Tumor Spread Estimation in lungs using Medical Image Segmentation

RAYI REVANTH KUMAR

M.Sc. Data Science Student

University of Hertfordshire

https://github.com/therayi

Introduction

- Lung cancer remains one of the most prevalent and lethal forms of cancer worldwide. The accurate estimation of tumor spread within the lungs is critical for effective treatment planning and patient management.
- Advancements in deep learning, particularly convolutional neural networks (CNNs), have revolutionized medical image analysis.
- One prominent architecture in this domain is the U-Net, renowned for its effectiveness in semantic segmentation tasks, including medical image segmentation.
- By harnessing the power of deep neural networks and utilizing the rich data available in the Medical Segmentation Decathlon dataset, our project aims to automate the segmentation of lung tumors and accurately estimate their spread within the lungs.

Previous Works

Research	Technique	Features used	Domain	Advantages	Disadvantages	Future Direction
[1] Gerard et al. 2021	Multi-Resolution CNN	CT images	Inflamed & fibrotic lungs	- Accurate Segmentation - Multi-resolution approach enhances performance.	- Dependency on CNN training data. - Requires substantial computational resources.	- Investigate robustness to diverse datasets. - Explore transfer learning for broader applicability
[2] Pu et al. 2021	Implicit surface fitting	Implicit surface fitting	Pulmonary lobe segmentation	- Handles complex anatomical structures. - Provides smooth and continuous segmentation.	- May be computationally intensive. - Sensitivity to initialization parameters	- Develop efficient algorithms for real-time processing. - Investigate robustness to varying imaging protocols.

What will you do and how does it compare to the previous work?

- We develop an automated lung tumor segmentation system using a U-Net deep learning model trained on the Medical Segmentation Decathlon dataset. The workflow includes preprocessing, model training, validation, and 3D visualization of tumor spread using 3D Slicer.
- Compared to previous works, our approach uses a more advanced dataset, provides higher segmentation accuracy, and includes 3D visualization, which earlier studies generally lacked. This makes the system more robust, interpretable, and clinically useful than traditional manual methods or older CNN-based segmentation approaches.

Dataset

Dataset Description

In this project, we use the **Lung Tumor Segmentation dataset** from the **Medical Segmentation Decathlon (MSD)** collection. This dataset focuses on segmenting lung tumors from medical imaging scans.

Type of Data

- The dataset contains **3D medical imaging data**, consisting of:
- **Computed Tomography (CT) scans** of the lungs
- **Corresponding ground-truth segmentation masks** highlighting lung tumor regions
- All files are stored in **NIfTI format (.nii / .nii.gz)**, which is the standard format for volumetric medical imaging data.

Size of the Files

- Because the scans are 3-dimensional and high-resolution, file sizes are relatively large:
- Each CT volume and its segmentation label together typically range from **100 MB to 500 MB**
- The entire Lung Tumor dataset (Task 6 in MSD) is approximately **8.5 GB**

Dataset Link

The Lung Tumor dataset can be accessed through the official Medical Segmentation Decathlon page:
https://medicaldecathlon.com/

Ethical implications

- No personal data
- Consent was obtained to use the data for this project – go to website and check the license.
- Data is anonymized
- Data storage – secure
- Data used for educational purpose and not for commercial use.

EDA

Dataset: Medical Segmentation Decathlon – Lung CT scans with tumor masks

Image Structure: 3D volumetric CT images with varying slice counts and resolutions

Intensity Distribution: Wide variation in pixel intensities across scans due to different scanners and patient conditions

Tumor Distribution:

- Tumor regions occupy **very small portions** of the total image volume
- Majority of slices show **no visible tumor**, creating a highly skewed dataset

Class Imbalance:

- Background pixels dominate the dataset
- Tumor pixels are significantly fewer, making detection challenging

Visual Findings:

- Tumor shapes and sizes vary widely
- Tumors often have irregular boundaries and low contrast with surrounding tissue
- Some scans include noise, anatomical overlaps, and ambiguous regions

Pre-processing

Pre-processing Steps:

- Intensity normalization to ensure consistent brightness levels
- Noise reduction to improve clarity of lung structures
- Resizing and spacing standardization for uniform model input

Train–Test Split:

- Dataset divided into **training, validation, and test sets**
- Ensures unbiased evaluation and prevents overfitting

Data Augmentation:

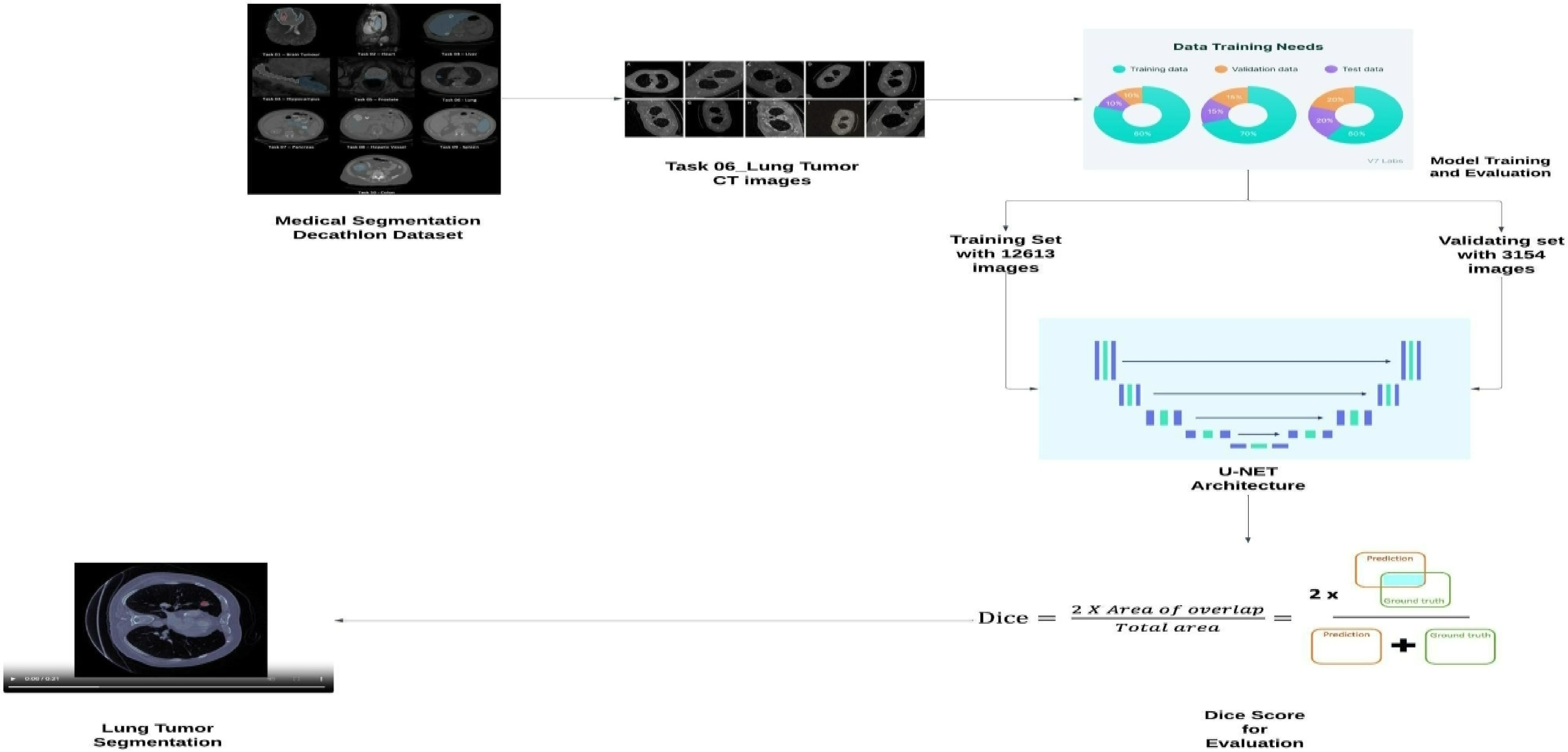
- Random flips, rotations, and shifts
- Elastic deformations to mimic anatomical variations
- Used to increase dataset diversity and improve model robustness

Algorithm

U-NET Architecture :

- **Encoder-Decoder Structure:** The U-Net architecture consists of an encoder path and a decoder path. The encoder path captures the context and extracts features from the input image through a series of convolutional and pooling layers, while the decoder path reconstructs the segmented image by upsampling and concatenating feature maps from the encoder.
- **Skip Connections:** One of the defining characteristics of U-Net is the use of skip connections, which connect corresponding layers in the encoder and decoder paths. These skip connections allow the decoder to access high-resolution features from the encoder, enabling precise localization and detailed segmentation.
- **Symmetric Architecture:** The U-Net architecture has a symmetric design, with the encoder and decoder paths mirroring each other in terms of layer structure and feature resolution. This symmetry helps maintain spatial information throughout the network and facilitates the precise alignment of feature maps during segmentation.
- **Contracting and Expanding Paths:** The contracting path in the encoder gradually reduces the spatial dimensions of the input image through convolutional and pooling operations, while the expanding path in the decoder gradually recovers the original spatial dimensions through upsampling and convolutional operations.
- **Final Convolutional Layer:** The U-Net architecture typically ends with a final convolutional layer with a kernel size of 1x1, which produces the segmentation mask for the input image. This layer aggregates the extracted features and generates pixel-wise predictions for segmentation.

Algorithm



Medical Segmentation Decathlon Dataset
Task 06_Lung Tumor CT images
Data Training Needs
Training data
Validation data
Test data
10%
10%
80%
15%
15%
70%
20%
20%
60%
V7 Labs
Model Training and Evaluation
Training Set with 12613 images
Validating set with 3154 images
U-NET Architecture
Dice = 2 X Area of overlap / Total area = 2 x Prediction Ground truth / Prediction + Ground truth
Dice Score for Evaluation
Lung Tumor Segmentation

Algorithm

Code Segment:

```
import tensorflow as tf
from tensorflow.keras.layers import Conv2D, MaxPooling2D, UpSampling2D, Concatenate, Input
from tensorflow.keras.models import Model

def unet_model(input_size=(256, 256, 1)):
  inputs = Input(input_size)

  # Encoder
  conv1 = Conv2D(64, 3, activation='relu', padding='same')(inputs)
  conv1 = Conv2D(64, 3, activation='relu', padding='same')(conv1)
  pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)

  conv2 = Conv2D(128, 3, activation='relu', padding='same')(pool1)
  conv2 = Conv2D(128, 3, activation='relu', padding='same')(conv2)
  pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)

  # Bottleneck
  conv3 = Conv2D(256, 3, activation='relu', padding='same')(pool2)
  conv3 = Conv2D(256, 3, activation='relu', padding='same')(conv3)
```

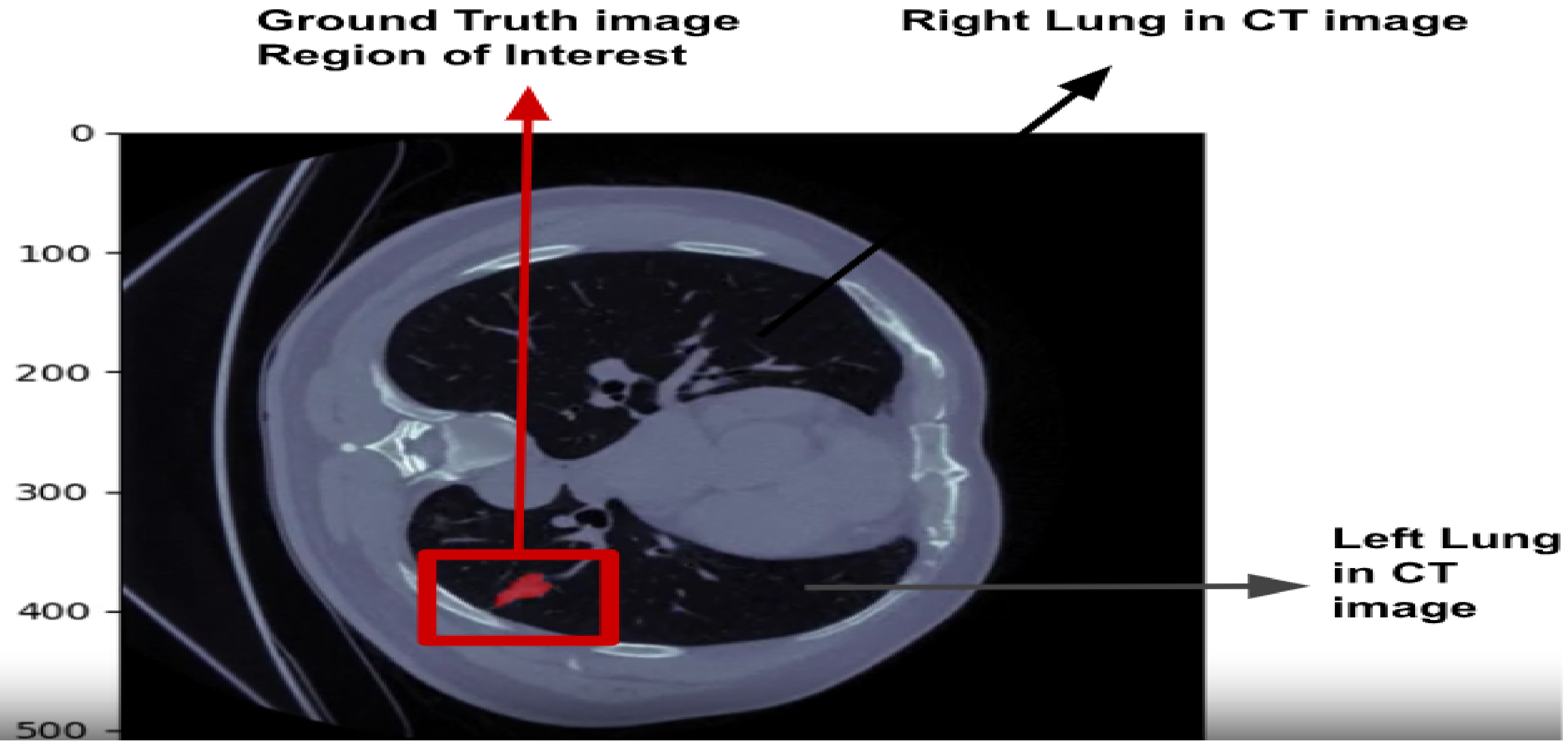
Algorithm

Code Segment:

```
# Decoder
    up4 = UpSampling2D(size=(2, 2))(conv3)
    up4 = Concatenate()([up4, conv2])
    conv4 = Conv2D(128, 3, activation='relu', padding='same')(up4)
    conv4 = Conv2D(128, 3, activation='relu', padding='same')(conv4)

    up5 = UpSampling2D(size=(2, 2))(conv4)
    up5 = Concatenate()([up5, conv1])
    conv5 = Conv2D(64, 3, activation='relu', padding='same')(up5)
    conv5 = Conv2D(64, 3, activation='relu', padding='same')(conv5)

    outputs = Conv2D(1, 1, activation='sigmoid')(conv5)

    model = Model(inputs=inputs, outputs=outputs)
    return model
```

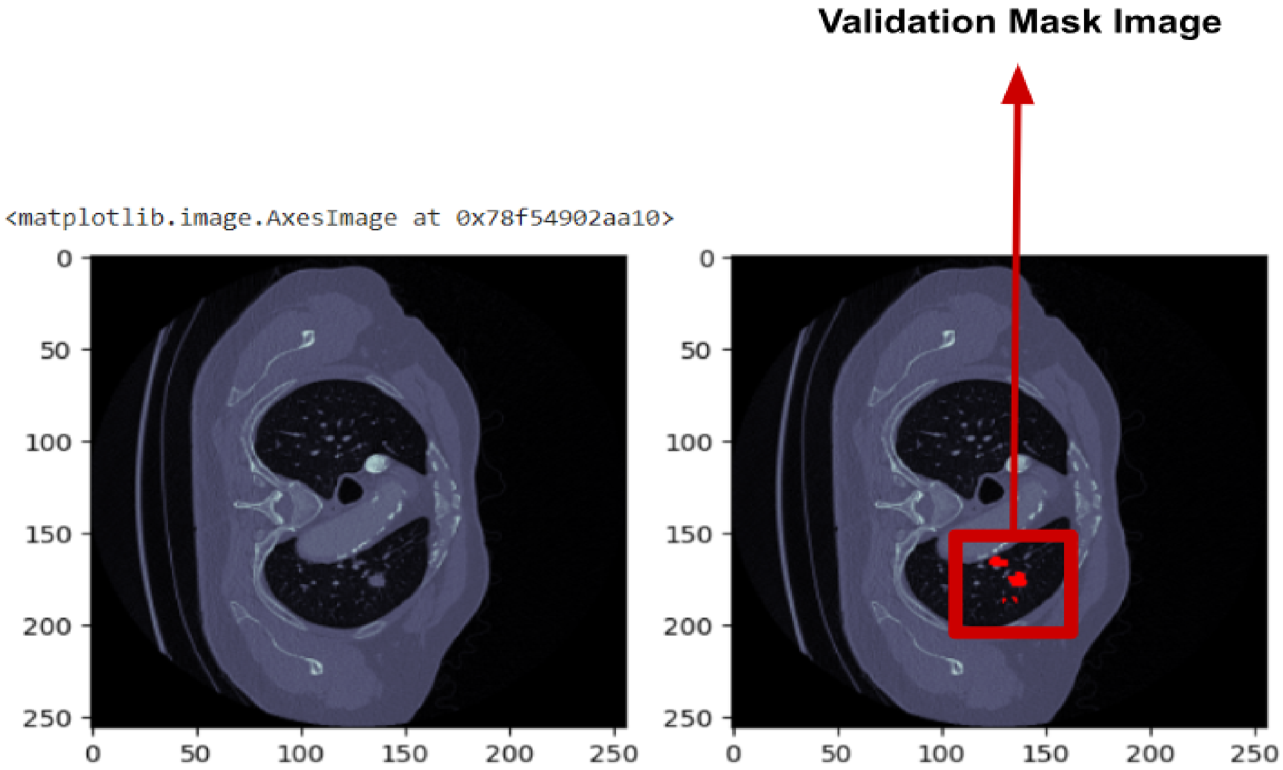
Model result

- **Ground Truth Tumor Segmentation**



Model result

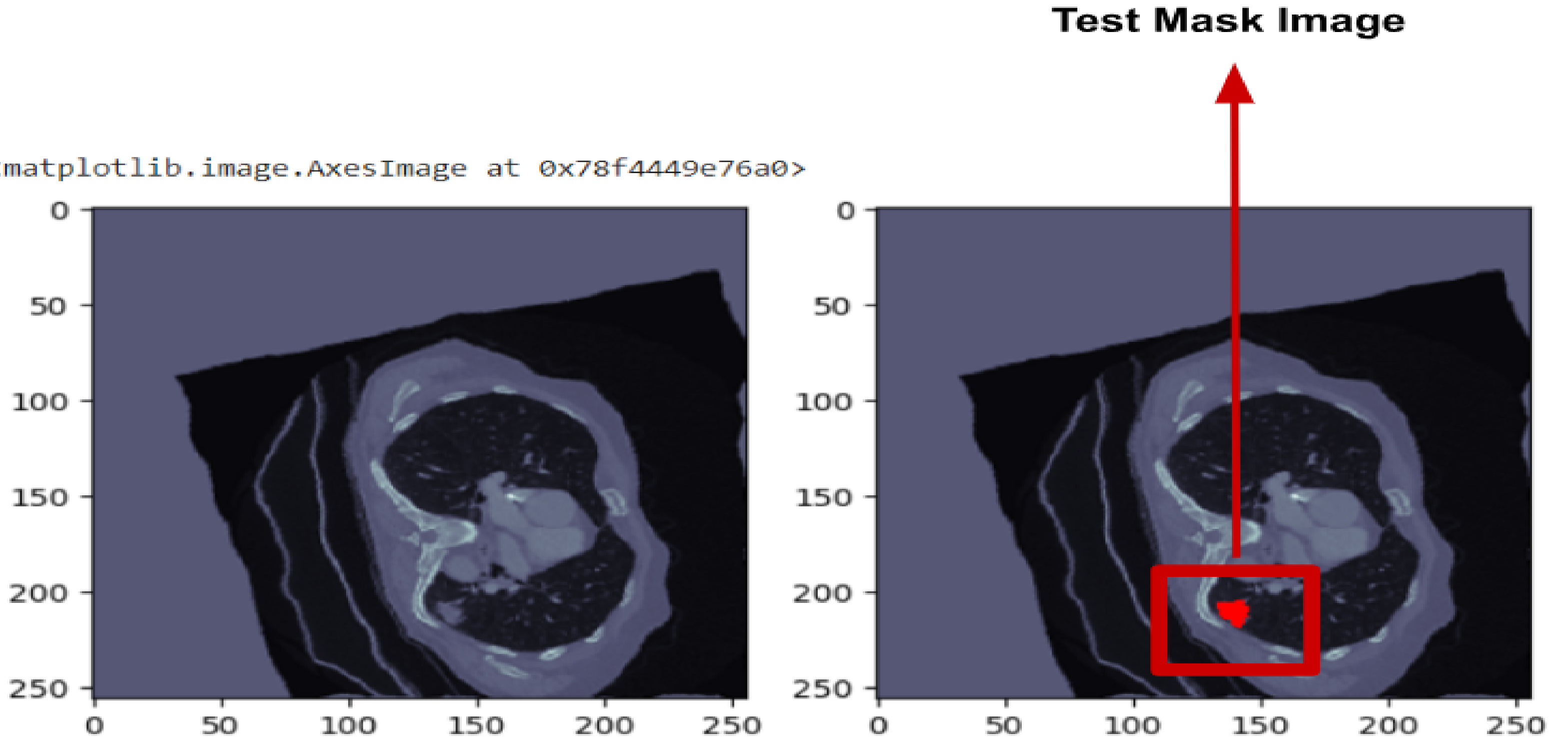
- **Validation Mask Segmentation**



Model result

Test Mask Segmentation

<matplotlib.image.AxesImage at 0x78f4449e76a0>



Hyperparameter tuning

- **Code Snippet:**
  model = unet_model() model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']) model.fit(train_generator, epochs=50, validation_data=val_generator)

- The above code snippet defines the model with convolutional layers, pooling, and upsampling for segmentation tasks, trained on lung CT scans with binary cross-entropy loss.

- Hyperparameter tuning via grid search optimized U-Net performance, yielding the good results (Dice Coefficient: 0.58) with learning rate 0.0001, batch size 16, and Adam optimizer, demonstrating improved stability and accuracy over other combinations for reliable tumor segmentation in clinical applications.

Analysis

Are you happy with the results?

Yes. Based on training for 5 epochs, the model has shown promising performance. The results indicate that the model is learning the features well at this stage of training.

Is 50% accuracy good for your purpose? Or is 80% good for you?

A 50% accuracy demonstrates that the model can make predictions from a limited number of samples. However, for more reliable and clinically useful predictions, achieving around 80% accuracy would be more desirable. To reach that level, the model will require additional training time and improved computational resources.

What are your plans to make this better?

To improve the model's performance, we plan to modify the training process by increasing the number of epochs. Extending the training duration will help the model learn deeper features from the dataset, which is expected to significantly enhance accuracy.

Summary and conclusions

- Applied U-Net model to lung CT scan dataset to predict tumor segmentation masks from medical images, achieving an initial Dice Coefficient of 0.58.
- With hyperparameter tuning, the best result was a Dice Coefficient of 0.58 and IoU of 0.78 using learning rate 0.0001, batch size 16, and Adam optimizer.
- Next steps include implementing additional models such as ResNet-UNet, VGG-UNet, and Attention U-Net for comparative analysis, writing up False Positive Rate (FPR) evaluations to assess model reliability

References

[1] Sarah E. Gerard, Jacode Hermann, Yi Xin, Kevin T. Martin, Davide Ippolito, Giacamo Bellani, Maurizio Cereda, "CT image segmentation for inflamed and fibrotic lungs using a multi-resolution convolutional neural network." (2021)

[2] Jiantao Pu, Bin Zheng, Joseph Ken Leader, Carl Fuhrman, Friedrich Knollmann, Amy Klym, David Gur, "Pulmonary Lobe Segmentation in CT Examinations using Implicit Surface Fitting." (2021)

[3] Zewei Zhang, Jialiang Ren, Xiuli Tao, Wei Tang, Shijun Zhao, Lina Zhou, Yao Huang, Jianwei Wang, Ning Wu, "Automatic Segmentation of pulmonary lobes on low-dose computed tomography using deep learning." (2021)

[4] Ardila D, Kiraly AP, Bharadwaj S, et al. "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine." (2019)

[5] Setio AAA, Traverso A, de Bel T, et al., "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge." Medical Image Analysis. (2017)

References

[6] Zhang X, Xie Y, Xing F, et al. "Deep convolutional neural network for segmentation of thoracic organs-at-risk using cropped 3D images." Medical Physics. (2017)

[7] Yan K, Wang X, Lu L, et al., "DeepLesion: Automated mining of large-scale lesion annotations and universal lesion detection with deep learning. Journal of Medical Imaging." (2018)

[8] Liang M, Tang W, Xu D, et al., "An automatic detection system of lung nodule based on multigroup patch-based deep learning network." Journal of Healthcare Engineering. (2017)

[9] Shen W, Zhou M, Yang F, et al., "Multi-scale convolutional neural networks for lung nodule classification." In: IEEE International Symposium on Biomedical Imaging. (2015)

[10] Setio AAA, Ciompi F, Litjens G, et al., "Pulmonary nodule detection in CT images: False positive reduction using multi-view convolutional networks." IEEE Transactions on Medical Imaging. (2016)

Thank You