gray")
            else:
                ax[row, col].imshow(Processed_images[row * img_per_row + 
col][0].astype('uint8'), cmap = "gray")
            ax[row, col].set_title(Processed_images[row * img_per_row + 
col][1])
            plt.axis('off')
    plt.show()
```

In [12]:

```
# Filters


# Efficient in reducing gaussian image
# Lowers the image resolution
def AnisotropicFilter_wholedataset(noised_dataset, niter, kappa, gamma, option, 
size):
    anisotropic_dataset = copy.deepcopy(noised_dataset)
    for i in tqdm(range(anisotropic_dataset.shape[0])):
        anisotropic_dataset[i] = anisotropic_diffusion(noised_dataset[i, :, :, 
:], niter=niter, kappa=kappa, gamma=gamma, option=option).reshape(size, size, 
1)
    return anisotropic_dataset


# Reduce spiky noise
# Non linear filter
# Smoothens the image
# Prevent blur and preserve sharp edges
# Only better for removing salt and pepper noise
def median_wholedataset(noised_dataset, filtersize, size):
    median_wholedata = copy.deepcopy(noised_dataset)
    for i in tqdm(range(noised_dataset.shape[0])):
        median_wholedata[i] = median(noised_dataset[i,:,:,:][:,:,0], 
np.ones((filtersize, filtersize))).reshape(size, size, 1)
    return median_wholedata
```

```
# Regardless of the frequency composition of the signal, eliminate noise while
keeping its properties
# Unable to preserve fine details in case of high noise data.
# There's a chance the wavelet coefficients are biased
def wavelet_wholedataset(noised_dataset, sigma, wavelet_levels, size):
    wavelet_wholedata = copy.deepcopy(noised_dataset)
    for i in tqdm(range(noised_dataset.shape[0])):
        wavelet_wholedata[i] = denoise_wavelet(noised_dataset[i,:,:,:],
sigma=sigma, channel_axis=-1, wavelet_levels=wavelet_levels,
rescale_sigma=True).reshape(size, size, 1)
    return wavelet_wholedata


# Better in preserving edges
# Effectively remove gaussian noise
# Not the best fit for salt and pepper noise
def BilateralFilter_wholedataset(noised_dataset, sigma_color, sigma_spatial,
channel_axis, size):
    bilateral_dataset = copy.deepcopy(noised_dataset)
    for i in tqdm(range(bilateral_dataset.shape[0])):
        bilateral_dataset[i] = denoise_bilateral(noised_dataset[i, :, :, :],
sigma_color = sigma_color, sigma_spatial=sigma_spatial,
channel_axis=channel_axis).reshape(size, size, 1)
    return bilateral_dataset


# Effect for gaussian noise
# Simplicity of algorithm is one of the pros
# Reduce the image details
# Unable to preserve edges
# Denoised images would be blurry.
def GaussianFilter_wholedataset(noised_dataset, sigma):
    gaussian_dataset = copy.deepcopy(noised_dataset)
    for i in tqdm(range(gaussian_dataset.shape[0])):
        gaussian_dataset[i] = nd.gaussian_filter(tuple(noised_dataset[i, :, :,
:]), sigma=sigma)
    return gaussian_dataset


# Preserve edges
# Better performance with redundant images
# Expensive, so non-suggestable for larger noise
def non_local_mean(image):
    sigma_est = np.mean(estimate_sigma(image, channel_axis=-1))
    patch_kw = dict(patch_size = 5,      # 5x5 patches
                patch_distance = 6,  # 13x13 search area
                channel_axis = -1)
    denoise_fast = denoise_nl_means(image, h = 0.6 * sigma_est,
sigma=sigma_est, fast_mode=True, **patch_kw)
    return denoise_fast


# Apply bilateral filter with d = 15
# sigmaColor = sigmaSpace = 75
# d: Diameter of each pixel neighborhood.
# sigmaColor: Value of sigma  in the color space. The greater the value, the
```

```
colors farther to each other will start to get mixed.
# sigmaSpace: Value of sigma  in the coordinate space. The greater its value,
the more further pixels will mix together, given that their colors lie within
the sigmaColor range.
def cv2bilateralFilter(image, d, sigmaColor, sigmaSpace):
    bilateral = cv2.bilateralFilter(image, d, sigmaColor, sigmaSpace)
    bilateral = bilateral.astype('uint8')
    return bilateral




# Execution
# ani = AnisotropicFilter_wholedataset(noised_dataset, 50, 20, 0.2, 1)
# bi = BilateralFilter_wholedataset(noised_dataset, 15, -1)
# gauss = GaussianFilter_wholedataset(noised_dataset, 2)

# median_img = median(Features[0, :, :, :][:,:,0], disk(3), mode = 'constant',
cval = 0.0).reshape(128, 128, 1)
# wavelet_img = denoise_wavelet(Features[0, :, :, :], sigma = 0.12,
channel_axis = -1,convert2ycbcr=True, method='BayesShrink', mode='soft',
rescale_sigma=True).reshape(128, 128, 3)
# bilateral_image = denoise_bilateral(Features[0], sigma_spatial = 10,
channel_axis = -1)
# anisotropic_image = anisotropic_diffusion(Features[700,:,:,:][:,:,0], niter =
100, kappa = 10, gamma = 0.02, option=1)
# anisotropic_image = anisotropic_diffusion(Features[700,:,:,:][:,:,0], niter =
50, kappa = 10, gamma = 0.02, option=1)
# anisotropic_image = anisotropic_diffusion(Features[60,:,:,:][:,:,0], niter =
50, kappa = 5, gamma = 0.005, option=1)
# anisotropic_image = anisotropic_diffusion(Features[600,:,:,:][:,:,0], niter =
50, kappa = 5, gamma = 0.000001, option=1)
# anisotropic_image = anisotropic_image.astype('uint8')


# Metrics
def psnr_wholedataset(dataset_original, dataset_denoised):

    sumpsnr = 0
    avgpsnr = 0
    for i in tqdm(range(dataset_original.shape[0])):
        true_min, true_max = np.min(dataset_original[i, :, :, :]),
np.max(dataset_original[i, :, :, :])
        dataRange = abs(true_min)+abs(true_max)
        psnr = peak_signal_noise_ratio(dataset_original[i],
dataset_denoised[i], data_range=dataRange)
        sumpsnr += psnr
    avgpsnr = sumpsnr/dataset_original.shape[0]

    return avgpsnr

# psnr_wholedataset(data, noised_dataset)


# Execution
# true_min, true_max = np.min(Features[0]), np.max(Features[0])
# dataRange = abs(true_min) + abs(true_max)
# print("PSNR: ", peak_signal_noise_ratio(Features[143, :, :, :][:,:,0],
anisotropic_image, data_range = dataRange))
# print("MSE: ", mean_squared_error(Features[143,:,:,:][:,:,0],
anisotropic_image))
# print("SSIM: ", ssim(Features[143, :, :, :][:,:,0], anisotropic_image,
multichannel=True, gaussian_weights = True, sigma=1.5,
use_sample_covariance=False))


# plt.imshow(np.hstack((wavelet_img, Features[143, :, :,
:].reshape(128,128,3))), cmap = "gray")
```

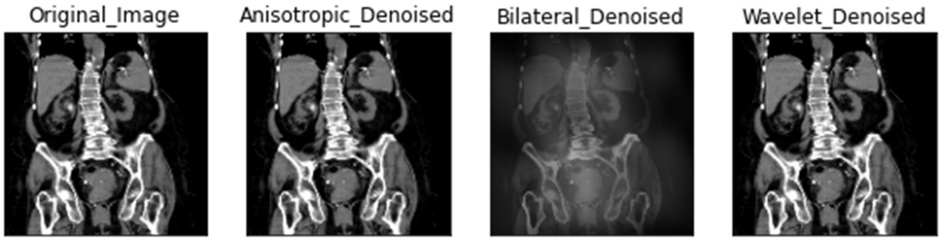
In [13]:

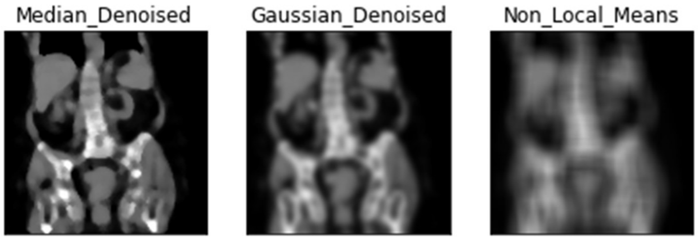
```
true_min, true_max = np.min(Features[143]), np.max(Features[143])
dataRange = abs(true_min) + abs(true_max)
bilateral = cv2bilateralFilter(Features[143,:,:,:], 15, 75, 75)
print("PSNR: ", peak_signal_noise_ratio(Features[143,:,:,:], bilateral,
data_range = dataRange))
print("MSE: ", mean_squared_error(Features[143,:,:,:], bilateral))
print("SSIM: ", ssim(Features[143,:,:,:], bilateral, multichannel = True,
gaussian_weights = True, sigma=1.5, use_sample_covariance=False))
```

```
PSNR:  31.786955991118436
MSE:  43.0908203125
SSIM:  0.9190078489547154
```

In [14]:

```
anisotropic_image = anisotropic_diffusion(Features[143,:,:,:][:,:,0], niter =
50, kappa = 5, gamma = 0.005, option=1)
bilateral_image = denoise_bilateral(Features[143], sigma_spatial = 10,
channel_axis = -1)
wavelet_img = denoise_wavelet(Features[143, :, :, :], sigma = 0.12,
channel_axis = -1,convert2ycbcr = True, method = 'BayesShrink', mode = 'soft',
rescale_sigma = True).reshape(128, 128, 3)
median_img = median(Features[143, :, :, :][:,:,0], disk(3), mode = 'constant',
cval = 0.0).reshape(128, 128, 1)
guassian_img = nd.gaussian_filter(tuple(Features[143, :, :, :]), sigma = 2)
non_local_mean_img = non_local_mean(Features[143])

Processed_images = [[Features[143, :, :, :], "Original_Image"],
[anisotropic_image, "Anisotropic_Denoised"], [bilateral_image,
"Bilateral_Denoised"], [wavelet_img, "Wavelet_Denoised"],
[median_img,"Median_Denoised"], [guassian_img, "Gaussian_Denoised"],
[non_local_mean_img, "Non_Local_Means"]]

Processed(Processed_images)
```





In [15]:

```
# Denoising the given data with anisotropic Filter
true_min, true_max = np.min(Features[0]), np.max(Features[0])
dataRange = abs(true_min) + abs(true_max)
anis_denoised = []
for i in range(len(Features)):
    anisotropic_image = anisotropic_diffusion(Features[i,:,:,:], niter = 100,
kappa = 10, gamma = 0.02, option = 1)
    anis_denoised.append(anisotropic_image)
anis_denoised = np.array(anis_denoised)
```

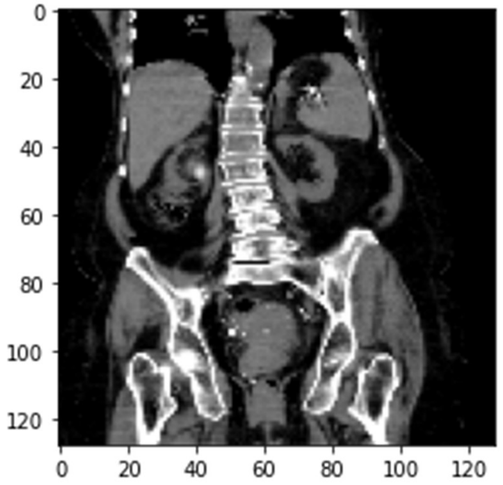
#Cropping, Thersholding(or masking), Edge Detection, Morphological Analysis

In [16]:

```
def crop(image_to_be_cropped):
    height, width = image_to_be_cropped.shape[:2]
    start_row, start_col = int(height * .24), int(width * .24)
    end_row, end_col = int(height * .78), int(width * .78)
#     start_row, start_col = int(height * .24), int(width * .24)
#     end_row, end_col = int(height * .42), int(width * .78)
    cropped_image = image_to_be_cropped[start_row:end_row, start_col:end_col]
    return cropped_image

def threshold(img, thresh1 = 254):
    return ((img > thresh1) * 255).astype('uint8')


# Canny edge detection is a technique to extract useful structural information
from different vision objects and dramatically reduce the amount of data to be
processed.
def edge_detection(Image):
    img = cv2.Canny(Image, 100, 200)
    return img

# To see the pixel values division in the images
def hist_plot(Image):
    plt.hist(Image.flat, bins = 100, range = (0, 255))
    plt.show()
```

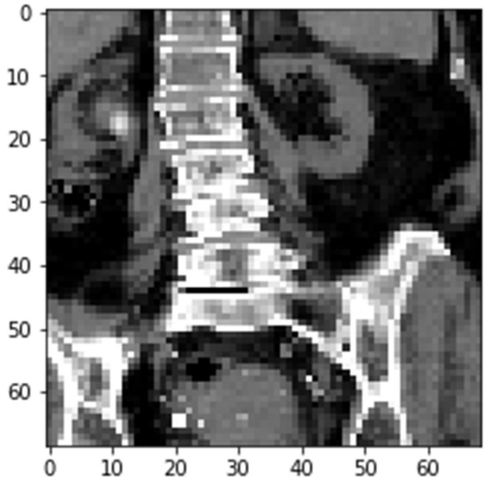
In [17]:

```
# Convert BGR image to Gray Scale image
gray_image = cv2.cvtColor(data[143][0], cv2.COLOR_BGR2GRAY)

# expanding dimensions from 128,128 to 128,128,1
gray_image = np.expand_dims(gray_image, axis = -1)

plt.imshow(gray_image, cmap = "gray")
plt.show()
```



In [18]:

```
# Crop a Image
cropped_image = crop(Features[143,:,:,:])
plt.imshow(cropped_image)
plt.show()
```



In [19]:

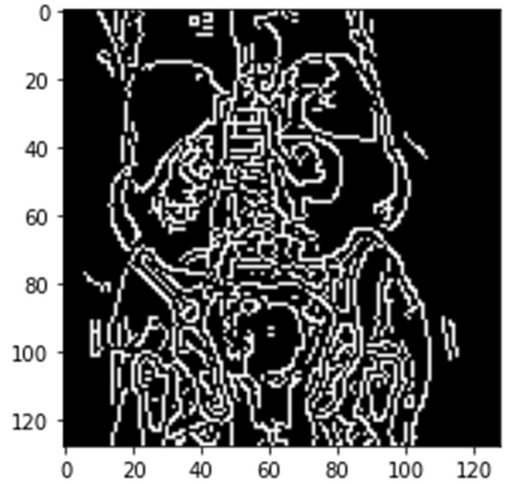
```
cropped_image.shape
```

Out[19]:

```
(69, 69, 3)
```

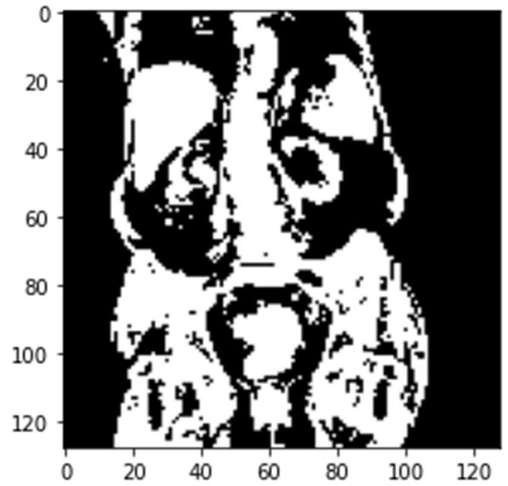
In [20]:

```
# Edge Detection
img = edge_detection(gray_image)
plt.imshow(img, cmap = "gray")
plt.show()
```



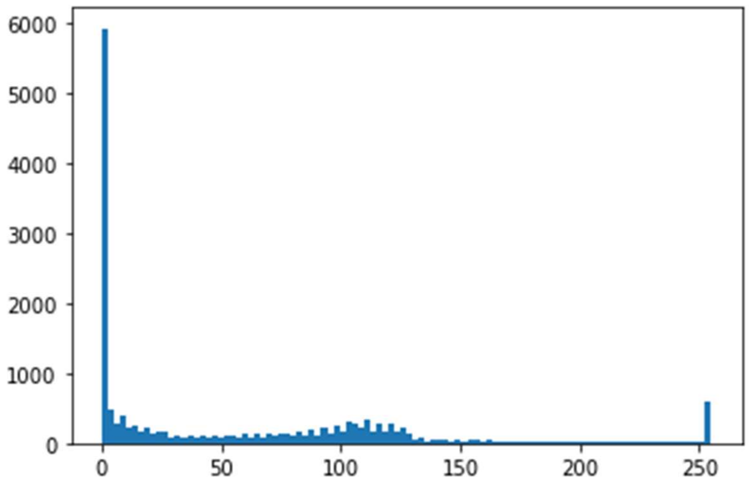
In [21]:

```
# Threshold masking using OTSU method
# Automatic Thresholding
# Otsu's method looks at every possible value for the threshold between
background and foreground, calculates the variance within each of the two
clusters, and selects the value for which the weighted sum of these variances
is the least.
re, th = cv2.threshold(gray_image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
plt.imshow(th, cmap = "gray")
plt.show()
```



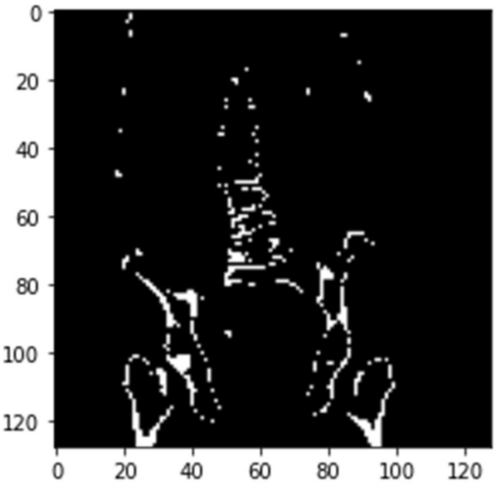
In [22]:

```
# See the pixel values division in the images
hist_plot(gray_image)
```



In [23]:

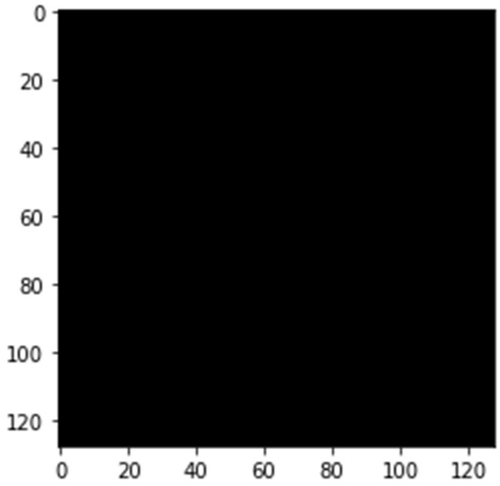
```
# Manual Thresholding on the basis of pixel divisions
img = threshold(gray_image)
plt.imshow(img, cmap = "gray")
plt.show()
```



In [24]:

```
# Mask of Image

# HSV or Hue Saturation Value is used to separate image luminance from color
information. This makes it easier when we are working on or need luminance of
the image/frame. HSV also used in situations where color description plays an
integral role.
# Hue, in the context of color and graphics, refers to the attribute of a
visible light due to which it is differentiated from or similar to the primary
colors
# Saturation describes the intensity of the color.
# Hue is determined by the dominant wavelength of the visible spectrum. It is
the attribute that permits colors to be classified as red, yellow, green, blue,
or an intermediate color. Saturation pertains the amount of white light mixed
with a hue.
# Value works in conjunction with saturation and describes the brightness or
intensity of the color, from 0 to 100 percent, where 0 is completely black, and
100 is the brightest and reveals the most color.

lower_yellow = np.array([35, 255, 255])
```

```
upper_yellow = np.array([25, 50, 70])
lower_blue = np.array([110,50,50])
upper_blue = np.array([130,255,255])
lower_white = np.array([180, 18, 255])
upper_white = np.array([0, 0, 231])
lower_black = np.array([180, 255, 30])
upper_black = np.array([0, 0, 0])

bgr_image = cv2.cvtColor(Features[143,:,:][:,:,0], cv2.COLOR_GRAY2BGR)

hsv = cv2.cvtColor(bgr_image, cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv, lower_black, upper_black)

detected_output = cv2.bitwise_and(hsv, hsv, mask =  mask)
plt.imshow(detected_output, cmap = "gray")
plt.show()
```



In [25]:

```
# Crop all the input images
Cropped_Features = []
for i in range(len(Features)):
    cropped_image = crop(Features[i,:,:,:])
    Cropped_Features.append(cropped_image)
Cropped_Features = np.array(Cropped_Features)
```

In [26]:

```
# Denoise all the input images with Bilateral Filte

# On original images
bil_de = []
for i in range(len(Features)):
    bilateral_de = cv2.bilateralFilter(Features[i], 15, 75, 75)
    bil_de.append(bilateral_de)
bil_de = np.array(bil_de)


# On cropped images
bilateral_denoised = []
for i in range(len(Cropped_Features)):
    bilateral = cv2.bilateralFilter(Cropped_Features[i], 15, 75, 75)
    bilateral_denoised.append(bilateral)
```

```
bilateral_denoised = np.array(bilateral_denoised)
```

In [27]:

```
# Threshold masking on all the input images

# Original
thresholded = []
for i in range(len(bilateral_denoised)):
    temp = threshold(bilateral_denoised[i])
    thresholded.append(temp)
thresholded = np.array(thresholded)


# Cropped
thresholded = []
for i in range(len(Cropped_Features)):
    temp = threshold(Cropped_Features[i])
    thresholded.append(temp)
thresholded = np.array(thresholded)
```

SPLITTING DATA

In [28]:

```
print(Features.shape)
print(Labels.shape)
```

```
(1609, 128, 128, 3)
(1609,)
```

In [29]:

```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(Features, Labels, test_size = 
0.30, random_state = 80, shuffle = True)

print(len(Xtrain), len(Xtest), len(Ytrain), len(Ytest))

# Converting the list to a numpy array as a requirement for the input in fit 
function.
Xtrain=np.array(Xtrain)
Xtest=np.array(Xtest)
Ytrain=np.array(Ytrain)
Ytest=np.array(Ytest)
```

```
1126 483 1126 483
```

In [ ]:

In [49]:

```
from tensorflow.keras.optimizers import SGD
```

In [50]:

```
model3 = Sequential()

model3.add(Conv2D(32, (3,3), activation='relu', input_shape=(128, 128, 3), 
padding='same'))
model3.add(MaxPool2D(2))
model3.add(Dropout(0.2))
```

```
model3.add(Conv2D(32, (3,3), activation='relu', padding='same'))
model3.add(MaxPool2D(pool_size=(2,2), strides = 2))
model3.add(Dropout(0.4))

model3.add(Conv2D(64, (3,3), activation='relu', padding='same'))
model3.add(MaxPool2D(pool_size=(2,2), strides = 2))
model3.add(Dropout(0.5))

model3.add(Conv2D(64, (3,3), activation='relu', padding='same'))
model3.add(MaxPool2D(pool_size=(2,2), strides = 2))

model3.add(Conv2D(128, (3,3), activation='relu', padding='same'))
model3.add(MaxPool2D(pool_size=(2,2), strides = 2))

model3.add(Conv2D(128, (3,3), activation='relu', padding='same'))
model3.add(MaxPool2D(pool_size=(2,2), strides = 2))
model3.add(Dropout(0.6))
model3.add(Flatten())
model3.add(Dense(512, activation='relu'))

model3.add(Dense(1, activation='sigmoid'))

model3.compile(loss = "binary_crossentropy", optimizer =
Adam(learning_rate=0.001), metrics=['accuracy'])
```

Different types of Model Training and Testing

CONVOLUTIONAL NEURAL NETWORK(CNN)

In [ ]:

In [51]:

```
# verbose=2 just specifies how much output to the console we want to see during
each epoch of training. The verbosity levels range from 0 to 2, so we're
getting the most verbose output.
# model.fit(Xtrain, Ytrain, epochs = 50, batch_size = 20, verbose = 2)

# Define a callback for early stopping
early_stopping = EarlyStopping(monitor = 'val_loss', patience = 10)

# Train the model
history = model3.fit(Xtrain, Ytrain, epochs = 200, batch_size = 20, callbacks =
[early_stopping])


#To this function, we pass in the test samples x, specify a batch_size, and
specify which level of verbosity we want from log messages during prediction
generation. The output from the predictions won't be relevant for us, so we're
setting verbose=0 for no output.
#Note that, unlike with training and validation sets, we do not pass the labels
of the test set to the model during the inference stage.
predictions = model3.predict(x = Xtest, batch_size = 20, verbose = 0)
Accuracy = model3.evaluate(Xtest, Ytest, verbose = 0)
print("Accuracy: ", Accuracy[1] * 100)
```

```
Epoch 1/200
57/57 [==============================] - ETA: 0s - loss: 9.2786 - accuracy:
0.4787WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 291ms/step - loss: 9.2786 -
accuracy: 0.4787
Epoch 2/200
57/57 [==============================] - ETA: 0s - loss: 0.7089 - accuracy:
0.5080WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 16s 283ms/step - loss: 0.7089 -
accuracy: 0.5080
Epoch 3/200
57/57 [==============================] - ETA: 0s - loss: 0.6944 - accuracy:
0.5169WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 17s 295ms/step - loss: 0.6944 -
accuracy: 0.5169
Epoch 4/200
57/57 [==============================] - ETA: 0s - loss: 0.6989 - accuracy:
0.5266WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 18s 323ms/step - loss: 0.6989 -
accuracy: 0.5266
Epoch 5/200
57/57 [==============================] - ETA: 0s - loss: 0.6980 - accuracy:
0.5160WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 332ms/step - loss: 0.6980 -
accuracy: 0.5160
Epoch 6/200
57/57 [==============================] - ETA: 0s - loss: 0.6983 - accuracy:
0.5027WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 417ms/step - loss: 0.6983 -
accuracy: 0.5027
Epoch 7/200
57/57 [==============================] - ETA: 0s - loss: 0.6969 - accuracy:
0.5098WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 406ms/step - loss: 0.6969 -
accuracy: 0.5098
Epoch 8/200
57/57 [==============================] - ETA: 0s - loss: 0.6946 - accuracy:
0.5169WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 387ms/step - loss: 0.6946 -
accuracy: 0.5169
Epoch 9/200
57/57 [==============================] - ETA: 0s - loss: 0.6986 - accuracy:
0.5053WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 400ms/step - loss: 0.6986 -
accuracy: 0.5053
Epoch 10/200
57/57 [==============================] - ETA: 0s - loss: 0.6921 - accuracy:
0.5249WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 347ms/step - loss: 0.6921 -
accuracy: 0.5249
```

```
Epoch 11/200
57/57 [==============================] - ETA: 0s - loss: 0.6959 - accuracy: 
0.5169WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 335ms/step - loss: 0.6959 - 
accuracy: 0.5169
Epoch 12/200
57/57 [==============================] - ETA: 0s - loss: 0.6928 - accuracy: 
0.5187WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 18s 324ms/step - loss: 0.6928 - 
accuracy: 0.5187
Epoch 13/200
57/57 [==============================] - ETA: 0s - loss: 0.6921 - accuracy: 
0.5355WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 328ms/step - loss: 0.6921 - 
accuracy: 0.5355
Epoch 14/200
57/57 [==============================] - ETA: 0s - loss: 0.6939 - accuracy: 
0.5169WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 327ms/step - loss: 0.6939 - 
accuracy: 0.5169
Epoch 15/200
57/57 [==============================] - ETA: 0s - loss: 0.6921 - accuracy: 
0.5231WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 346ms/step - loss: 0.6921 - 
accuracy: 0.5231
Epoch 16/200
57/57 [==============================] - ETA: 0s - loss: 0.6958 - accuracy: 
0.5151WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 352ms/step - loss: 0.6958 - 
accuracy: 0.5151
Epoch 17/200
57/57 [==============================] - ETA: 0s - loss: 0.6931 - accuracy: 
0.5018WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 356ms/step - loss: 0.6931 - 
accuracy: 0.5018
Epoch 18/200
57/57 [==============================] - ETA: 0s - loss: 0.6883 - accuracy: 
0.5471WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 353ms/step - loss: 0.6883 - 
accuracy: 0.5471
Epoch 19/200
57/57 [==============================] - ETA: 0s - loss: 0.6944 - accuracy: 
0.5115WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 440ms/step - loss: 0.6944 - 
accuracy: 0.5115
Epoch 20/200
57/57 [==============================] - ETA: 0s - loss: 0.6940 - accuracy: 
0.5329WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 30s 538ms/step - loss: 0.6940 -
accuracy: 0.5329
Epoch 21/200
57/57 [==============================] - ETA: 0s - loss: 0.6914 - accuracy:
0.5382WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 31s 547ms/step - loss: 0.6914 -
accuracy: 0.5382
Epoch 22/200
57/57 [==============================] - ETA: 0s - loss: 0.6925 - accuracy:
0.5240WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 41s 712ms/step - loss: 0.6925 -
accuracy: 0.5240
Epoch 23/200
57/57 [==============================] - ETA: 0s - loss: 0.6941 - accuracy:
0.5151WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 45s 790ms/step - loss: 0.6941 -
accuracy: 0.5151
Epoch 24/200
57/57 [==============================] - ETA: 0s - loss: 0.6897 - accuracy:
0.5391WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 38s 670ms/step - loss: 0.6897 -
accuracy: 0.5391
Epoch 25/200
57/57 [==============================] - ETA: 0s - loss: 0.6888 - accuracy:
0.5320WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 53s 929ms/step - loss: 0.6888 -
accuracy: 0.5320
Epoch 26/200
57/57 [==============================] - ETA: 0s - loss: 0.6934 - accuracy:
0.5142WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 55s 966ms/step - loss: 0.6934 -
accuracy: 0.5142
Epoch 27/200
57/57 [==============================] - ETA: 0s - loss: 0.6886 - accuracy:
0.5346WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 56s 982ms/step - loss: 0.6886 -
accuracy: 0.5346
Epoch 28/200
57/57 [==============================] - ETA: 0s - loss: 0.6889 - accuracy:
0.5462WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 50s 883ms/step - loss: 0.6889 -
accuracy: 0.5462
Epoch 29/200
57/57 [==============================] - ETA: 0s - loss: 0.6878 - accuracy:
0.5480WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 57s 998ms/step - loss: 0.6878 -
accuracy: 0.5480
```

```
Epoch 30/200
57/57 [==============================] - ETA: 0s - loss: 0.6866 - accuracy:
0.5497WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 50s 887ms/step - loss: 0.6866 -
accuracy: 0.5497
Epoch 31/200
57/57 [==============================] - ETA: 0s - loss: 0.6891 - accuracy:
0.5702WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 51s 899ms/step - loss: 0.6891 -
accuracy: 0.5702
Epoch 32/200
57/57 [==============================] - ETA: 0s - loss: 0.6870 - accuracy:
0.5488WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 48s 841ms/step - loss: 0.6870 -
accuracy: 0.5488
Epoch 33/200
57/57 [==============================] - ETA: 0s - loss: 0.6872 - accuracy:
0.5284WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 49s 860ms/step - loss: 0.6872 -
accuracy: 0.5284
Epoch 34/200
57/57 [==============================] - ETA: 0s - loss: 0.6866 - accuracy:
0.5417WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 47s 823ms/step - loss: 0.6866 -
accuracy: 0.5417
Epoch 35/200
57/57 [==============================] - ETA: 0s - loss: 0.6831 - accuracy:
0.5560WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 49s 859ms/step - loss: 0.6831 -
accuracy: 0.5560
Epoch 36/200
57/57 [==============================] - ETA: 0s - loss: 0.6807 - accuracy:
0.5728WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 438ms/step - loss: 0.6807 -
accuracy: 0.5728
Epoch 37/200
57/57 [==============================] - ETA: 0s - loss: 0.6782 - accuracy:
0.5728WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 352ms/step - loss: 0.6782 -
accuracy: 0.5728
Epoch 38/200
57/57 [==============================] - ETA: 0s - loss: 0.6820 - accuracy:
0.5604WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 381ms/step - loss: 0.6820 -
accuracy: 0.5604
Epoch 39/200
57/57 [==============================] - ETA: 0s - loss: 0.6744 - accuracy:
0.5933WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 385ms/step - loss: 0.6744 -
accuracy: 0.5933
Epoch 40/200
57/57 [==============================] - ETA: 0s - loss: 0.6715 - accuracy:
0.5933WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 383ms/step - loss: 0.6715 -
accuracy: 0.5933
Epoch 41/200
57/57 [==============================] - ETA: 0s - loss: 0.6738 - accuracy:
0.5941WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 368ms/step - loss: 0.6738 -
accuracy: 0.5941
Epoch 42/200
57/57 [==============================] - ETA: 0s - loss: 0.6603 - accuracy:
0.6128WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 372ms/step - loss: 0.6603 -
accuracy: 0.6128
Epoch 43/200
57/57 [==============================] - ETA: 0s - loss: 0.6719 - accuracy:
0.5737WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 392ms/step - loss: 0.6719 -
accuracy: 0.5737
Epoch 44/200
57/57 [==============================] - ETA: 0s - loss: 0.6821 - accuracy:
0.5631WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 374ms/step - loss: 0.6821 -
accuracy: 0.5631
Epoch 45/200
57/57 [==============================] - ETA: 0s - loss: 0.6603 - accuracy:
0.5959WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 399ms/step - loss: 0.6603 -
accuracy: 0.5959
Epoch 46/200
57/57 [==============================] - ETA: 0s - loss: 0.6686 - accuracy:
0.5906WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 384ms/step - loss: 0.6686 -
accuracy: 0.5906
Epoch 47/200
57/57 [==============================] - ETA: 0s - loss: 0.6618 - accuracy:
0.6030WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 393ms/step - loss: 0.6618 -
accuracy: 0.6030
Epoch 48/200
57/57 [==============================] - ETA: 0s - loss: 0.6737 - accuracy:
0.5728WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 28s 491ms/step - loss: 0.6737 -
accuracy: 0.5728
```

```
Epoch 49/200
57/57 [==============================] - ETA: 0s - loss: 0.6581 - accuracy: 
0.5933WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 382ms/step - loss: 0.6581 - 
accuracy: 0.5933
Epoch 50/200
57/57 [==============================] - ETA: 0s - loss: 0.6498 - accuracy: 
0.6146WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 394ms/step - loss: 0.6498 - 
accuracy: 0.6146
Epoch 51/200
57/57 [==============================] - ETA: 0s - loss: 0.6608 - accuracy: 
0.6190WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 384ms/step - loss: 0.6608 - 
accuracy: 0.6190
Epoch 52/200
57/57 [==============================] - ETA: 0s - loss: 0.6550 - accuracy: 
0.6163WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 36s 640ms/step - loss: 0.6550 - 
accuracy: 0.6163
Epoch 53/200
57/57 [==============================] - ETA: 0s - loss: 0.6353 - accuracy: 
0.6297WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 48s 838ms/step - loss: 0.6353 - 
accuracy: 0.6297
Epoch 54/200
57/57 [==============================] - ETA: 0s - loss: 0.6466 - accuracy: 
0.6234WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 50s 875ms/step - loss: 0.6466 - 
accuracy: 0.6234
Epoch 55/200
57/57 [==============================] - ETA: 0s - loss: 0.6411 - accuracy: 
0.6261WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 52s 910ms/step - loss: 0.6411 - 
accuracy: 0.6261
Epoch 56/200
57/57 [==============================] - ETA: 0s - loss: 0.6482 - accuracy: 
0.6217WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 52s 914ms/step - loss: 0.6482 - 
accuracy: 0.6217
Epoch 57/200
57/57 [==============================] - ETA: 0s - loss: 0.6581 - accuracy: 
0.6128WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 51s 899ms/step - loss: 0.6581 - 
accuracy: 0.6128
Epoch 58/200
57/57 [==============================] - ETA: 0s - loss: 0.6472 - accuracy: 
0.6083WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 51s 899ms/step - loss: 0.6472 -
accuracy: 0.6083
Epoch 59/200
57/57 [==============================] - ETA: 0s - loss: 0.6380 - accuracy:
0.6385WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 52s 909ms/step - loss: 0.6380 -
accuracy: 0.6385
Epoch 60/200
57/57 [==============================] - ETA: 0s - loss: 0.6367 - accuracy:
0.6226WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 53s 937ms/step - loss: 0.6367 -
accuracy: 0.6226
Epoch 61/200
57/57 [==============================] - ETA: 0s - loss: 0.6261 - accuracy:
0.6456WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 47s 825ms/step - loss: 0.6261 -
accuracy: 0.6456
Epoch 62/200
57/57 [==============================] - ETA: 0s - loss: 0.6241 - accuracy:
0.6439WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 50s 873ms/step - loss: 0.6241 -
accuracy: 0.6439
Epoch 63/200
57/57 [==============================] - ETA: 0s - loss: 0.6318 - accuracy:
0.6385WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 31s 539ms/step - loss: 0.6318 -
accuracy: 0.6385
Epoch 64/200
57/57 [==============================] - ETA: 0s - loss: 0.6261 - accuracy:
0.6554WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 361ms/step - loss: 0.6261 -
accuracy: 0.6554
Epoch 65/200
57/57 [==============================] - ETA: 0s - loss: 0.6164 - accuracy:
0.6750WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 368ms/step - loss: 0.6164 -
accuracy: 0.6750
Epoch 66/200
57/57 [==============================] - ETA: 0s - loss: 0.6307 - accuracy:
0.6448WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 371ms/step - loss: 0.6307 -
accuracy: 0.6448
Epoch 67/200
57/57 [==============================] - ETA: 0s - loss: 0.6000 - accuracy:
0.6696WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 368ms/step - loss: 0.6000 -
accuracy: 0.6696
```

```
Epoch 68/200
57/57 [==============================] - ETA: 0s - loss: 0.6052 - accuracy:
0.6758WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 456ms/step - loss: 0.6052 -
accuracy: 0.6758
Epoch 69/200
57/57 [==============================] - ETA: 0s - loss: 0.6024 - accuracy:
0.6767WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 397ms/step - loss: 0.6024 -
accuracy: 0.6767
Epoch 70/200
57/57 [==============================] - ETA: 0s - loss: 0.5964 - accuracy:
0.6572WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 423ms/step - loss: 0.5964 -
accuracy: 0.6572
Epoch 71/200
57/57 [==============================] - ETA: 0s - loss: 0.6036 - accuracy:
0.6643WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 387ms/step - loss: 0.6036 -
accuracy: 0.6643
Epoch 72/200
57/57 [==============================] - ETA: 0s - loss: 0.5950 - accuracy:
0.7007WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 391ms/step - loss: 0.5950 -
accuracy: 0.7007
Epoch 73/200
57/57 [==============================] - ETA: 0s - loss: 0.6010 - accuracy:
0.6767WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 371ms/step - loss: 0.6010 -
accuracy: 0.6767
Epoch 74/200
57/57 [==============================] - ETA: 0s - loss: 0.5845 - accuracy:
0.6803WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 399ms/step - loss: 0.5845 -
accuracy: 0.6803
Epoch 75/200
57/57 [==============================] - ETA: 0s - loss: 0.5599 - accuracy:
0.7114WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 427ms/step - loss: 0.5599 -
accuracy: 0.7114
Epoch 76/200
57/57 [==============================] - ETA: 0s - loss: 0.5740 - accuracy:
0.7034WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 404ms/step - loss: 0.5740 -
accuracy: 0.7034
Epoch 77/200
57/57 [==============================] - ETA: 0s - loss: 0.5890 - accuracy:
0.7034WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 398ms/step - loss: 0.5890 -
accuracy: 0.7034
Epoch 78/200
57/57 [==============================] - ETA: 0s - loss: 0.5718 - accuracy:
0.6998WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 450ms/step - loss: 0.5718 -
accuracy: 0.6998
Epoch 79/200
57/57 [==============================] - ETA: 0s - loss: 0.5539 - accuracy:
0.7105WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 470ms/step - loss: 0.5539 -
accuracy: 0.7105
Epoch 80/200
57/57 [==============================] - ETA: 0s - loss: 0.5508 - accuracy:
0.7131WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 452ms/step - loss: 0.5508 -
accuracy: 0.7131
Epoch 81/200
57/57 [==============================] - ETA: 0s - loss: 0.5720 - accuracy:
0.7025WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 458ms/step - loss: 0.5720 -
accuracy: 0.7025
Epoch 82/200
57/57 [==============================] - ETA: 0s - loss: 0.5137 - accuracy:
0.7425WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 444ms/step - loss: 0.5137 -
accuracy: 0.7425
Epoch 83/200
57/57 [==============================] - ETA: 0s - loss: 0.5099 - accuracy:
0.7558WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 465ms/step - loss: 0.5099 -
accuracy: 0.7558
Epoch 84/200
57/57 [==============================] - ETA: 0s - loss: 0.5500 - accuracy:
0.7176WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 450ms/step - loss: 0.5500 -
accuracy: 0.7176
Epoch 85/200
57/57 [==============================] - ETA: 0s - loss: 0.5210 - accuracy:
0.7380WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 468ms/step - loss: 0.5210 -
accuracy: 0.7380
Epoch 86/200
57/57 [==============================] - ETA: 0s - loss: 0.5316 - accuracy:
0.7318WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 437ms/step - loss: 0.5316 -
accuracy: 0.7318
```

```
Epoch 87/200
57/57 [==============================] - ETA: 0s - loss: 0.5143 - accuracy:
0.7425WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 426ms/step - loss: 0.5143 -
accuracy: 0.7425
Epoch 88/200
57/57 [==============================] - ETA: 0s - loss: 0.4957 - accuracy:
0.7593WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 453ms/step - loss: 0.4957 -
accuracy: 0.7593
Epoch 89/200
57/57 [==============================] - ETA: 0s - loss: 0.4914 - accuracy:
0.7664WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 420ms/step - loss: 0.4914 -
accuracy: 0.7664
Epoch 90/200
57/57 [==============================] - ETA: 0s - loss: 0.4879 - accuracy:
0.7638WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 426ms/step - loss: 0.4879 -
accuracy: 0.7638
Epoch 91/200
57/57 [==============================] - ETA: 0s - loss: 0.4889 - accuracy:
0.7647WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 443ms/step - loss: 0.4889 -
accuracy: 0.7647
Epoch 92/200
57/57 [==============================] - ETA: 0s - loss: 0.4656 - accuracy:
0.7789WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 445ms/step - loss: 0.4656 -
accuracy: 0.7789
Epoch 93/200
57/57 [==============================] - ETA: 0s - loss: 0.4707 - accuracy:
0.7718WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 442ms/step - loss: 0.4707 -
accuracy: 0.7718
Epoch 94/200
57/57 [==============================] - ETA: 0s - loss: 0.4484 - accuracy:
0.8108WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 424ms/step - loss: 0.4484 -
accuracy: 0.8108
Epoch 95/200
57/57 [==============================] - ETA: 0s - loss: 0.4595 - accuracy:
0.7869WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 458ms/step - loss: 0.4595 -
accuracy: 0.7869
Epoch 96/200
57/57 [==============================] - ETA: 0s - loss: 0.4748 - accuracy:
0.7718WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 422ms/step - loss: 0.4748 -
accuracy: 0.7718
Epoch 97/200
57/57 [==============================] - ETA: 0s - loss: 0.4451 - accuracy:
0.7957WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 439ms/step - loss: 0.4451 -
accuracy: 0.7957
Epoch 98/200
57/57 [==============================] - ETA: 0s - loss: 0.4308 - accuracy:
0.8020WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 444ms/step - loss: 0.4308 -
accuracy: 0.8020
Epoch 99/200
57/57 [==============================] - ETA: 0s - loss: 0.4155 - accuracy:
0.8099WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 469ms/step - loss: 0.4155 -
accuracy: 0.8099
Epoch 100/200
57/57 [==============================] - ETA: 0s - loss: 0.4048 - accuracy:
0.8046WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 425ms/step - loss: 0.4048 -
accuracy: 0.8046
Epoch 101/200
57/57 [==============================] - ETA: 0s - loss: 0.3837 - accuracy:
0.8268WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 29s 506ms/step - loss: 0.3837 -
accuracy: 0.8268
Epoch 102/200
57/57 [==============================] - ETA: 0s - loss: 0.4237 - accuracy:
0.8046WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 28s 496ms/step - loss: 0.4237 -
accuracy: 0.8046
Epoch 103/200
57/57 [==============================] - ETA: 0s - loss: 0.4039 - accuracy:
0.8179WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 447ms/step - loss: 0.4039 -
accuracy: 0.8179
Epoch 104/200
57/57 [==============================] - ETA: 0s - loss: 0.3806 - accuracy:
0.8348WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 434ms/step - loss: 0.3806 -
accuracy: 0.8348
Epoch 105/200
57/57 [==============================] - ETA: 0s - loss: 0.4013 - accuracy:
0.8126WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 431ms/step - loss: 0.4013 -
accuracy: 0.8126
```

```
Epoch 106/200
57/57 [==============================] - ETA: 0s - loss: 0.3892 - accuracy: 
0.8206WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 467ms/step - loss: 0.3892 - 
accuracy: 0.8206
Epoch 107/200
57/57 [==============================] - ETA: 0s - loss: 0.3991 - accuracy: 
0.8268WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 424ms/step - loss: 0.3991 - 
accuracy: 0.8268
Epoch 108/200
57/57 [==============================] - ETA: 0s - loss: 0.3669 - accuracy: 
0.8321WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 448ms/step - loss: 0.3669 - 
accuracy: 0.8321
Epoch 109/200
57/57 [==============================] - ETA: 0s - loss: 0.3449 - accuracy: 
0.8517WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 442ms/step - loss: 0.3449 - 
accuracy: 0.8517
Epoch 110/200
57/57 [==============================] - ETA: 0s - loss: 0.3660 - accuracy: 
0.8321WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 28s 486ms/step - loss: 0.3660 - 
accuracy: 0.8321
Epoch 111/200
57/57 [==============================] - ETA: 0s - loss: 0.3239 - accuracy: 
0.8526WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 457ms/step - loss: 0.3239 - 
accuracy: 0.8526
Epoch 112/200
57/57 [==============================] - ETA: 0s - loss: 0.3350 - accuracy: 
0.8552WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 27s 481ms/step - loss: 0.3350 - 
accuracy: 0.8552
Epoch 113/200
57/57 [==============================] - ETA: 0s - loss: 0.3270 - accuracy: 
0.8632WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 430ms/step - loss: 0.3270 - 
accuracy: 0.8632
Epoch 114/200
57/57 [==============================] - ETA: 0s - loss: 0.3674 - accuracy: 
0.8393WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which 
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 436ms/step - loss: 0.3674 - 
accuracy: 0.8393
Epoch 115/200
57/57 [==============================] - ETA: 0s - loss: 0.3181 - accuracy: 
0.8597WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 423ms/step - loss: 0.3181 -
accuracy: 0.8597
Epoch 116/200
57/57 [==============================] - ETA: 0s - loss: 0.3481 - accuracy:
0.8375WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 428ms/step - loss: 0.3481 -
accuracy: 0.8375
Epoch 117/200
57/57 [==============================] - ETA: 0s - loss: 0.3192 - accuracy:
0.8686WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 421ms/step - loss: 0.3192 -
accuracy: 0.8686
Epoch 118/200
57/57 [==============================] - ETA: 0s - loss: 0.3201 - accuracy:
0.8615WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 421ms/step - loss: 0.3201 -
accuracy: 0.8615
Epoch 119/200
57/57 [==============================] - ETA: 0s - loss: 0.2874 - accuracy:
0.8694WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 430ms/step - loss: 0.2874 -
accuracy: 0.8694
Epoch 120/200
57/57 [==============================] - ETA: 0s - loss: 0.3060 - accuracy:
0.8783WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 400ms/step - loss: 0.3060 -
accuracy: 0.8783
Epoch 121/200
57/57 [==============================] - ETA: 0s - loss: 0.2825 - accuracy:
0.8757WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 397ms/step - loss: 0.2825 -
accuracy: 0.8757
Epoch 122/200
57/57 [==============================] - ETA: 0s - loss: 0.2736 - accuracy:
0.8828WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 366ms/step - loss: 0.2736 -
accuracy: 0.8828
Epoch 123/200
57/57 [==============================] - ETA: 0s - loss: 0.3097 - accuracy:
0.8766WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 405ms/step - loss: 0.3097 -
accuracy: 0.8766
Epoch 124/200
57/57 [==============================] - ETA: 0s - loss: 0.2874 - accuracy:
0.8828WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 388ms/step - loss: 0.2874 -
accuracy: 0.8828
```

```
Epoch 125/200
57/57 [==============================] - ETA: 0s - loss: 0.2644 - accuracy:
0.8863WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 382ms/step - loss: 0.2644 -
accuracy: 0.8863
Epoch 126/200
57/57 [==============================] - ETA: 0s - loss: 0.2529 - accuracy:
0.8917WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 421ms/step - loss: 0.2529 -
accuracy: 0.8917
Epoch 127/200
57/57 [==============================] - ETA: 0s - loss: 0.2570 - accuracy:
0.8917WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 385ms/step - loss: 0.2570 -
accuracy: 0.8917
Epoch 128/200
57/57 [==============================] - ETA: 0s - loss: 0.2908 - accuracy:
0.8721WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 393ms/step - loss: 0.2908 -
accuracy: 0.8721
Epoch 129/200
57/57 [==============================] - ETA: 0s - loss: 0.2453 - accuracy:
0.8979WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 393ms/step - loss: 0.2453 -
accuracy: 0.8979
Epoch 130/200
57/57 [==============================] - ETA: 0s - loss: 0.2597 - accuracy:
0.8925WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 390ms/step - loss: 0.2597 -
accuracy: 0.8925
Epoch 131/200
57/57 [==============================] - ETA: 0s - loss: 0.2472 - accuracy:
0.9076WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 404ms/step - loss: 0.2472 -
accuracy: 0.9076
Epoch 132/200
57/57 [==============================] - ETA: 0s - loss: 0.2694 - accuracy:
0.8996WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 429ms/step - loss: 0.2694 -
accuracy: 0.8996
Epoch 133/200
57/57 [==============================] - ETA: 0s - loss: 0.2662 - accuracy:
0.8899WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 434ms/step - loss: 0.2662 -
accuracy: 0.8899
Epoch 134/200
57/57 [==============================] - ETA: 0s - loss: 0.3109 - accuracy:
0.8792WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 425ms/step - loss: 0.3109 -
accuracy: 0.8792
Epoch 135/200
57/57 [==============================] - ETA: 0s - loss: 0.2409 - accuracy:
0.9076WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 447ms/step - loss: 0.2409 -
accuracy: 0.9076
Epoch 136/200
57/57 [==============================] - ETA: 0s - loss: 0.2320 - accuracy:
0.9032WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 424ms/step - loss: 0.2320 -
accuracy: 0.9032
Epoch 137/200
57/57 [==============================] - ETA: 0s - loss: 0.2225 - accuracy:
0.9156WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 426ms/step - loss: 0.2225 -
accuracy: 0.9156
Epoch 138/200
57/57 [==============================] - ETA: 0s - loss: 0.2412 - accuracy:
0.9041WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 452ms/step - loss: 0.2412 -
accuracy: 0.9041
Epoch 139/200
57/57 [==============================] - ETA: 0s - loss: 0.2138 - accuracy:
0.9130WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 440ms/step - loss: 0.2138 -
accuracy: 0.9130
Epoch 140/200
57/57 [==============================] - ETA: 0s - loss: 0.2575 - accuracy:
0.9059WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 430ms/step - loss: 0.2575 -
accuracy: 0.9059
Epoch 141/200
57/57 [==============================] - ETA: 0s - loss: 0.2278 - accuracy:
0.9094WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 439ms/step - loss: 0.2278 -
accuracy: 0.9094
Epoch 142/200
57/57 [==============================] - ETA: 0s - loss: 0.1992 - accuracy:
0.9325WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 427ms/step - loss: 0.1992 -
accuracy: 0.9325
Epoch 143/200
57/57 [==============================] - ETA: 0s - loss: 0.2151 - accuracy:
0.9147WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 25s 430ms/step - loss: 0.2151 -
accuracy: 0.9147
```

```
Epoch 144/200
57/57 [==============================] - ETA: 0s - loss: 0.2364 - accuracy:
0.9076WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 417ms/step - loss: 0.2364 -
accuracy: 0.9076
Epoch 145/200
57/57 [==============================] - ETA: 0s - loss: 0.1787 - accuracy:
0.9298WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 358ms/step - loss: 0.1787 -
accuracy: 0.9298
Epoch 146/200
57/57 [==============================] - ETA: 0s - loss: 0.2013 - accuracy:
0.9174WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 359ms/step - loss: 0.2013 -
accuracy: 0.9174
Epoch 147/200
57/57 [==============================] - ETA: 0s - loss: 0.1938 - accuracy:
0.9201WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 19s 339ms/step - loss: 0.1938 -
accuracy: 0.9201
Epoch 148/200
57/57 [==============================] - ETA: 0s - loss: 0.2095 - accuracy:
0.9139WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 343ms/step - loss: 0.2095 -
accuracy: 0.9139
Epoch 149/200
57/57 [==============================] - ETA: 0s - loss: 0.1701 - accuracy:
0.9352WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 358ms/step - loss: 0.1701 -
accuracy: 0.9352
Epoch 150/200
57/57 [==============================] - ETA: 0s - loss: 0.2032 - accuracy:
0.9245WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 353ms/step - loss: 0.2032 -
accuracy: 0.9245
Epoch 151/200
57/57 [==============================] - ETA: 0s - loss: 0.1895 - accuracy:
0.9218WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 354ms/step - loss: 0.1895 -
accuracy: 0.9218
Epoch 152/200
57/57 [==============================] - ETA: 0s - loss: 0.1949 - accuracy:
0.9210WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 29s 517ms/step - loss: 0.1949 -
accuracy: 0.9210
Epoch 153/200
57/57 [==============================] - ETA: 0s - loss: 0.1761 - accuracy:
0.9325WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 46s 806ms/step - loss: 0.1761 -
accuracy: 0.9325
Epoch 154/200
57/57 [==============================] - ETA: 0s - loss: 0.2756 - accuracy:
0.8988WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 45s 795ms/step - loss: 0.2756 -
accuracy: 0.8988
Epoch 155/200
57/57 [==============================] - ETA: 0s - loss: 0.2010 - accuracy:
0.9227WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 48s 837ms/step - loss: 0.2010 -
accuracy: 0.9227
Epoch 156/200
57/57 [==============================] - ETA: 0s - loss: 0.2119 - accuracy:
0.9165WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 33s 565ms/step - loss: 0.2119 -
accuracy: 0.9165
Epoch 157/200
57/57 [==============================] - ETA: 0s - loss: 0.1721 - accuracy:
0.9361WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 367ms/step - loss: 0.1721 -
accuracy: 0.9361
Epoch 158/200
57/57 [==============================] - ETA: 0s - loss: 0.2040 - accuracy:
0.9254WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 366ms/step - loss: 0.2040 -
accuracy: 0.9254
Epoch 159/200
57/57 [==============================] - ETA: 0s - loss: 0.1802 - accuracy:
0.9343WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 376ms/step - loss: 0.1802 -
accuracy: 0.9343
Epoch 160/200
57/57 [==============================] - ETA: 0s - loss: 0.1783 - accuracy:
0.9352WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 374ms/step - loss: 0.1783 -
accuracy: 0.9352
Epoch 161/200
57/57 [==============================] - ETA: 0s - loss: 0.1643 - accuracy:
0.9405WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 373ms/step - loss: 0.1643 -
accuracy: 0.9405
Epoch 162/200
57/57 [==============================] - ETA: 0s - loss: 0.1316 - accuracy:
0.9449WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 379ms/step - loss: 0.1316 -
accuracy: 0.9449
```

```
Epoch 163/200
57/57 [==============================] - ETA: 0s - loss: 0.1839 - accuracy:
0.9245WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 376ms/step - loss: 0.1839 -
accuracy: 0.9245
Epoch 164/200
57/57 [==============================] - ETA: 0s - loss: 0.1588 - accuracy:
0.9405WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 382ms/step - loss: 0.1588 -
accuracy: 0.9405
Epoch 165/200
57/57 [==============================] - ETA: 0s - loss: 0.1821 - accuracy:
0.9352WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 381ms/step - loss: 0.1821 -
accuracy: 0.9352
Epoch 166/200
57/57 [==============================] - ETA: 0s - loss: 0.1590 - accuracy:
0.9449WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 380ms/step - loss: 0.1590 -
accuracy: 0.9449
Epoch 167/200
57/57 [==============================] - ETA: 0s - loss: 0.1467 - accuracy:
0.9458WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 386ms/step - loss: 0.1467 -
accuracy: 0.9458
Epoch 168/200
57/57 [==============================] - ETA: 0s - loss: 0.1885 - accuracy:
0.9334WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 393ms/step - loss: 0.1885 -
accuracy: 0.9334
Epoch 169/200
57/57 [==============================] - ETA: 0s - loss: 0.1854 - accuracy:
0.9307WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 398ms/step - loss: 0.1854 -
accuracy: 0.9307
Epoch 170/200
57/57 [==============================] - ETA: 0s - loss: 0.1721 - accuracy:
0.9307WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 393ms/step - loss: 0.1721 -
accuracy: 0.9307
Epoch 171/200
57/57 [==============================] - ETA: 0s - loss: 0.1529 - accuracy:
0.9325WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 407ms/step - loss: 0.1529 -
accuracy: 0.9325
Epoch 172/200
57/57 [==============================] - ETA: 0s - loss: 0.1431 - accuracy:
0.9503WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 406ms/step - loss: 0.1431 -
accuracy: 0.9503
Epoch 173/200
57/57 [==============================] - ETA: 0s - loss: 0.1168 - accuracy:
0.9609WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 372ms/step - loss: 0.1168 -
accuracy: 0.9609
Epoch 174/200
57/57 [==============================] - ETA: 0s - loss: 0.1526 - accuracy:
0.9369WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 362ms/step - loss: 0.1526 -
accuracy: 0.9369
Epoch 175/200
57/57 [==============================] - ETA: 0s - loss: 0.1409 - accuracy:
0.9440WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 358ms/step - loss: 0.1409 -
accuracy: 0.9440
Epoch 176/200
57/57 [==============================] - ETA: 0s - loss: 0.1286 - accuracy:
0.9520WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 372ms/step - loss: 0.1286 -
accuracy: 0.9520
Epoch 177/200
57/57 [==============================] - ETA: 0s - loss: 0.1753 - accuracy:
0.9334WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 371ms/step - loss: 0.1753 -
accuracy: 0.9334
Epoch 178/200
57/57 [==============================] - ETA: 0s - loss: 0.1302 - accuracy:
0.9512WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 368ms/step - loss: 0.1302 -
accuracy: 0.9512
Epoch 179/200
57/57 [==============================] - ETA: 0s - loss: 0.1262 - accuracy:
0.9467WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 371ms/step - loss: 0.1262 -
accuracy: 0.9467
Epoch 180/200
57/57 [==============================] - ETA: 0s - loss: 0.1602 - accuracy:
0.9272WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 369ms/step - loss: 0.1602 -
accuracy: 0.9272
Epoch 181/200
57/57 [==============================] - ETA: 0s - loss: 0.1905 - accuracy:
0.9263WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 375ms/step - loss: 0.1905 -
accuracy: 0.9263
```

```
Epoch 182/200
57/57 [==============================] - ETA: 0s - loss: 0.1250 - accuracy:
0.9467WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 21s 369ms/step - loss: 0.1250 -
accuracy: 0.9467
Epoch 183/200
57/57 [==============================] - ETA: 0s - loss: 0.1202 - accuracy:
0.9600WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 383ms/step - loss: 0.1202 -
accuracy: 0.9600
Epoch 184/200
57/57 [==============================] - ETA: 0s - loss: 0.1339 - accuracy:
0.9520WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 381ms/step - loss: 0.1339 -
accuracy: 0.9520
Epoch 185/200
57/57 [==============================] - ETA: 0s - loss: 0.1202 - accuracy:
0.9512WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 24s 428ms/step - loss: 0.1202 -
accuracy: 0.9512
Epoch 186/200
57/57 [==============================] - ETA: 0s - loss: 0.1349 - accuracy:
0.9556 WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 3025s 54s/step - loss: 0.1349 -
accuracy: 0.9556
Epoch 187/200
57/57 [==============================] - ETA: 0s - loss: 0.1997 - accuracy:
0.9263WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 389ms/step - loss: 0.1997 -
accuracy: 0.9263
Epoch 188/200
57/57 [==============================] - ETA: 0s - loss: 0.1505 - accuracy:
0.9405WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 22s 383ms/step - loss: 0.1505 -
accuracy: 0.9405
Epoch 189/200
57/57 [==============================] - ETA: 0s - loss: 0.1039 - accuracy:
0.9654WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 26s 461ms/step - loss: 0.1039 -
accuracy: 0.9654
Epoch 190/200
57/57 [==============================] - ETA: 0s - loss: 0.1636 - accuracy:
0.9414WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 399ms/step - loss: 0.1636 -
accuracy: 0.9414
Epoch 191/200
57/57 [==============================] - ETA: 0s - loss: 0.1330 - accuracy:
0.9432WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
```

```
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 20s 355ms/step - loss: 0.1330 -
accuracy: 0.9432
Epoch 192/200
57/57 [==============================] - ETA: 0s - loss: 0.1473 - accuracy:
0.9467WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 23s 407ms/step - loss: 0.1473 -
accuracy: 0.9467
Epoch 193/200
57/57 [==============================] - ETA: 0s - loss: 0.1188 - accuracy:
0.9583WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 35s 622ms/step - loss: 0.1188 -
accuracy: 0.9583
Epoch 194/200
57/57 [==============================] - ETA: 0s - loss: 0.1377 - accuracy:
0.9512WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 51s 888ms/step - loss: 0.1377 -
accuracy: 0.9512
Epoch 195/200
57/57 [==============================] - ETA: 0s - loss: 0.1219 - accuracy:
0.9618WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 49s 855ms/step - loss: 0.1219 -
accuracy: 0.9618
Epoch 196/200
57/57 [==============================] - ETA: 0s - loss: 0.1239 - accuracy:
0.9529WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 51s 903ms/step - loss: 0.1239 -
accuracy: 0.9529
Epoch 197/200
57/57 [==============================] - ETA: 0s - loss: 0.1485 - accuracy:
0.9547WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 48s 847ms/step - loss: 0.1485 -
accuracy: 0.9547
Epoch 198/200
57/57 [==============================] - ETA: 0s - loss: 0.1002 - accuracy:
0.9654WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 45s 793ms/step - loss: 0.1002 -
accuracy: 0.9654
Epoch 199/200
57/57 [==============================] - ETA: 0s - loss: 0.1515 - accuracy:
0.9440WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 53s 931ms/step - loss: 0.1515 -
accuracy: 0.9440
Epoch 200/200
57/57 [==============================] - ETA: 0s - loss: 0.1066 - accuracy:
0.9574WARNING:tensorflow:Early stopping conditioned on metric `val_loss` which
is not available. Available metrics are: loss,accuracy
57/57 [==============================] - 49s 867ms/step - loss: 0.1066 -
accuracy: 0.9574
```

```
Accuracy:  93.3747410774231
```

SVM CLASSIFIER

In [52]:

```
Xtrain.shape
```

Out[52]:

```
(1126, 128, 128, 3)
```

In [53]:

```
Xtest.shape
```

Out[53]:

```
(483, 128, 128, 3)
```

In [54]:

```
Xtrain_SVC = Xtrain.reshape(1126, 128 * 128 * 3)
Xtest_SVC = Xtest.reshape(483, 128 * 128 * 3)
```

In [55]:

```
model_SVC = SVC(C = 1,kernel='poly',gamma = 'auto')
model_SVC.fit(Xtrain_SVC, Ytrain)
```

Out[55]:

```
SVC(C=1, gamma='auto', kernel='poly')
```

In [56]:

```
prediction = model_SVC.predict(Xtest_SVC)
accuracy = model_SVC.score(Xtest_SVC , Ytest)
```

In [57]:

```
train_accuracy = model_SVC.score(Xtrain_SVC, Ytrain)
print("Train_Accuracy", train_accuracy * 100)
```

```
Train_Accuracy 100.0
```

In [58]:

```
print("Test_Accuracy ",accuracy * 100)
```

```
Test_Accuracy  93.16770186335404
```

Confusion Matrix

In [59]:

```
# cm = confusion_matrix(y_true = Ytest, y_pred = np.argmax(predictions,axis=-
1))
cm = confusion_matrix(Ytest , (prediction > 0.75) * 1)
```

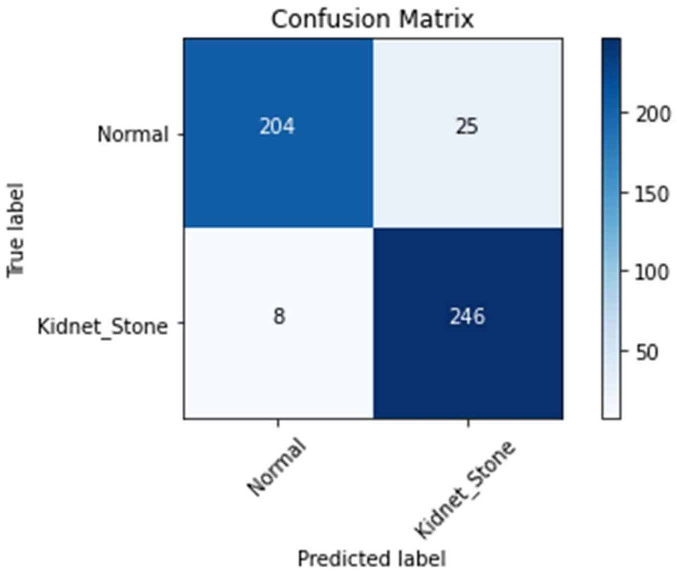
In [60]:

```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
            horizontalalignment="center",
            color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

In [61]:

```
cm_plot_labels = ['Normal','Kidnet_Stone']
plot_confusion_matrix(cm = cm, classes = cm_plot_labels, title = 'Confusion
Matrix')
```

```
Confusion matrix, without normalization
[[204  25]
 [  8 246]]
```



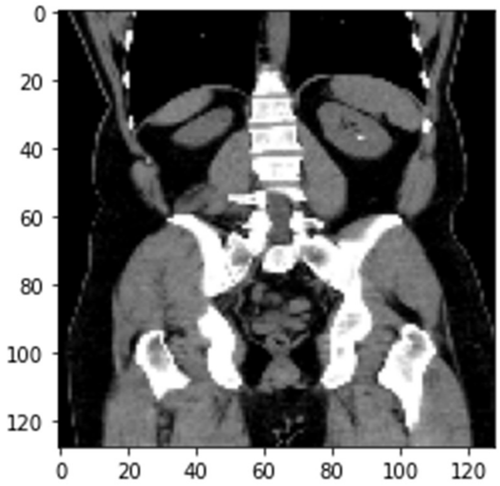
In [62]:

```
def image_prediction(img):
    plt.imshow(img)
    img1=img.reshape(1,128,128,3)
    predict = model3.predict(img1)
    if ((predict > 0.75) * 1):
        print("The condition is normal and stable.")
    else:
        print("Person is having kidney stone(s)")
```

In [63]:

```
image_prediction(Xtest[150])
```

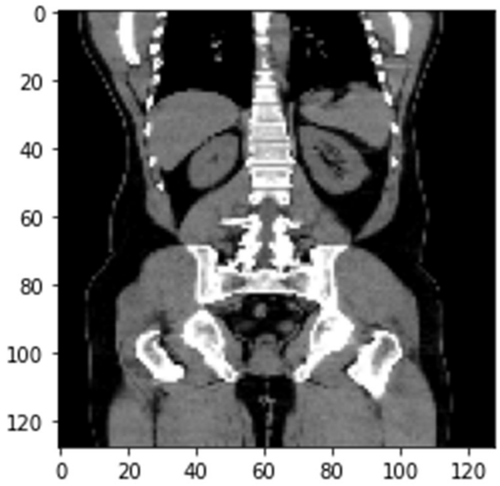
```
1/1 [==============================] - 0s 32ms/step
Person is having kidney stone(s)
```



In [64]:

```
image_prediction(Xtest[50])
```

```
1/1 [==============================] - 0s 34ms/step
The condition is normal and stable.
```



In [ ]:

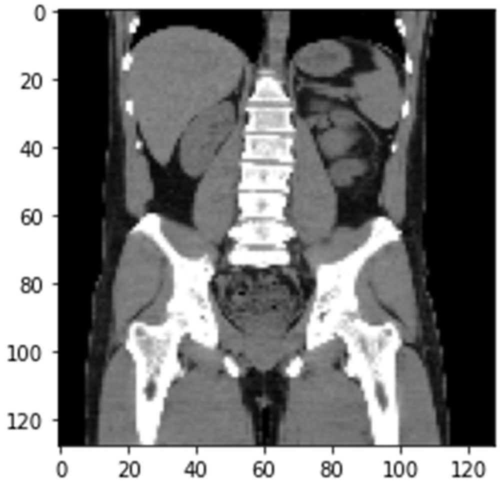
```
!pip install split-folders
```

In [74]:

```
def image_prediction_KMeans(img):
    img = img.reshape(128,128,3)
    plt.imshow(img)
    img1 = img.reshape(1, 128 * 128 * 3)
    predict = kmeans_model.predict(img1)
    print(predict)
    if (predict):
        print("Person is having kidney stone(s)")
    else:
        print("The condition is normal and stable.")
```

In [77]:

```
path_2 = "C:/Users/prajwal mr/Downloads/archive/CT_SCAN/Normal/N1.png"
image_array_2 = cv2.imread(path_2)
new_image_2 = cv2.resize(image_array_2,(128,128))
plt.imshow(new_image_2)
plt.show()
```



In [95]:

```
model3.save('kidney_stones_model.h5')
```

In [ ]:

4. Conclusion

In conclusion, a kidney stone detection project utilizing digital image processing (DIP) techniques can be an interesting research endeavor. However, it is important to note that DIP-based kidney stone detection is not commonly used in clinical settings for diagnosing kidney stones. Medical imaging techniques such as ultrasound, X-ray, or computed tomography (CT) scans are the established methods for kidney stone detection and diagnosis

Ultimately, the aim of a kidney stone detection project using DIP would be to explore the feasibility of utilizing image processing techniques to aid in the identification and diagnosis of kidney stones. While it may not replace established medical imaging methods, it could potentially contribute to the development of supplementary tools or assist in early identification of kidney stones, enhancing patient care and treatment.