Neuro Segmenter 3D

[ ]:
```
#Instillation of Required Libraries
!pip install pydicom
!pip install nilearn
!pip install tqdm
!pip install imageio
!pip install imageio-ffmpeg
!pip install segmentation_models_3D
```

```
Collecting pydicom
  Downloading pydicom-3.0.0-py3-none-any.whl.metadata (9.4 kB)
Downloading pydicom-3.0.0-py3-none-any.whl (1.9 MB)
                         1.9/1.9 MB
61.2 MB/s eta 0:00:00
Installing collected packages: pydicom
Successfully installed pydicom-3.0.0
Collecting nilearn
  Downloading nilearn-0.10.4-py3-none-any.whl.metadata (7.8 kB)
Requirement already satisfied: joblib>=1.0.0 in /usr/local/lib/python3.10/dist-
packages (from nilearn) (1.4.2)
Requirement already satisfied: lxml in /usr/local/lib/python3.10/dist-packages
(from nilearn) (4.9.4)
Requirement already satisfied: nibabel>=4.0.0 in /usr/local/lib/python3.10/dist-
packages (from nilearn) (5.2.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-
packages (from nilearn) (1.26.4)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from nilearn) (24.1)
Requirement already satisfied: pandas>=1.1.5 in /usr/local/lib/python3.10/dist-
packages (from nilearn) (2.1.4)
Requirement already satisfied: requests>=2.25.0 in
/usr/local/lib/python3.10/dist-packages (from nilearn) (2.32.3)
Requirement already satisfied: scikit-learn>=1.0.0 in
/usr/local/lib/python3.10/dist-packages (from nilearn) (1.3.2)
Requirement already satisfied: scipy>=1.8.0 in /usr/local/lib/python3.10/dist-
packages (from nilearn) (1.13.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->nilearn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
```

```
packages (from pandas>=1.1.5->nilearn) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=1.1.5->nilearn) (2024.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.25.0->nilearn) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests>=2.25.0->nilearn) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.25.0->nilearn) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.25.0->nilearn)
(2024.8.30)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0.0->nilearn)
(3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas>=1.1.5->nilearn) (1.16.0)
Downloading nilearn-0.10.4-py3-none-any.whl (10.4 MB)
                         10.4/10.4 MB
84.3 MB/s eta 0:00:00
Installing collected packages: nilearn
Successfully installed nilearn-0.10.4
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages
(4.66.5)
Requirement already satisfied: imageio in /usr/local/lib/python3.10/dist-
packages (2.35.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from imageio) (1.26.4)
Requirement already satisfied: pillow>=8.3.2 in /usr/local/lib/python3.10/dist-
packages (from imageio) (10.4.0)
Requirement already satisfied: imageio-ffmpeg in /usr/local/lib/python3.10/dist-
packages (0.5.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-
packages (from imageio-ffmpeg) (71.0.4)
Collecting segmentation_models_3D
  Downloading segmentation_models_3D-1.1.1-py3-none-any.whl.metadata (539 bytes)
Collecting keras==3.3.3 (from segmentation_models_3D)
  Downloading keras-3.3.3-py3-none-any.whl.metadata (5.7 kB)
Collecting classification-models-3D==1.1.0 (from segmentation_models_3D)
  Downloading classification_models_3D-1.1.0-py3-none-any.whl.metadata (638
bytes)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-
packages (from keras==3.3.3->segmentation_models_3D) (1.4.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from keras==3.3.3->segmentation_models_3D) (1.26.4)
Requirement already satisfied: rich in /usr/local/lib/python3.10/dist-packages
(from keras==3.3.3->segmentation_models_3D) (13.8.1)
Requirement already satisfied: namex in /usr/local/lib/python3.10/dist-packages
```

```
(from keras==3.3.3->segmentation_models_3D) (0.0.8)
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
(from keras==3.3.3->segmentation_models_3D) (3.11.0)
Requirement already satisfied: optree in /usr/local/lib/python3.10/dist-packages
(from keras==3.3.3->segmentation_models_3D) (0.12.1)
Requirement already satisfied: ml-dtypes in /usr/local/lib/python3.10/dist-
packages (from keras==3.3.3->segmentation_models_3D) (0.4.1)
Requirement already satisfied: typing-extensions>=4.5.0 in
/usr/local/lib/python3.10/dist-packages (from
optree->keras==3.3.3->segmentation_models_3D) (4.12.2)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.10/dist-packages (from
rich->keras==3.3.3->segmentation_models_3D) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.10/dist-packages (from
rich->keras==3.3.3->segmentation_models_3D) (2.18.0)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-
packages (from markdown-it-
py>=2.2.0->rich->keras==3.3.3->segmentation_models_3D) (0.1.2)
Downloading segmentation_models_3D-1.1.1-py3-none-any.whl (34 kB)
Downloading classification_models_3D-1.1.0-py3-none-any.whl (69 kB)
                         69.8/69.8 kB
6.1 MB/s eta 0:00:00
Downloading keras-3.3.3-py3-none-any.whl (1.1 MB)
                         1.1/1.1 MB
49.6 MB/s eta 0:00:00
Installing collected packages: classification-models-3D, keras,
segmentation_models_3D
  Attempting uninstall: keras
    Found existing installation: keras 3.4.1
    Uninstalling keras-3.4.1:
      Successfully uninstalled keras-3.4.1
Successfully installed classification-models-3D-1.1.0 keras-3.3.3
segmentation_models_3D-1.1.1
```

[ ]:
```python
#Importing Libraries and Functions
from tqdm import tqdm
import os
import tensorflow as tf

import time
from random import randint
import glob
import gc
import numpy as np
from scipy import stats
import pandas as pd
```

```
from tensorflow.keras.utils import to_categorical

from sklearn.model_selection import train_test_split
from sklearn.model_selection import StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVR
from sklearn.model_selection import KFold
from sklearn.preprocessing import MinMaxScaler
import nibabel as nib
import pydicom as pdm
import nilearn as nl
import nilearn.plotting as nlplt
import h5py
from keras.metrics import MeanIoU
import matplotlib.pyplot as plt
from matplotlib import cm
import matplotlib.animation as anim
import matplotlib.patches as mpatches
import matplotlib.gridspec as gridspec

import seaborn as sns
import imageio
from skimage.transform import resize
from skimage.util import montage

from IPython.display import Image as show_gif
from IPython.display import clear_output
from IPython.display import YouTubeVideo
from torchvision import transforms
from torch import nn
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader
import torch.nn.functional as F
import torchvision.transforms.functional as TF
from torchvision import transforms
from torch.optim import Adam
from torch.optim.lr_scheduler import ReduceLROnPlateau
from torch.nn import MSELoss

# !pip install albumentations==0.4.6
import albumentations as A
# from albumentations.pytorch import ToTensor, ToTensorV2


from albumentations import Compose, HorizontalFlip
# from albumentations.pytorch import ToTensor, ToTensorV2
```

```
import warnings
```

[ ]:
```
!pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.0.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-
packages (from datasets) (3.16.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
packages (from datasets) (1.26.4)
Collecting pyarrow>=15.0.0 (from datasets)
  Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (3.3
kB)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from datasets) (2.1.4)
Requirement already satisfied: requests>=2.32.2 in
/usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-
packages (from datasets) (4.66.5)
Collecting xxhash (from datasets)
  Downloading
xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(12 kB)
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in
/usr/local/lib/python3.10/dist-packages (from
fsspec[http]<=2024.6.1,>=2023.1.0->datasets) (2024.6.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-
packages (from datasets) (3.10.5)
Requirement already satisfied: huggingface-hub>=0.22.0 in
/usr/local/lib/python3.10/dist-packages (from datasets) (0.24.7)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
```

```
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-
packages (from aiohttp->datasets) (1.11.1)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.22.0->datasets)
(4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets)
(2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets)
(2024.8.30)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
Downloading datasets-3.0.0-py3-none-any.whl (474 kB)
                         474.3/474.3 kB
24.1 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
                         116.3/116.3 kB
9.0 MB/s eta 0:00:00
Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (39.9 MB)
                         39.9/39.9 MB
39.3 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
                         134.8/134.8 kB
6.7 MB/s eta 0:00:00
Downloading
xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
                         194.1/194.1 kB
16.3 MB/s eta 0:00:00
Installing collected packages: xxhash, pyarrow, dill, multiprocess,
datasets
  Attempting uninstall: pyarrow
    Found existing installation: pyarrow 14.0.2
```

```
    Uninstalling pyarrow-14.0.2:
      Successfully uninstalled pyarrow-14.0.2
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
cudf-cu12 24.4.1 requires pyarrow<15.0.0a0,>=14.0.1, but you have pyarrow 17.0.0
which is incompatible.
ibis-framework 8.0.0 requires pyarrow<16,>=2, but you have pyarrow 17.0.0 which
is incompatible.
Successfully installed datasets-3.0.0 dill-0.3.8 multiprocess-0.70.16
pyarrow-17.0.0 xxhash-3.5.0
```

[ ]:
```python
class BRATSDataset(Dataset):
    def __init__(self, data_dir, scaler,transform=None):
        self.data_dir = data_dir
        self.t2_dir = sorted(glob.glob(self.data_dir + "BraTS20_*/*t2.nii"))
        self.t1ce_dir = sorted(glob.glob(self.data_dir + "BraTS20_*/*t1ce.nii"))
        self.flair_dir = sorted(glob.glob(self.data_dir + "BraTS20_*/*flair.
 ↪nii"))
        self.mask_dir = sorted(glob.glob(self.data_dir + "BraTS20_*/*seg.nii"))
        self.scaler = scaler

        self.trans_t = transforms.Compose([
            transforms.ToTensor()
        ])
    def __len__(self):
        return len(self.t2_dir)
    def __getitem__(self, idx):
      t2 = nib.load(self.t2_dir[idx]).get_fdata()
      t2 = self.scaler.fit_transform(t2.reshape(-1, t2.shape[-1])).reshape(t2.
 ↪shape)
      t1ce = nib.load(self.t1ce_dir[idx]).get_fdata()
      t1ce = self.scaler.fit_transform(t1ce.reshape(-1, t1ce.shape[-1])).
 ↪reshape(t1ce.shape)
      flair = nib.load(self.flair_dir[idx]).get_fdata()
      flair = self.scaler.fit_transform(flair.reshape(-1, flair.shape[-1])).
 ↪reshape(flair.shape)
      mask = nib.load(self.mask_dir[idx]).get_fdata()
      mask = mask.astype(np.uint8)
      mask[mask == 4] = 3
      mask = to_categorical(mask, num_classes=4)
      image = np.stack([t2, t1ce, flair], axis=3)
      image = image[56:184, 56:184, 13:141]
```

```
        mask = mask[56:184, 56:184, 13:141]
        image = image.reshape(3,128,128,128)
        mask = mask.reshape(4,128,128,128)
        image = torch.from_numpy(image.astype(np.float32))
        mask = torch.from_numpy(mask)
        return image, mask

class DatasetRepeater(Dataset):

    def __init__(self, dataset, num_repeats=1):
        self.dataset = dataset
        self.num_repeats = num_repeats
        self.data_len = self.dataset.__len__()

    def __len__(self):
        return self.num_repeats * self.data_len

    def __getitem__(self, idx):
        return self.dataset[idx % self.data_len]
```

[ ]:
```
class DoubleConv(nn.Module):
  def __init__(self, in_channels, out_channels):
    super(DoubleConv, self).__init__()
    self.conv = nn.Sequential(
        nn.Conv3d(in_channels, out_channels, 3, 1, 1, bias=False),
        nn.BatchNorm3d(out_channels),
        nn.ReLU(inplace=True),
        nn.Conv3d(out_channels, out_channels, 3, 1, 1, bias=False),
        nn.BatchNorm3d(out_channels),
        nn.ReLU(inplace=True),
    )
  def forward(self, x):
    return self.conv(x)
class UNET(nn.Module):
  def __init__(self, in_channels = 3, out_channels = 3, features = [64, 128,␣
 ↪256, 512]):
    super(UNET, self).__init__()
    self.ups = nn.ModuleList()
    self.downs = nn.ModuleList()
    self.pool = nn.MaxPool3d(kernel_size=2, stride=2)
    self.softmax = nn.Softmax(dim=1)


    for feature in features:
      self.downs.append(DoubleConv(in_channels, feature))
      in_channels = feature
```

```
    for feature in reversed(features):
      self.ups.append(nn.ConvTranspose3d(feature*2, feature, kernel_size=2,␣
 ↪stride=2))
      self.ups.append(DoubleConv(feature*2, feature))

    self.bottlneck = DoubleConv(features[-1], features[-1]*2)
    self.final_conv = nn.Conv3d(features[0], out_channels, kernel_size=1)



  def forward(self, x):
    skip_connections = []

    for down in self.downs:
      x = down(x)
      skip_connections.append(x)
      x = self.pool(x)

    x = self.bottlneck(x)
    skip_connections = skip_connections[::-1]


    for idx in range(0, len(self.ups),2):
      x = self.ups[idx](x)
      skip_connection = skip_connections[idx//2]
      x = torch.cat((skip_connection, x), dim=1)
      x = self.ups[idx+1](x)
    x = self.final_conv(x)
    return x
```

[ ]:
```
scaler = MinMaxScaler()
train_dir = "/content/drive/MyDrive/Research/BRATS2020/BraTS2020_TrainingData/
 ↪MICCAI_BraTS2020_TrainingData/"
test_dir = "/content/drive/MyDrive/Research/BRATS2020/BraTS2020_ValidationData/
 ↪MICCAI_BraTS2020_ValidationData/"
dataset = BRATSDataset(train_dir, scaler)
train_dataloader, test_dataloader = torch.utils.data.random_split(dataset, [0.
 ↪8, 0.2])
train_loader = DataLoader(train_dataloader, batch_size=1, shuffle=True,␣
 ↪num_workers=5)
test_loader = DataLoader(test_dataloader, batch_size=1, shuffle=False,␣
 ↪num_workers=5)
```

[ ]:
```
def FocalLoss(inputs, targets):
  BCE = F.binary_cross_entropy_with_logits(inputs, targets, reduction='mean')
  BCE_EXP = torch.exp(-BCE)
```

```
  focal_loss = .8 * (1 - BCE_EXP)**2 * BCE

  return focal_loss
```

[ ]:
```
def DiceLoss( inputs, targets, smooth=1):
      inputs = F.sigmoid(inputs)
      intersection = (inputs * targets).sum()
      dice = (2. * intersection + smooth) / \
      (inputs.sum() + targets.sum() + smooth)
      return 1 - dice
```

[ ]:
```
def criterion(x,y):
  y = y.to(device)
  x = x.to(device)
  return DiceLoss(x,y) + (1 * FocalLoss(x,y))
```

[ ]:
```
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
model = UNET(in_channels=3, out_channels=4).to(device)
optimizer = Adam(model.parameters(), lr=1e-4, weight_decay=1e-5)
```

[ ]:
```
def batch_process(model, image, mask):
  pred = model(image)
  loss = criterion(pred, mask)
  return loss
```

[ ]:
```
def train_one_epoch(model,optimizer,train_loader,device):
  model.train()
  total_loss = 0.
  for i,(image, mask )in enumerate(train_loader):
    image = image.to(device)
    mask = mask.to(device)
    loss = batch_process(model, image, mask)
    loss.backward()
    optimizer.step()
    optimizer.zero_grad()
    total_loss += loss.detach().item()
  return total_loss/len(train_loader)
```

[ ]:
```
@torch.no_grad()
def evaluate(model,val_loader,device):
  model.eval()
  total_loss = 0.

  for i,(image, mask ) in enumerate(val_loader):
    image = image.to(device)
    mask = mask.to(device)
    loss = batch_process(model, image, mask)
```

```
    total_loss += loss.detach().item()
  return total_loss/len(val_loader)
```

[ ]:
```
num_epochs = 25
history = {"loss": [], "val_loss": []}
for epoch in range(num_epochs):
  train_loss = train_one_epoch(model, optimizer, train_loader, device)


  val_loss = evaluate(model, test_loader, device)

  history["loss"].append(train_loss)
  history["val_loss"].append(val_loss)

  print(f"Epoch: {epoch+1}, Train Loss: {train_loss}, Val Loss: {val_loss}")
```

[ ]:
```
def plot_metrics(history, stidx=0):
    epochs = len(history["loss"][stidx:])
    plt.figure(figsize=(15,5))
    plt.plot(range(epochs), history["loss"][stidx:], label="Train")
    plt.plot(range(epochs), history["val_loss"][stidx:], label="Val")
    plt.ylabel('Loss')
    plt.xlabel('epoch')
    plt.legend()

    plt.suptitle("Metrics vs epoch")
    plt.show()
```

[ ]:
```
plot_metrics(history)
```

[ ]:
```
torch.save(model.state_dict(), '/content/drive/MyDrive/Research/BRATS2020/UNET.
↪pth')
```

[ ]:
```
model.load_state_dict(torch.load('/content/drive/MyDrive/Research/BRATS2020/
↪UNET.pth'))
```

```
<ipython-input-6-2b3c3b65f8a4>:1: FutureWarning: You are using `torch.load` with
`weights_only=False` (the current default value), which uses the default pickle
module implicitly. It is possible to construct malicious pickle data which will
execute arbitrary code during unpickling (See
https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for
more details). In a future release, the default value for `weights_only` will be
flipped to `True`. This limits the functions that could be executed during
unpickling. Arbitrary objects will no longer be allowed to be loaded via this
mode unless they are explicitly allowlisted by the user via
`torch.serialization.add_safe_globals`. We recommend you start setting
```

```
`weights_only=True` for any use case where you don't have full control of the
loaded file. Please open an issue on GitHub for any issues related to this
experimental feature.
  model.load_state_dict(torch.load('/content/drive/MyDrive/Research/BRATS2020/UN
ET.pth'))
```

[ ]:
```
<All keys matched successfully>
```

[ ]:
```
output = model(test_dataloader[0][0].to(device).unsqueeze(0)).squeeze()
output = output.cpu().detach().numpy()
```

[ ]:
```
output = np.argmax(output, axis=0)
output.shape
```

[ ]:
```
(128, 128, 128)
```

[ ]:
```
mask = np.argmax(test_dataloader[0][1], axis=0)
mask.shape
```

[ ]:
```
torch.Size([128, 128, 128])
```

[ ]:
```
from sklearn.metrics import recall_score, precision_score, f1_score,␣
 ↪confusion_matrix, jaccard_score
recall = []
percision = []
f1 = []
jaccard = []
```

[ ]:
```
image = test_dataloader[0][0]
image.shape
```

[ ]:
```
torch.Size([3, 128, 128, 128])
```

[ ]:
```
image = np.rot90(montage(image[0]))
mask = np.rot90(montage(mask))

fig, ax = plt.subplots(1, 1, figsize = (20, 20))
ax.imshow(image, cmap ='bone')
ax.imshow(np.ma.masked_where(mask == False, mask),
          cmap='cool', alpha=0.6)
```

[ ]:
```
<matplotlib.image.AxesImage at 0x7f1d0de8ab00>
```