```
!pip install kaggle
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5.16)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.2)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.7)
```

```
# configuring the path of Kaggle.json file
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

```
#importing dataset
!kaggle datasets download -d vuppalaadithyasairam/bone-fracture-detection-using-xrays
!kaggle datasets download -d bebofekry/bone-fracture-atlas-ds-compressed
```

```
Downloading bone-fracture-detection-using-xrays.zip to /content
100% 172M/172M [00:06<00:00, 38.5MB/s]
100% 172M/172M [00:06<00:00, 29.0MB/s]
Downloading bone-fracture-atlas-ds-compressed.zip to /content
 99% 123M/125M [00:04<00:00, 33.9MB/s]
100% 125M/125M [00:04<00:00, 28.1MB/s]
```

```
!ls
```

```
bone-fracture-atlas-ds-compressed.zip    kaggle.json
bone-fracture-detection-using-xrays.zip  sample_data
```

```
#extract the file
from zipfile import ZipFile

dataset1 ='/content/bone-fracture-atlas-ds-compressed.zip'

dataset2 = '/content/bone-fracture-detection-using-xrays.zip'

with ZipFile(dataset1, 'r') as zip:
  zip.extractall()
  print('The dataset is extracted')

with ZipFile(dataset2, 'r') as zip:
  zip.extractall()
  print('The dataset is extracted')
```

```
The dataset is extracted
The dataset is extracted
```

```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import cv2
from google.colab.patches import cv2_imshow
from PIL import Image
from sklearn.model_selection import train_test_split
```

```
fractured_files1=os.listdir('/content/archive (6)/train/fractured')
fractured_files2=os.listdir('/content/FracAtlas/FracAtlas/images/Train/Fractured')
fractured_files=fractured_files1+fractured_files2
print(fractured_files[0:5])
print(fractured_files[-5:])
```

```
['118-rotated3-rotated3.jpg', '18-rotated2-rotated2.jpg', '72-rotated2-rotated2-rotated2.jpg', '32-rotated2-rotated2-rotated1.jpg',
['IMG0002565.jpg', 'IMG0003767.jpg', 'IMG0004150.jpg', 'IMG0003332.jpg', 'IMG0002382.jpg']
```

```
not_fractured_files1=os.listdir('/content/archive (6)/train/not fractured')
not_fractured_files2=os.listdir('/content/FracAtlas/FracAtlas/images/Train/Non_fractured')
not_fractured_files=not_fractured_files1+not_fractured_files2
print(fractured_files[0:5])
print(fractured_files[-5:])
```

```
['118-rotated3-rotated3.jpg', '18-rotated2-rotated2.jpg', '72-rotated2-rotated2-rotated2.jpg', '32-rotated2-rotated2-rotated1.jpg',
['IMG0002565.jpg', 'IMG0003767.jpg', 'IMG0004150.jpg', 'IMG0003332.jpg', 'IMG0002382.jpg']
```

```
print('Number of with fractured images:', len(fractured_files))
print('Number of not fractured images:', len(not_fractured_files))
```

```
Number of with fractured images: 5197
Number of not fractured images: 5275
```

Creating Labels

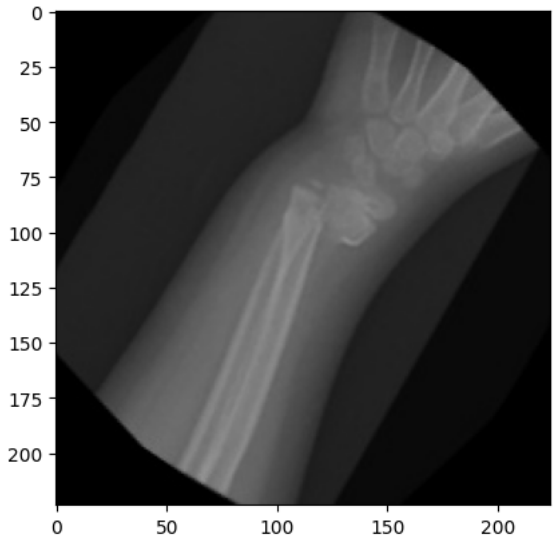
- 0 -> Not Fractured
- 1 -> Fractured

```
# create the labels

fractured_labels = [1]*5197

not_fractured_labels = [0]*5275


# calculating total labels

print(len(fractured_labels))
print(len(not_fractured_labels))

labels = fractured_labels+not_fractured_labels

print(len(labels))
print(labels[0:5])
print(labels[-5:])
```

```
5197
5275
10472
[1, 1, 1, 1, 1]
[0, 0, 0, 0, 0]
```

Displaying The Images

```
#display fractured image
img = mpimg.imread('/content/archive (6)/train/fractured/118-rotated3-rotated2-rotated3.jpg')
imgplot = plt.imshow(img)
plt.show()
```



```
# displaying not fractured image
img = mpimg.imread('/content/archive (6)/train/not fractured/25-rotated3-rotated3-rotated1.jpg')
imgplot = plt.imshow(img)
plt.show()
```

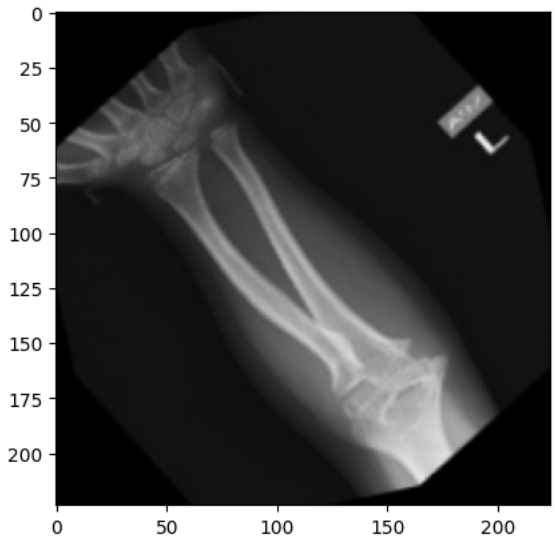


Image Processing

Resizing Images

Converting images into numpy arrays

```
data=[]

fractured_path1='/content/archive (6)/train/fractured/'
fractured_path2='/content/FracAtlas/FracAtlas/images/Train/Fractured/'

for img_file in fractured_files1:
  image = Image.open(fractured_path1 + img_file)
  image = image.resize((128,128))
  image = image.convert('RGB')
  image = np.array(image)
  data.append(image)

for img_file in fractured_files2:
  image = Image.open(fractured_path2 + img_file)
  image = image.resize((128,128))
  image = image.convert('RGB')
  image = np.array(image)
  data.append(image)


not_fractured_path1='/content/archive (6)/train/not fractured/'
not_fractured_path2='/content/FracAtlas/FracAtlas/images/Train/Non_fractured/'

for img_file in not_fractured_files1:
  image = Image.open(not_fractured_path1 + img_file)
  image = image.resize((128,128))
  image = image.convert('RGB')
  image = np.array(image)
  data.append(image)

for img_file in not_fractured_files2:
  image = Image.open(not_fractured_path2 + img_file)
  image = image.resize((128,128))
  image = image.convert('RGB')
  image = np.array(image)
  data.append(image)


type(data)
```

```
list
```

```
len(data)
```

```
10472
```

```
data[0]
```

```
ndarray (128, 128, 3) show data
```

```
type(data[0])
```

```
numpy.ndarray
```

```
data[0].shape
```

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

```
# converting image list and label list to numpy arrays

X = np.array(data)
Y = np.array(labels)
```

```
type(X)
```

```
numpy.ndarray
```

```
type(Y)
```

```
numpy.ndarray
```

```
print(X.shape)
print(Y.shape)
```

```
(10472, 128, 128, 3)
(10472,)
```

```
print(Y)
```

```
[1 1 1 ... 0 0 0]
```

Train Test Split

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
print(X.shape, X_train.shape, X_test.shape)
```

```
(10472, 128, 128, 3) (8377, 128, 128, 3) (2095, 128, 128, 3)
```

```
# scaling the data

X_train_scaled = X_train/255

X_test_scaled = X_test/255
```

```
X_train[0]
```

```
ndarray (128, 128, 3) show data
```

```
X_train_scaled[0]
```

```
array([[[0.3254902 , 0.3254902 , 0.3254902 ],
        [0.0627451 , 0.0627451 , 0.0627451 ],
        [0.10196078, 0.10196078, 0.10196078],
        ...,
        [0.11372549, 0.11372549, 0.11372549],
        [0.16862745, 0.16862745, 0.16862745],
```

```
        [0.30980392, 0.30980392, 0.30980392]],

       [[0.08627451, 0.08627451, 0.08627451],
        [0.07843137, 0.07843137, 0.07843137],
        [0.10196078, 0.10196078, 0.10196078],
        ...,
        [0.07058824, 0.07058824, 0.07058824],
        [0.0745098 , 0.0745098 , 0.0745098 ],
        [0.10980392, 0.10980392, 0.10980392]],

       [[0.0745098 , 0.0745098 , 0.0745098 ],
        [0.0745098 , 0.0745098 , 0.0745098 ],
        [0.09803922, 0.09803922, 0.09803922],
        ...,
        [0.10588235, 0.10588235, 0.10588235],
        [0.10980392, 0.10980392, 0.10980392],
        [0.14509804, 0.14509804, 0.14509804]],

       ...,

       [[0.90980392, 0.90980392, 0.90980392],
        [0.90980392, 0.90980392, 0.90980392],
        [0.90980392, 0.90980392, 0.90980392],
        ...,
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235]],

       [[0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        ...,
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235]],

       [[0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        ...,
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235],
        [0.90588235, 0.90588235, 0.90588235]]])
```

Convolution Neural Network - CNN

```
import tensorflow as tf
from tensorflow import keras


num_of_classes = 2

model = keras.Sequential()

model.add(keras.layers.Conv2D(32, kernel_size=(3,3), activation='relu', input_shape=(128,128,3)))
model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))


model.add(keras.layers.Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(keras.layers.MaxPooling2D(pool_size=(2,2)))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(128, activation='relu'))
model.add(keras.layers.Dropout(0.5))

model.add(keras.layers.Dense(64, activation='relu'))
model.add(keras.layers.Dropout(0.5))


model.add(keras.layers.Dense(num_of_classes, activation='sigmoid'))


# compile the neural network
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['acc'])


# training the neural network
history = model.fit(X_train_scaled, Y_train, validation_split=0.1, epochs=5)
```

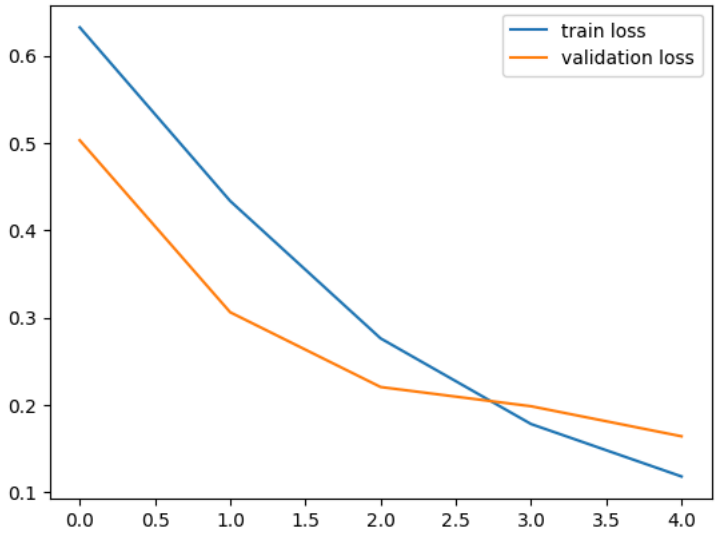
```
Epoch 1/5
236/236 [==============================] - 194s 817ms/step - loss: 0.6322 - acc: 0.6346 - val_loss: 0.5030 - val_acc: 0.7506
Epoch 2/5
236/236 [==============================] - 190s 804ms/step - loss: 0.4335 - acc: 0.7989 - val_loss: 0.3061 - val_acc: 0.8783
```

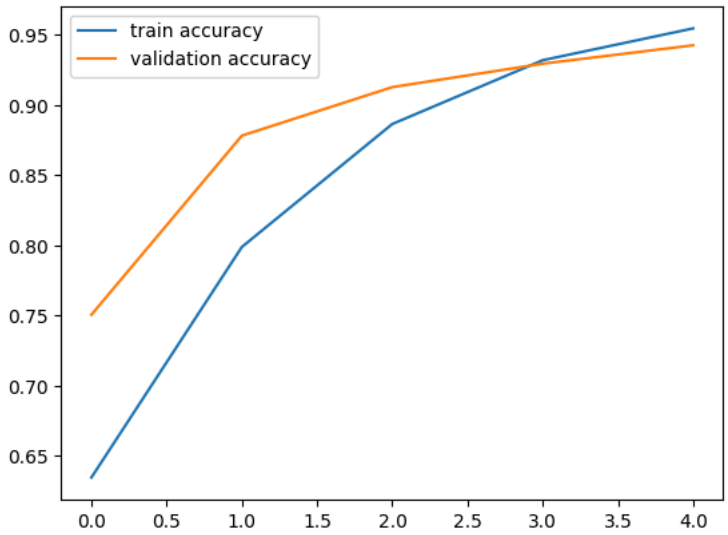
```
Epoch 3/5
236/236 [==============================] - 190s 808ms/step - loss: 0.2761 - acc: 0.8866 - val_loss: 0.2205 - val_acc: 0.9129
Epoch 4/5
236/236 [==============================] - 192s 812ms/step - loss: 0.1781 - acc: 0.9321 - val_loss: 0.1985 - val_acc: 0.9296
Epoch 5/5
236/236 [==============================] - 181s 767ms/step - loss: 0.1181 - acc: 0.9548 - val_loss: 0.1641 - val_acc: 0.9427
```

```
# Evaluation of Accuracy

loss, accuracy = model.evaluate(X_test_scaled, Y_test)
print('Test Accuracy =', accuracy)
```

```
66/66 [==============================] - 13s 202ms/step - loss: 0.1846 - acc: 0.9346
Test Accuracy = 0.9346061944961548
```

```
h=history

# plot the loss value
plt.plot(h.history['loss'], label='train loss')
plt.plot(h.history['val_loss'], label='validation loss')
plt.legend()
plt.show()

# plot the accuracy value
plt.plot(h.history['acc'], label='train accuracy')
plt.plot(h.history['val_acc'], label='validation accuracy')
plt.legend()
plt.show()
```






```
# Input from User and Prediction

input_image_path = input('Path of the image to be predicted: ')

input_image = cv2.imread(input_image_path)

cv2_imshow(input_image)
```

```
input_image_resized = cv2.resize(input_image, (128,128))

input_image_scaled = input_image_resized/255

input_image_reshaped = np.reshape(input_image_scaled, [1,128,128,3])

input_prediction = model.predict(input_image_reshaped)

print(input_prediction)


input_pred_label = np.argmax(input_prediction)

print(input_pred_label)


if input_pred_label == 1:

  print('The bone is fractured')

else:

  print('The bone is not fractured')
```

```
Path of the image to be predicted: /content/FracAtlas/FracAtlas/images/Train/Fractured/IMG0000058.jpg
```


```
1/1 [==============================] - 0s 141ms/step
[[3.7377598e-04 9.9065489e-01]]
1
The bone is fractured
```

```
!pip install streamlit
```

```
Collecting streamlit
  Downloading streamlit-1.33.0-py2.py3-none-any.whl (8.1 MB)
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.1/8.1 MB 22.0 MB/s eta 0:00:00
Requirement already satisfied: altair<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.2.2)
Requirement already satisfied: blinker<2,>=1.0.0 in /usr/lib/python3/dist-packages (from streamlit) (1.4)
Requirement already satisfied: cachetools<6,>=4.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (5.3.3)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.1.7)
Requirement already satisfied: numpy<2,>=1.19.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (1.25.2)
Requirement already satisfied: packaging<25,>=16.8 in /usr/local/lib/python3.10/dist-packages (from streamlit) (24.0)
Requirement already satisfied: pandas<3,>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.0.3)
Requirement already satisfied: pillow<11,>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (9.4.0)
Requirement already satisfied: protobuf<5,>=3.20 in /usr/local/lib/python3.10/dist-packages (from streamlit) (3.20.3)
Requirement already satisfied: pyarrow>=7.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (14.0.2)
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.10/dist-packages (from streamlit) (2.31.0)
Requirement already satisfied: rich<14,>=10.14.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (13.7.1)
Requirement already satisfied: tenacity<9,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (8.2.3)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in /usr/local/lib/python3.10/dist-packages (from streamlit) (4.11.0)
Collecting gitpython!=3.1.19,<4,>=3.0.7 (from streamlit)
  Downloading GitPython-3.1.43-py3-none-any.whl (207 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.3/207.3 kB 18.6 MB/s eta 0:00:00
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
  Downloading pydeck-0.8.1b0-py2.py3-none-any.whl (4.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.8/4.8 MB 60.6 MB/s eta 0:00:00
Requirement already satisfied: tornado<7,>=6.0.3 in /usr/local/lib/python3.10/dist-packages (from streamlit) (6.3.3)
Collecting watchdog>=2.1.5 (from streamlit)
  Downloading watchdog-4.0.0-py3-none-manylinux2014_x86_64.whl (82 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 83.0/83.0 kB 9.5 MB/s eta 0:00:00
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.4)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (3.1.3)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (4.19.2)
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6,>=4.0->streamlit) (0.12.1)
Collecting gitdb<5,>=4.0.1 (from gitpython!=3.1.19,<4,>=3.0.7->streamlit)
  Downloading gitdb-4.0.11-py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 kB 6.7 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3,>=1.3.0->streamlit) (2024.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streaml
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.27->streamlit) (2
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0->streamlit)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich<14,>=10.14.0->streamli
Collecting smmap<6,>=3.0.1 (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit)
  Downloading smmap-5.0.1-py3-none-any.whl (24 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->altair<6,>=4.0->streamlit)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->stre
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6,>=4.0->str
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich<14,>=10.14.0
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas<3,>=1.3.0->
Installing collected packages: watchdog, smmap, pydeck, gitdb, gitpython, streamlit
Successfully installed gitdb-4.0.11 gitpython-3.1.43 pydeck-0.8.1b0 smmap-5.0.1 streamlit-1.33.0 watchdog-4.0.0
```

```
import streamlit as st
import requests

# Define endpoint URL
ENDPOINT_URL = 'http://<ngrok_url>/<endpoint>'

# Define Streamlit UI components
st.title('Fracture Detection App')

uploaded_file = st.file_uploader("Choose an X-ray image...", type=["jpg", "jpeg", "png"])

if uploaded_file is not None:
    st.image(uploaded_file, caption='Uploaded X-ray Image', use_column_width=True)
    if st.button('Detect Fracture'):
        files = {'file': uploaded_file.getvalue()}
```