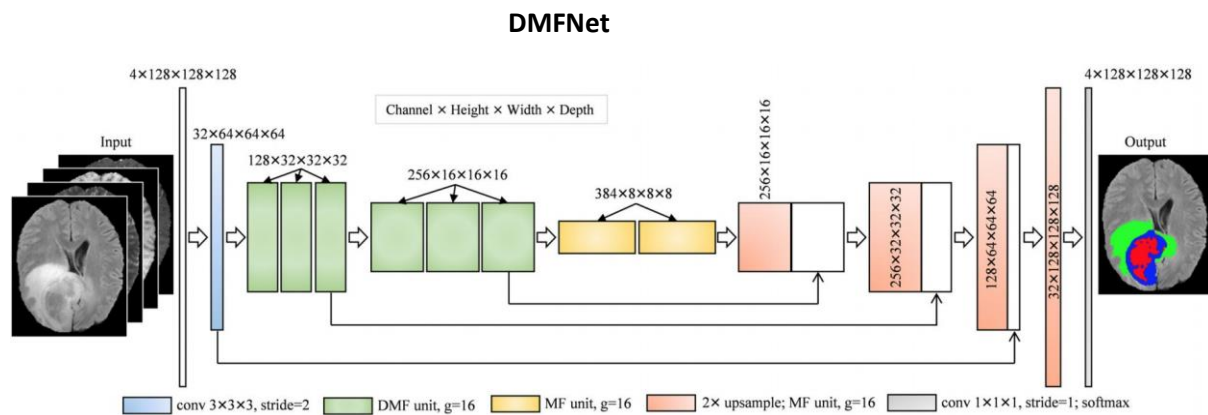


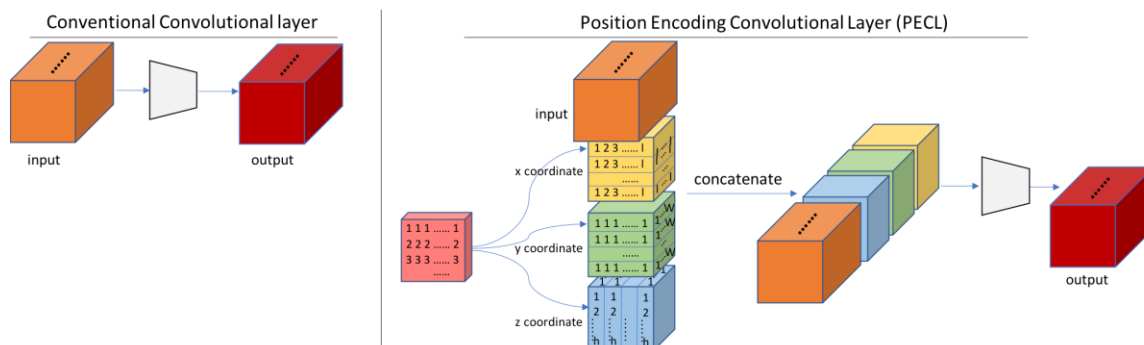
## Proposed Model in given article:

### Domain Knowledge Based Brain Tumor Segmentation and Overall Survival Prediction

The BraTS segmentation challenge involves segmenting four modality MRI scans of brain tumors, which is computationally expensive. To address this, the authors use a DMFNet model that reduces computational cost by slicing a complex neural network into lightweight fibers and incorporating multiplexer modules. They also add dilated convolution to capture multi-scale 3D spatial correlations. The authors use weighted dice loss to address the imbalanced class problem in contrast-enhancing tumor segmentation, where the smallest tumor volume often leads to the worst accuracy. They use the reciprocal of each tumor volume as the dice weight, which results in 0.38, 0.15, and 0.47 for necrosis, edema, and enhancing tumor, respectively.



Position encoding is a technique used in deep learning models to incorporate spatial information into the feature maps extracted from images or other types of data. In the context of the given text, the proposed position encoding convolutional layer (PECL) extends the standard 3D convolutional layer by adding three extra channels representing the x, y, and z coordinates of each voxel in the input image. These channels are normalized by dividing their maximum value and concatenated with the original feature maps before further processing. This allows the model to consider the spatial location of each voxel in the image, which is crucial for accurate prediction of overall survival (OS) in patients with gliomas. The PECL is used in a PECL-DenseNet architecture, which alternates between densely connected convolutional blocks and transition layers to extract deep features from T1ce and T2 MRI images and segmentations of the tumor subregions. The deep features are then concatenated with a resection status digit and reduced to 50 dimensions before being input to a gradient boosted decision tree (GBDT) regression model to predict OS.



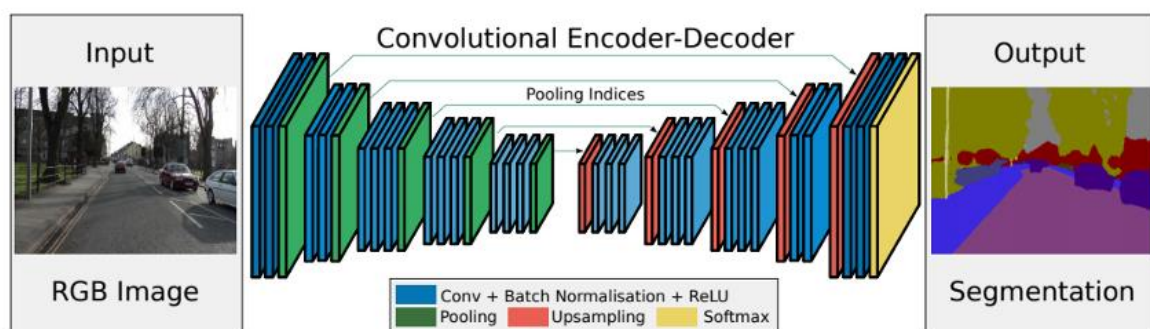
**Comparison of 3D convolutional layer and the proposed PECL.**

## Conclusion:

The authors of this paper proposed a framework for brain tumor segmentation and overall survival (OS) prediction using medical imaging data. They utilized a deep learning model called DMFNet with weighted dice loss for segmentation, which resulted in a balanced prediction for small and large objects in MRI scans with reduced computation cost. They also proposed a PECL-DenseNet for feature extraction and combined it with handcrafted and clinical features for GBDT regression-based OS prediction. The study showed promising results for both segmentation and OS prediction, but the authors suggest that integrating more MRI modality data and molecular information may improve the performance of the proposed framework.

## Model Used For Training on the dataset [Task01\\_BrainTumour](#):

The model used composes **PyTorch Lightning** module for training a segmentation model on medical imaging data using MONAI, a deep learning framework for healthcare. The architecture used for the model is **SegResNet**, which is a residual network that has been specifically designed for medical image segmentation. It has a series of down-sampling and up-sampling blocks, with skip connections between the corresponding encoder and decoder blocks.



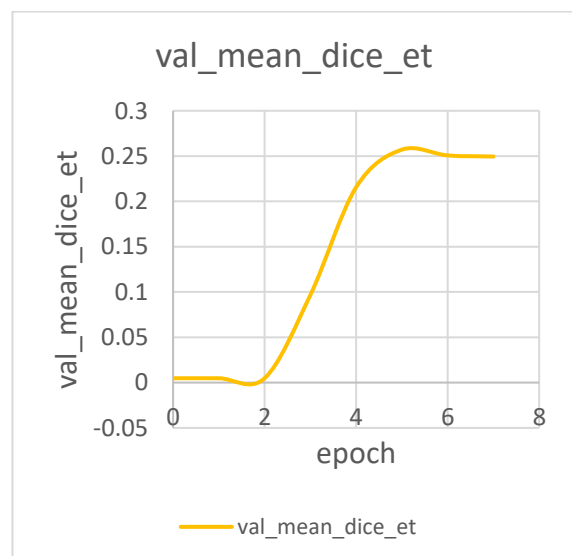
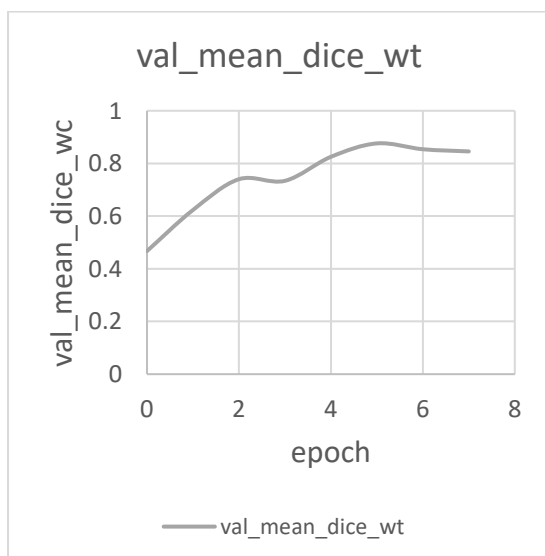
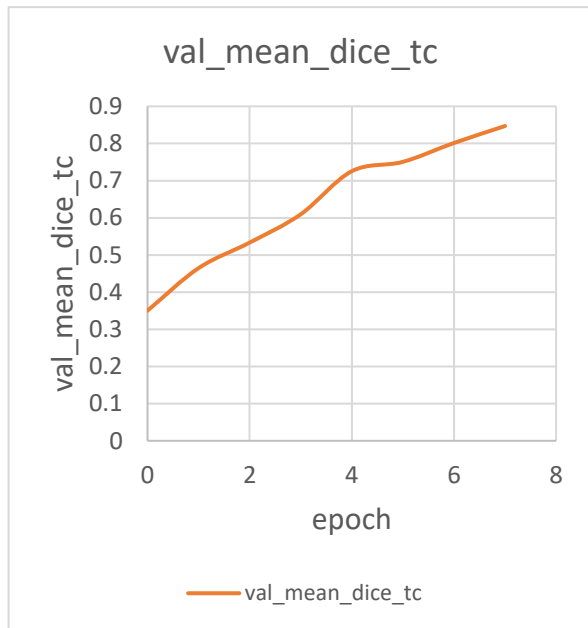
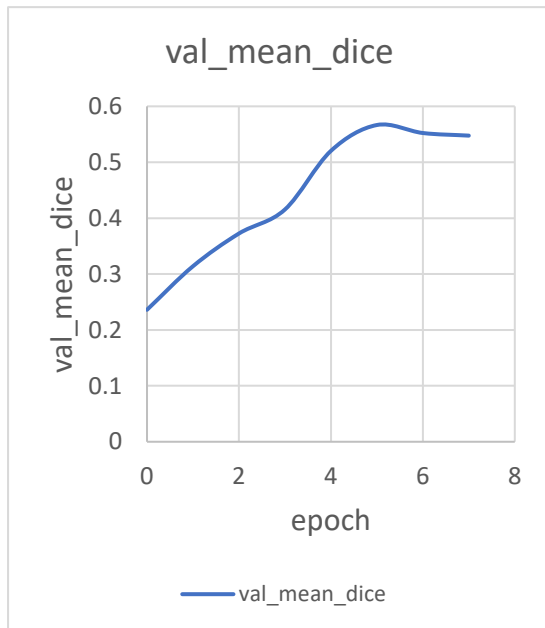
**SegResNet Architecture**

The loss function used is DiceLoss, which is a common loss function for segmentation tasks that measures the overlap between the predicted and ground truth segmentation masks. The data is preprocessed using a series of MONAI transforms, such as normalization, cropping, and flipping. The train\_transform contains more data augmentation techniques compared to val\_transform. The optimizer used is Adam. During training, the training\_step method is called for each batch of data, and the validation\_step method is called for each batch of validation data. The metrics used in this module are dice loss, dice coefficient, batch dice coefficient, and Hausdorff distance. The module also includes methods to set up data loaders for training and validation, and to configure the optimizer.

I have tried to train model for more epochs but due to storage, ram and time constraints on google colab, I am not able to train model for longer periods. So the submitted model is trained for only 8 epoch and still the accuracy in segmentation is good. If the same model is allowed to train for longer epochs, it will perform much better compared to present trained model.

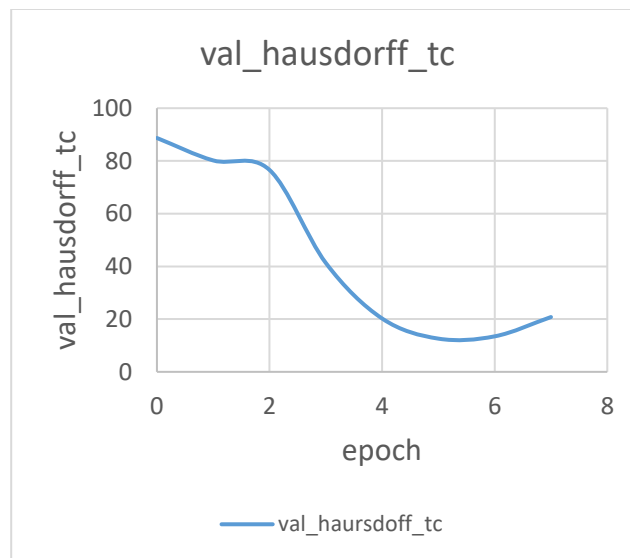
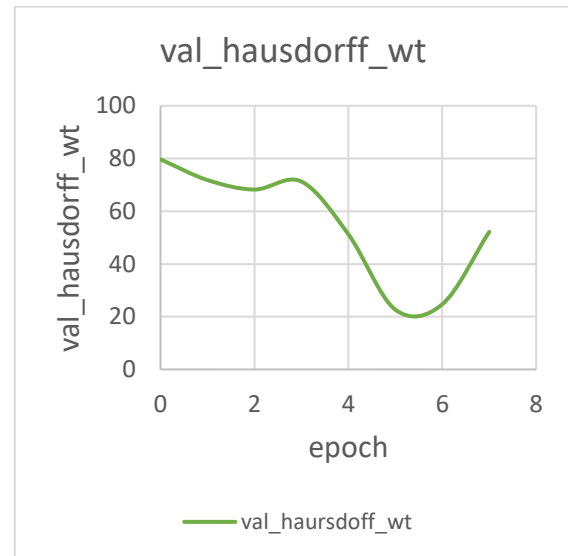
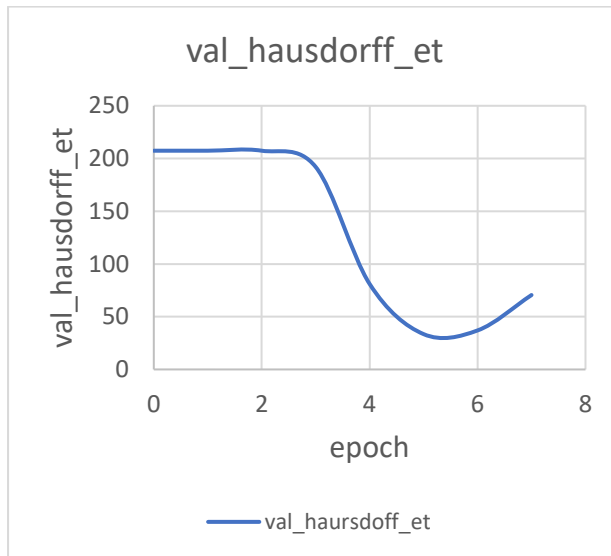
## Results :

**Metric : DiceMetric**



val_mean_dice	val_mean_dice_tc	val_mean_dice_wt	val_mean_dice_et
0.5666	0. 7506	0. 8761	0.2571

### Metric : Hausdorff Metric



val_hausdorff_tc	val_hausdorff_wt	val_hausdorff_et
12.5935	22.6943	33.5108

# Predictions :

label 0 is the peritumoral edema,

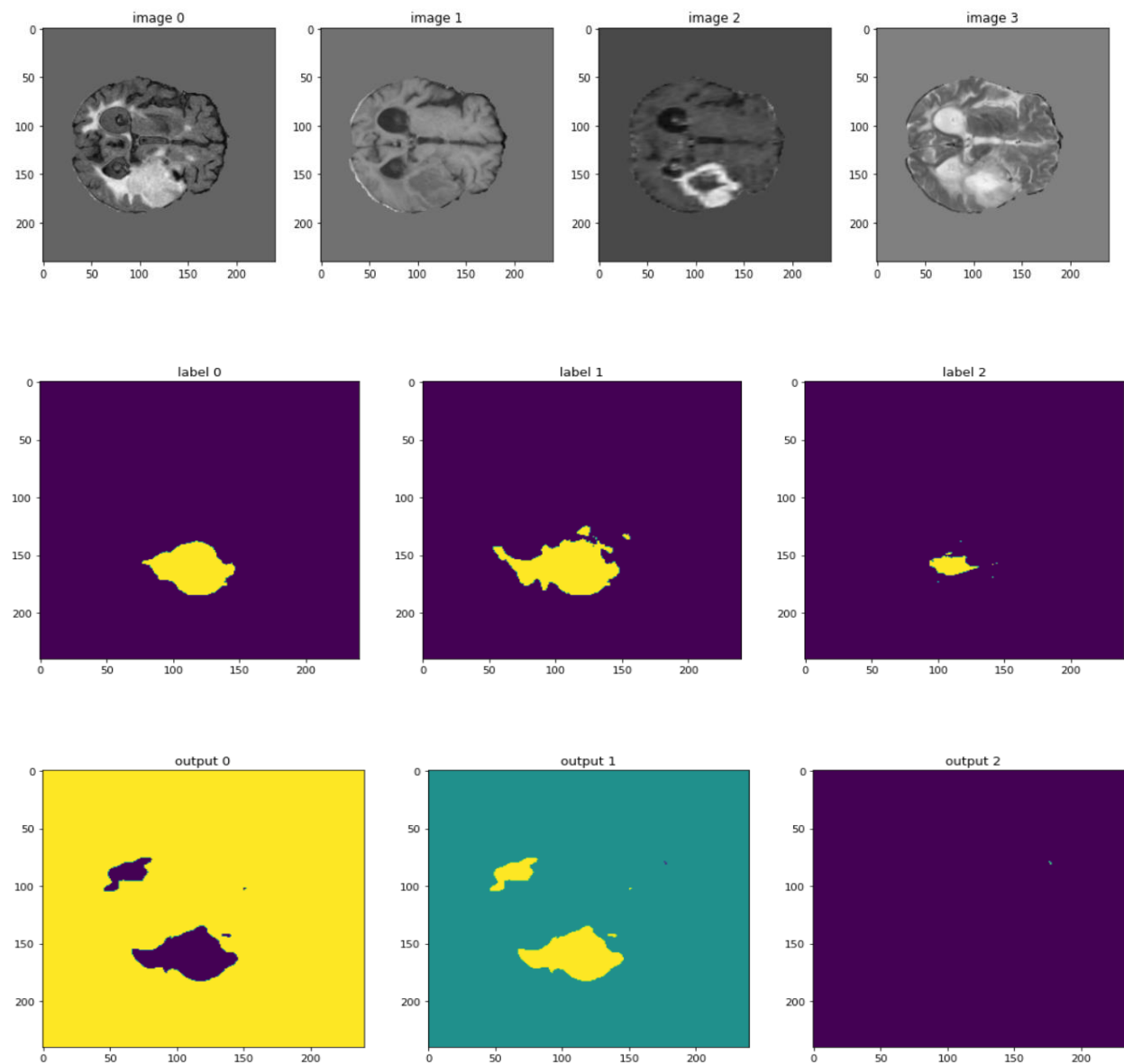
label 1 is the GD-enhancing tumor,

label 2 is the necrotic and non-enhancing tumor core.

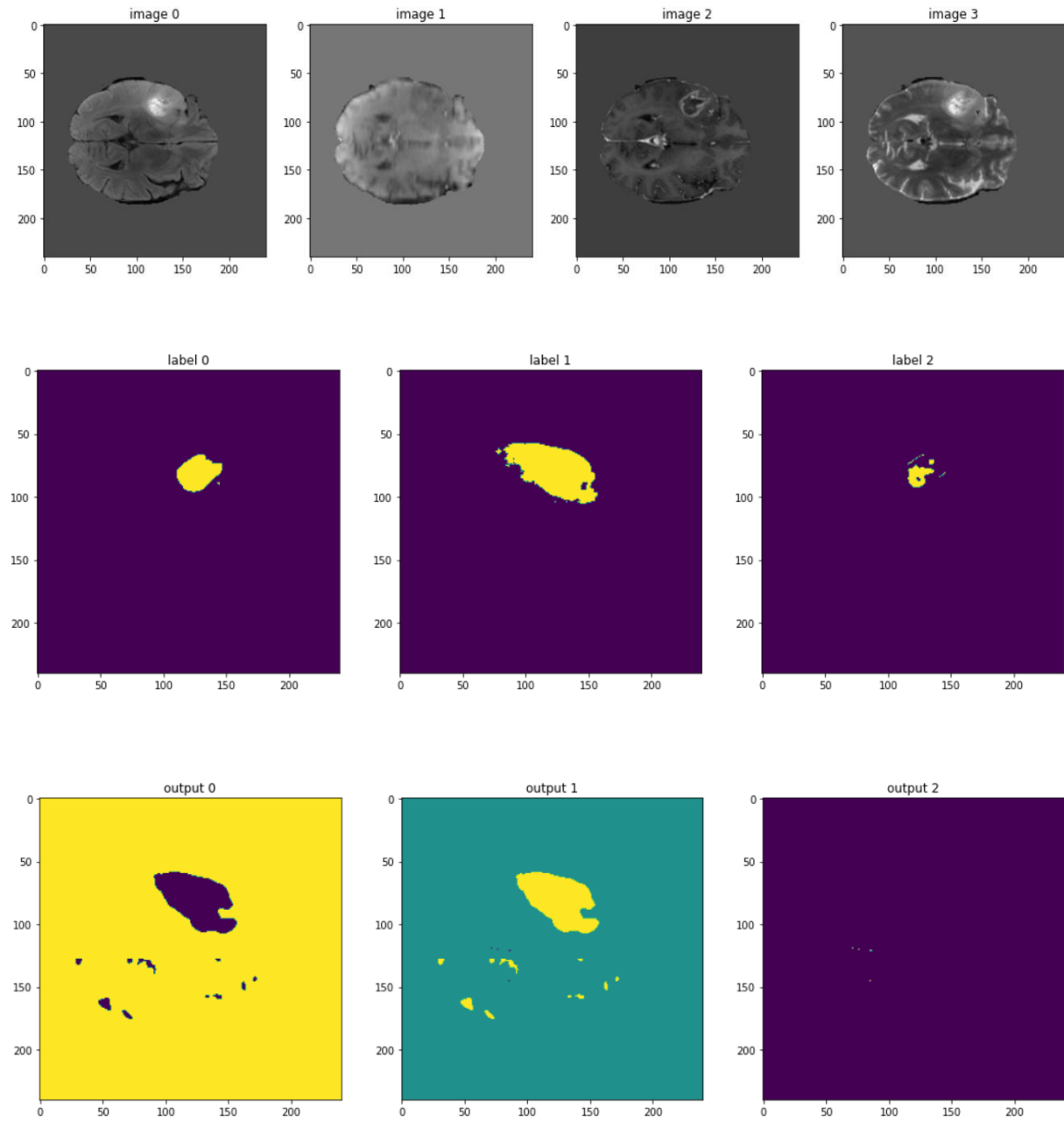
The possible classes are **TC (Tumor core)**, **WT (Whole tumor)** and **ET (Enhancing tumor)**.

Image 0 : FLAIR, Image 1 : T1w, Image 2 :T1gd, Image 3 : T2w

## Prediction 1:



## Prediction 2:



### Prediction 3:

