**REPORT**

**AIM:**

Creating a report on 3D classification and segmentation in the context of Computed Tomography (CT) involves explaining the importance, methods, and applications of these techniques in the field of medical imaging.

**PROCEDURE:**

\*Collect CT scan data of the patient's body, ensuring high-resolution 3D images.

\*Remove noise, correct artifacts, and enhance image quality to prepare data for analysis.

\*Employ machine learning algorithms to categorize structures (e.g.,organs, tumors) within the 3D image data.

\* Use advanced techniques to separate and label distinct regions, enabling precise visualization and measurements.

**\***Generate a detailed CT report with classified and segmented findings, aiding diagnosis and treatment planning for healthcare professionals.

**U-Net :**

It is a convolutional neural network (CNN) architecture designed for image segmentation tasks, particularly in medical imaging. It's named "U-Net" because of its U-shaped architecture, where it has a contracting path to capture context and an expansive path for precise localization.

**Pre-processing:** The process begins with pre-processing to enhance the quality of CT images. This may involve noise reduction, contrast adjustment, and the removal of artifacts.

**Region of Interest (ROI) Selection:** The radiologist or automated system defines the region of interest within the CT volume where classification and segmentation will be performed. This step narrows down the analysis to specific areas, such as a particular organ or tissue.

**Classification:** For 3D classification, machine learning algorithms are applied to categorize structures within the ROI. These algorithms can include convolutional neural networks (CNNs) and other deep learning methods. Classification determines the type of structure present, such as identifying different organs or abnormalities.

**Segmentation:** After classification, segmentation methods are used to delineate and label structures within the ROI. This step partitions the 3D volume into distinct regions corresponding to different structures, allowing precise measurements and visualization.

**Post-processing:** Post-processing techniques are applied to refine the segmented regions, smooth boundaries, and correct errors that may have occurred during classification and segmentation.

**Visualization and Reporting:** The segmented 3D data can be visualized in 3D renderings, cross-sectional slices, or other formats to assist in diagnosis. A detailed report is often generated, summarizing the findings and measurements.

**IMPLEMENTATION:**

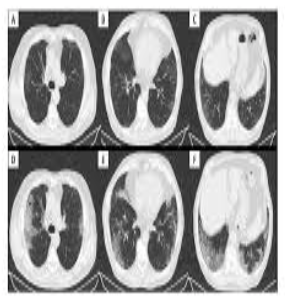
from IPython.display import Image

Image(url = 'https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcQc1dsRByBtIHIPIHDTDkSZvXblWD7EU9YRoEJS4KmZWPdeNDaB',width=400,height=400)



from IPython.display import Image

Image(url = 'https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcT9H9fUX-umZtjalXLoelye60N4tvM273Pe7R21nOFZdrvP4GCV',width=400,height=400)



import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn import feature\_extraction, linear\_model, model\_selection, preprocessing

import plotly.graph\_objs as go

import plotly.offline as py

import plotly.express as px

# Input data files are available in the "../input/" directory.

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os

for dirname, \_, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

# Any results you write to the current directory are saved as output.



nRowsRead = 1000 # specify 'None' if want to read whole file

df = pd.read\_csv('../input/cusersmarildownloadsctcsv/ct.csv', delimiter=';', encoding = "ISO-8859-1", nrows = nRowsRead)

df.dataframeName = 'ct.csv'

nRow, nCol = df.shape

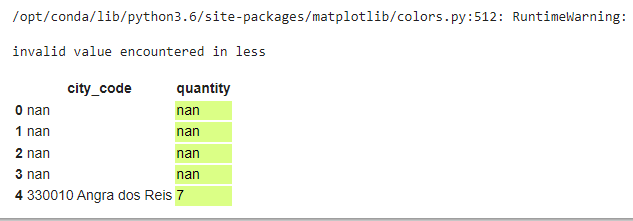
print(f'There are {nRow} rows and {nCol} columns')



from IPython.display import Image

Image(url = 'https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcTd3FY0zJNrZQApjJHGnH-nctY4w10Wrwy3-nZ34duhaOyY3\_RP',width=400,height=400)

df.head().style.background\_gradient(cmap='Wistia')



cnt\_srs = df['quantity'].value\_counts().head()

trace = go.Bar(

    y=cnt\_srs.index[::-1],

    x=cnt\_srs.values[::-1],

    orientation = 'h',

    marker=dict(

        color=cnt\_srs.values[::-1],

        colorscale = 'Greens',

        reversescale = True

    ),

)

layout = dict(

    title='CT Scan Distribution',

    )

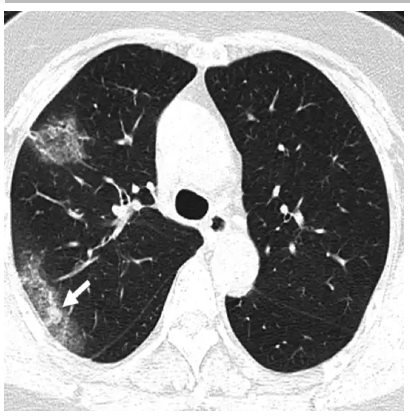
data = [trace]

fig = go.Figure(data=data, layout=layout)

py.iplot(fig, filename="quantity")

from IPython.display import Image

Image(url = 'https://www.businessinsider.in/thumb/msid-74126642,width-640,resizemode-4,imgsize-598663/but-coronavirus-scans-tend-to-have-white-patches-that-radiologists-refer-to-as-ground-glass-opacity-.jpg',width=400,height=400)



quantity = df.groupby(["city\_code"])["quantity"].sum().reset\_index().sort\_values("quantity",ascending=False).reset\_index(drop=True)

quantity

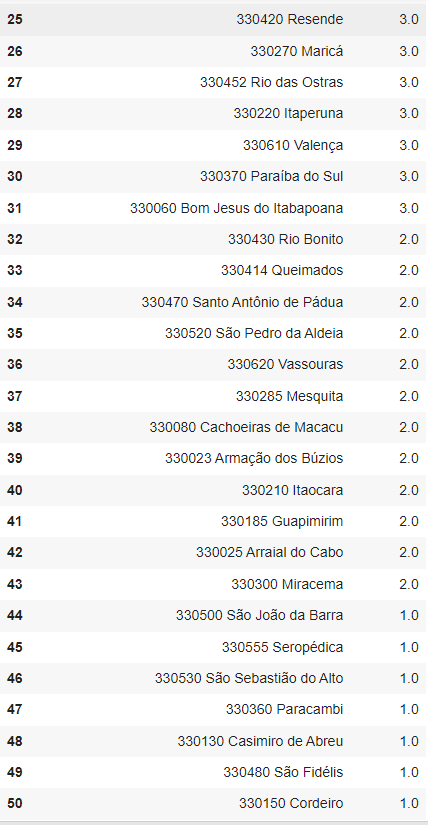


fig = go.Figure(data=[go.Bar(

            x=quantity['city\_code'][0:10], y=quantity['quantity'][0:10],

            text=quantity['quantity'][0:10],

            textposition='auto',

            marker\_color='black'

        )])

fig.update\_layout(

    title='Quantity',

    xaxis\_title="city\_code",

    yaxis\_title="quantity",

)

fig.show()

fig = go.Figure(data=[go.Scatter(

    x=quantity['city\_code'][0:10],

    y=quantity['quantity'][0:10],

    mode='markers',

    marker=dict(

        color=[145, 140, 135, 130, 125, 120,115,110,105,100],

        size=[100, 90, 70, 60, 60, 60,50,50,40,35],

        showscale=True

        )

)])

fig.update\_layout(

    title='CT Scan',

    xaxis\_title="Cities",

    yaxis\_title="Quantity",

)

fig.show()

labels = ["city\_code","quantity"]

values = quantity.loc[0, ["city\_code","quantity"]]

df = px.data.tips()

fig = px.pie(quantity, values=values, names=labels, color\_discrete\_sequence=['royalblue','darkblue','lightcyan'])

fig.update\_layout(

    title='Total quantity : '+str(quantity["quantity"][0]),

)

fig.show()

from IPython.display import Image

Image(url = 'https://www.businessinsider.in/thumb/msid-74126639,width-640,resizemode-4,imgsize-499455/scans-from-a-27-year-old-woman-who-worked-in-wuhan-showed-a-ground-glass-halo-white-patches-that-surround-a-small-nodule-.jpg',width=400,height=400)

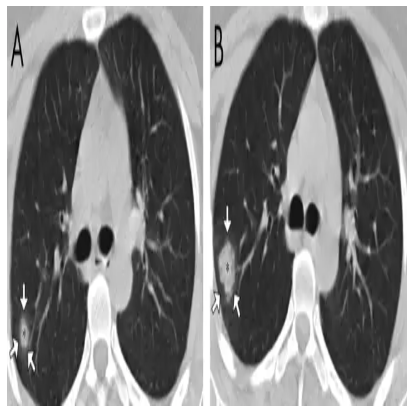


fig = px.pie(quantity, values=quantity['quantity'], names=quantity['city\_code'],

             title='Recovered cases',

            )

fig.update\_traces(textposition='inside', textinfo='percent+label')

fig.show()

from IPython.display import Image

Image(url = 'https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcSr6esjaF9D1xae\_4gmDVGT0qpi7x--J66ApPpKUdTkGiLE7P1J',width=400,height=400)



**RESULT:**

In the realm of 3D Classification and Segmentation in CT (Computed Tomography), cutting-edge techniques merge image processing and machine learning to unlock valuable insights within voluminous medical imaging data.