"Author" Zhifang Zheng(45788167)

Pattern Recognition

Description

This report illustrates how to train a U-Net for OASIS brain tumor segmentation, using a minimum Dice similarity coefficient of 0.9 on the test dataset. The data is consist of MR images of the 300+ human brains with accompanying segmentations, which we need to download from Cloidstore with the link 'https://cloudstor.aarnet.edu.au/plus/s/n5aZ4XX1WBKp6HZ/download'.

There are 2 scipt files, a README.md file, and one image file(final results and model_summary)

Environment

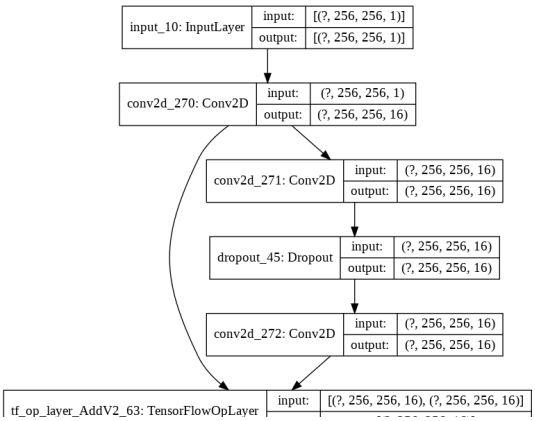
*tensorflow version: 2.3.0
*python version: 3.7.9

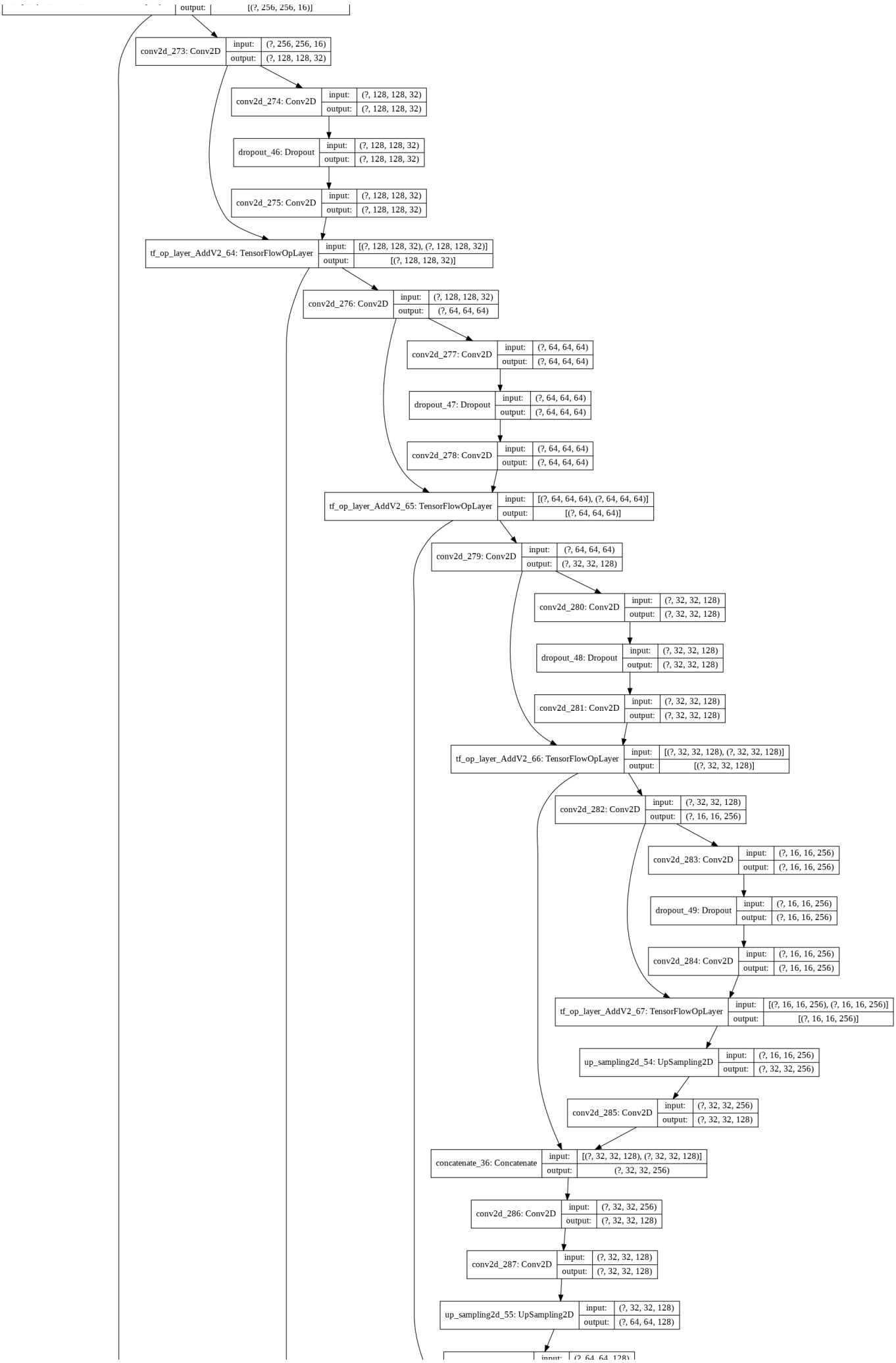
Alogrithm

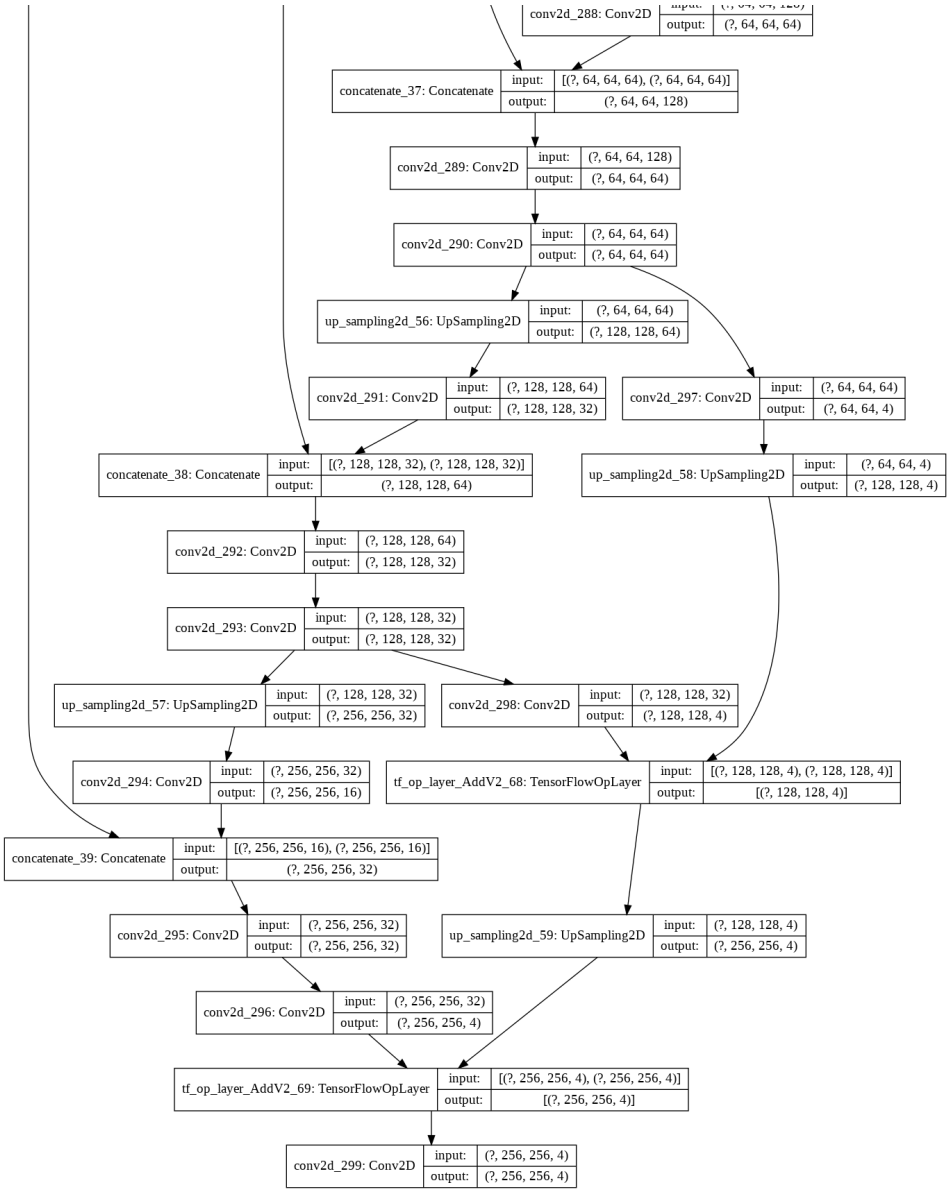
- Unet_Model: Depending on the U-Net architecure, we designed the network to process large 2D input block of 256 * 256 voxsels.
  *Parameters:
  - out_channel: 4
  - input_data: tensor.shape(256, 256, 1)
  - activation: LealyRelu(alpha=0.01), softmax
  - epochs: 20
  - train_ds.batch: 20
  - val_ds.batch: 20

  *Loss Function:
  - dice_coef: metrics=[(2. * intersection + smooth) / (K.sum(train_ds_f) + K.sum(test_ds_f) + smooth)]
  - dice_coef_loss: loss=dice_coef_loss

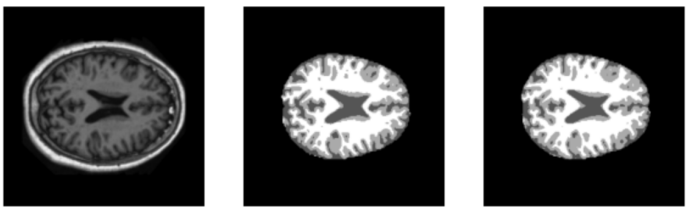
Model flow chart (the process of building the model)




output: [(?, 256, 256, 16)]
conv2d_273: Conv2D
input: (?, 256, 256, 16)
output: (?, 128, 128, 32)
conv2d_274: Conv2D
input: (?, 128, 128, 32)
output: (?, 128, 128, 32)
dropout_46: Dropout
input: (?, 128, 128, 32)
output: (?, 128, 128, 32)
conv2d_275: Conv2D
input: (?, 128, 128, 32)
output: (?, 128, 128, 32)
tf_op_layer_AddV2_64: TensorFlowOpLayer
input: [(?, 128, 128, 32), (?, 128, 128, 32)]
output: [(?, 128, 128, 32)]
conv2d_276: Conv2D
input: (?, 128, 128, 32)
output: (?, 64, 64, 64)
conv2d_277: Conv2D
input: (?, 64, 64, 64)
output: (?, 64, 64, 64)
dropout_47: Dropout
input: (?, 64, 64, 64)
output: (?, 64, 64, 64)
conv2d_278: Conv2D
input: (?, 64, 64, 64)
output: (?, 64, 64, 64)
tf_op_layer_AddV2_65: TensorFlowOpLayer
input: [(?, 64, 64, 64), (?, 64, 64, 64)]
output: [(?, 64, 64, 64)]
conv2d_279: Conv2D
input: (?, 64, 64, 64)
output: (?, 32, 32, 128)
conv2d_280: Conv2D
input: (?, 32, 32, 128)
output: (?, 32, 32, 128)
dropout_48: Dropout
input: (?, 32, 32, 128)
output: (?, 32, 32, 128)
conv2d_281: Conv2D
input: (?, 32, 32, 128)
output: (?, 32, 32, 128)
tf_op_layer_AddV2_66: TensorFlowOpLayer
input: [(?, 32, 32, 128), (?, 32, 32, 128)]
output: [(?, 32, 32, 128)]
conv2d_282: Conv2D
input: (?, 32, 32, 128)
output: (?, 16, 16, 256)
conv2d_283: Conv2D
input: (?, 16, 16, 256)
output: (?, 16, 16, 256)
dropout_49: Dropout
input: (?, 16, 16, 256)
output: (?, 16, 16, 256)
conv2d_284: Conv2D
input: (?, 16, 16, 256)
output: (?, 16, 16, 256)
tf_op_layer_AddV2_67: TensorFlowOpLayer
input: [(?, 16, 16, 256), (?, 16, 16, 256)]
output: [(?, 16, 16, 256)]
up_sampling2d_54: UpSampling2D
input: (?, 16, 16, 256)
output: (?, 32, 32, 256)
conv2d_285: Conv2D
input: (?, 32, 32, 256)
output: (?, 32, 32, 128)
concatenate_36: Concatenate
input: [(?, 32, 32, 128), (?, 32, 32, 128)]
output: (?, 32, 32, 256)
conv2d_286: Conv2D
input: (?, 32, 32, 256)
output: (?, 32, 32, 128)
conv2d_287: Conv2D
input: (?, 32, 32, 128)
output: (?, 32, 32, 128)
up_sampling2d_55: UpSampling2D
input: (?, 32, 32, 128)
output: (?, 64, 64, 128)



Visualization (Prediction)



```
484/484 [==============================] - 167s 345ms/step - loss: -0.9903 - dice_coef: 0.9903 - val_loss: -0.9870 - val_dice_coef: 0.9870
```