```
In [1]:
            # pip install tensorflow
          2
In [2]:
            # pip install matplotlib
          2
In [3]:
            # pip install opencv-python
          2
            # pip install numpy
In [4]:
          1
          2
In [5]:
          1 import tensorflow as tf
          2 import matplotlib.pyplot as plt
          3 import cv2
          4 import os
          5 import numpy as np
          6 | from tensorflow.keras.preprocessing.image import ImageDataGenerator
          7 from tensorflow.keras.preprocessing import image
            from tensorflow.keras.optimizers import RMSprop
        E:\AnacondaDATASCIENCE\lib\site-packages\scipy\__init__.py:155: UserWarnin
        g: A NumPy version >=1.18.5 and <1.25.0 is required for this version of Sci
        Py (detected version 1.26.4
          warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
In [6]:
            img=image.load_img("/content/drive/MyDrive/colab/basedata/train/Loam Soit
        FileNotFoundError
                                                   Traceback (most recent call last)
        ~\AppData\Local\Temp\ipykernel 13068\1106985946.py in <module>
        ----> 1 img=image.load img("/content/drive/MyDrive/colab/basedata/train/Loa
        m Soil/10.jpeg")
        E:\AnacondaDATASCIENCE\lib\site-packages\keras\src\utils\image_utils.py in
        load_img(path, color_mode, target_size, interpolation, keep_aspect_ratio)
            233
                        if isinstance(path, pathlib.Path):
            234
                             path = str(path.resolve())
                        with open(path, "rb") as f:
        --> 235
            236
                            img = pil_image.open(io.BytesIO(f.read()))
            237
                    else:
        FileNotFoundError: [Errno 2] No such file or directory: '/content/drive/MyD
        rive/colab/basedata/train/Loam Soil/10.jpeg'
In [ ]:
            plt.imshow(img)
In [ ]:
          1 cv2.imread("/content/drive/MyDrive/colab/basedata/train/Loam Soil/10.jpe
In [ ]:
          1 train =ImageDataGenerator(rescale=1/255)
          2
            validation =ImageDataGenerator(rescale=1/255)
          3
```

```
In [ ]:
                                       train_dataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/colab/bataset=train.flow_from_directory('/content/drive/MyDrive/Colab/bataset=train.flow_from_directory('/content/drive/MyDrive/Colab/bataset=train.flow_from_directory('/content/drive/MyDriv
                               2
                                                                                                                                                                 target_size=(200,200),
                               3
                                                                                                                                                                 batch_size=3,
                              4
                                                                                                                                                                 class mode='binary')
                               5
                                       validation_dataset=validation.flow_from_directory('/content/drive/MyDrive)
                              6
                                                                                                                                                                 target_size=(200,200),
                              7
                                                                                                                                                                 batch size=3,
                                                                                                                                                                 class_mode='binary')
                              8
                               9
In [ ]:
                                    train_dataset.class_indices
In [ ]:
                               1
                                      train_dataset.classes
In [ ]:
                                       model=tf.keras.models.Sequential([tf.keras.layers.Conv2D(16,(3,3),activa
                               2
                                                                                                                                            tf.keras.layers.MaxPool2D(2,2),
                               3
                              4
                                                                                                                                            tf.keras.layers.Conv2D(32,(3,3),activat
                              5
                                                                                                                                              tf.keras.layers.MaxPool2D(2,2),
                              6
                               7
                              8
                                                                                                                                              tf.keras.layers.Conv2D(64,(3,3),activa
                              9
                                                                                                                                              tf.keras.layers.MaxPool2D(2,2),
                            10
                                                                                                                                              tf.keras.layers.Flatten(),
                            11
                            12
                            13
                                                                                                                                              tf.keras.layers.Dense(512,activation=
                            14
                            15
                                                                                                                                                 tf.keras.layers.Dense(1,activation='s
                            16
                            17
                            18
                            19
                                                                                                                                            ])
In [ ]:
                                       model.compile(loss ='binary_crossentropy',
                               2
                                                                                 optimizer= RMSprop(lr=0.001),
                               3
                                                                                 metrics=['accuracy'])
                                       model_fit=model.fit(train_dataset,
In [ ]:
                              1
                                                                                                    steps_per_epoch=3,
                               2
                               3
                                                                                                    epochs=10, #epochs are iterations you have to give to
                                                                                                    validation data=validation dataset)
                               4
```

```
In [ ]:
             dir_path='/content/drive/MyDrive/colab/basedata/test'
          2
                 for i in os.listdir(dir_path):
                 img=image.load_img(dir_path+'//'+i,target_size=(200,200))
          3
          4
                       plt.imshow(img)
          5
                       plt.show()
          6
                     X=image.img_to_array(img)
          7
                       X=np.expand_dims(X,axis=0)
          8
                     images=np.vstack([X])
          9
                           val=model.predict(images)
         10
                           if val[0][0] < 0.5:</pre>
         11
                             print("Loam Soil")
         12
                               else:
                                  print("Sandy Soil")
         13
```