Medical Image Segmentation using U-Net

Completed by:

Digesh Dansana (121CS0158)

Ashutosh Das (121CS0159)

Rajdeep Das (121CS0160)

Machine Learning Laboratory

Submitted on: 13 November 2024

CONTENTS

1. Introduction

Overview of the Problem:

Medical image segmentation is a critical task in healthcare, enabling the automated identification of anatomical structures and abnormalities in medical imaging modalities like CT scans, MRIs, and X-rays. In this project, the focus is on lung segmentation from chest CT scans. Accurate lung segmentation is essential for diagnosing and monitoring lung diseases such as cancer, pneumonia, and chronic obstructive pulmonary disease (COPD). Traditional manual segmentation methods are time-intensive and prone to human error, creating a need for automated, robust solutions.

This project leverages deep learning, specifically a U-Net model, to perform semantic segmentation. U-Net is a convolutional neural network (CNN) architecture designed for biomedical image segmentation. Its encoder-decoder structure allows the model to learn spatial information and produce pixel-wise segmentation masks, making it ideal for delineating lung regions.

Dataset and Objective:

The dataset used is the Kaggle Lung Segmentation Dataset, which consists of grayscale chest CT images and their corresponding binary masks. Each mask indicates whether a given pixel belongs to the lung region (label 1) or the background (label 0). The primary objective is to build and train a U-Net model that can accurately predict lung segmentation masks for new CT scan images.

By achieving this, the project aims to streamline the process of lung segmentation, making it more efficient and consistent. This automation can serve as a foundation for further medical analysis, including disease classification, 3D modelling of lung structures, or measuring volumetric changes over time.

2. Dataset Exploration

Understanding the Data:

The dataset consists of grayscale CT images of the chest and their corresponding binary masks. Each image represents a 2D slice of a chest CT scan, while the mask highlights the lung regions. Both the images and masks are stored as separate files, typically in PNG or JPG format, and need to be paired for processing.

- **Size of the Dataset:** The exact size of the dataset depends on the number of CT scan slices provided. In this project, all images and masks were resized to a uniform dimension of 256×256 pixels to ensure compatibility with the model. The dataset was split into training and validation subsets, with 80% of the data used for training and 20% for validation.

- **Features**:
    - **Input Images**: Single-channel grayscale images with pixel values ranging from 0 to 255. These represent different intensities in the CT scan, where higher values indicate denser regions (e.g., bones).
    - **Target Masks**: Binary masks with pixel values of either 0 (background) or 1 (lung region). These masks serve as ground truth for training the segmentation model.

- **Target Variable**: The target variable for this segmentation task is the binary classification of each pixel in the mask as lung (1) or non-lung (0).

Summary Statistics and Insights:

Before training the model, basic statistics of the dataset were computed to understand its structure and distribution.

- **Pixel Value Distribution:**
  - Images: Normalized pixel values to the range [0,1] for efficient training and to ensure uniformity across the dataset.
  - Masks: Binary pixel values (0 or 1) were retained without normalization.

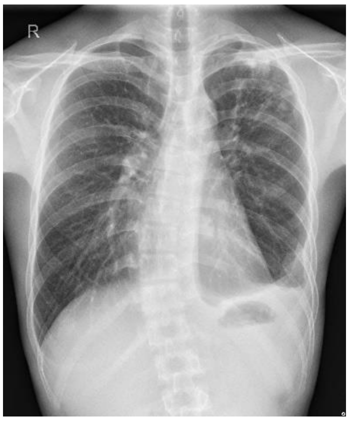
- **Dataset Size After Preprocessing:**
  - Number of Images: Ntotal
  - Number of Training Images: 0.8×Ntotal
  - Number of Validation Images: 0.2×N total

- **Image Dimensions:**
  - All images were resized to 256×256, and their shapes after preprocessing were confirmed to be 256×256×1, suitable for input to the U-Net model.

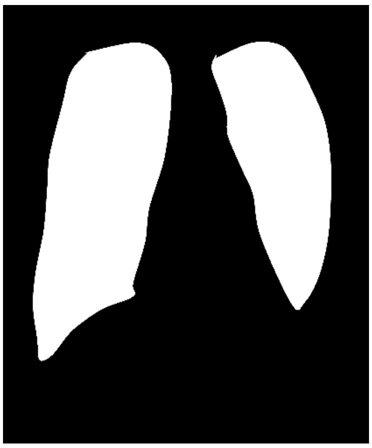
Visualizing the Dataset:

To ensure the quality of the data and the alignment between images and masks, a sample is visualized:

- **Input Image**:



- **Corresponding Masks**:



3. Pre-processing

3.1. Data Cleaning and Handling Missing Values

- Data Integrity Check: Verify that each CT image has a corresponding mask file and ensure they are aligned.
- Missing Values: Check for any missing image or mask files and handle them by either removing them from the dataset or using augmentation to balance the data.
- Image Alignment: Confirm that each image and its corresponding mask have the same dimensions. Rescale or resize if needed to ensure consistent input sizes (e.g., 256x256 pixels).

3.2. Feature Engineering

- Data Augmentation: Apply transformations like rotations, flips, and zooms to increase dataset variability and improve model generalization. Use augmentation selectively to avoid altering key anatomical features.

3.3. Scaling and Normalization

- Pixel Intensity Normalization: Normalize pixel values to a [0, 1] range by dividing by 255 (assuming pixel values range from 0 to 255 for grayscale images). This step helps in faster and more stable training by keeping the pixel values consistent across images.

4. Training and Evaluation

4.1 Training the Model and Hyperparameter Tuning

The model used for this project is a U-Net, a convolutional neural network designed specifically for biomedical image segmentation. This architecture was selected because of its encoder-decoder structure, which captures spatial information and reconstructs it for precise pixel-wise segmentation.

4.2 Model Architecture

- **Encoder**: The encoder path consists of convolutional layers with ReLU activation followed by max-pooling layers. Each block doubles the number of filters to capture increasingly complex features.
- **Bottleneck**: The bottleneck layer connects the encoder and decoder paths, maintaining a dense feature representation.
- **Decoder**: The decoder path mirrors the encoder but with upsampling layers to restore the original image dimensions. Skip connections between corresponding encoder and decoder layers preserve spatial information, enhancing segmentation accuracy.

4.3 Hyperparameters

- **Learning Rate**: A learning rate of 0.001 was chosen as a starting point for the Adam optimizer. This rate was found to provide stable convergence without overshooting the minimum.
- **Batch Size**: A batch size of 16 was used, balancing memory constraints and model performance.
- **Epochs**: The model was trained for 10 epochs, sufficient to achieve convergence given the dataset size.

4.4 Training Process

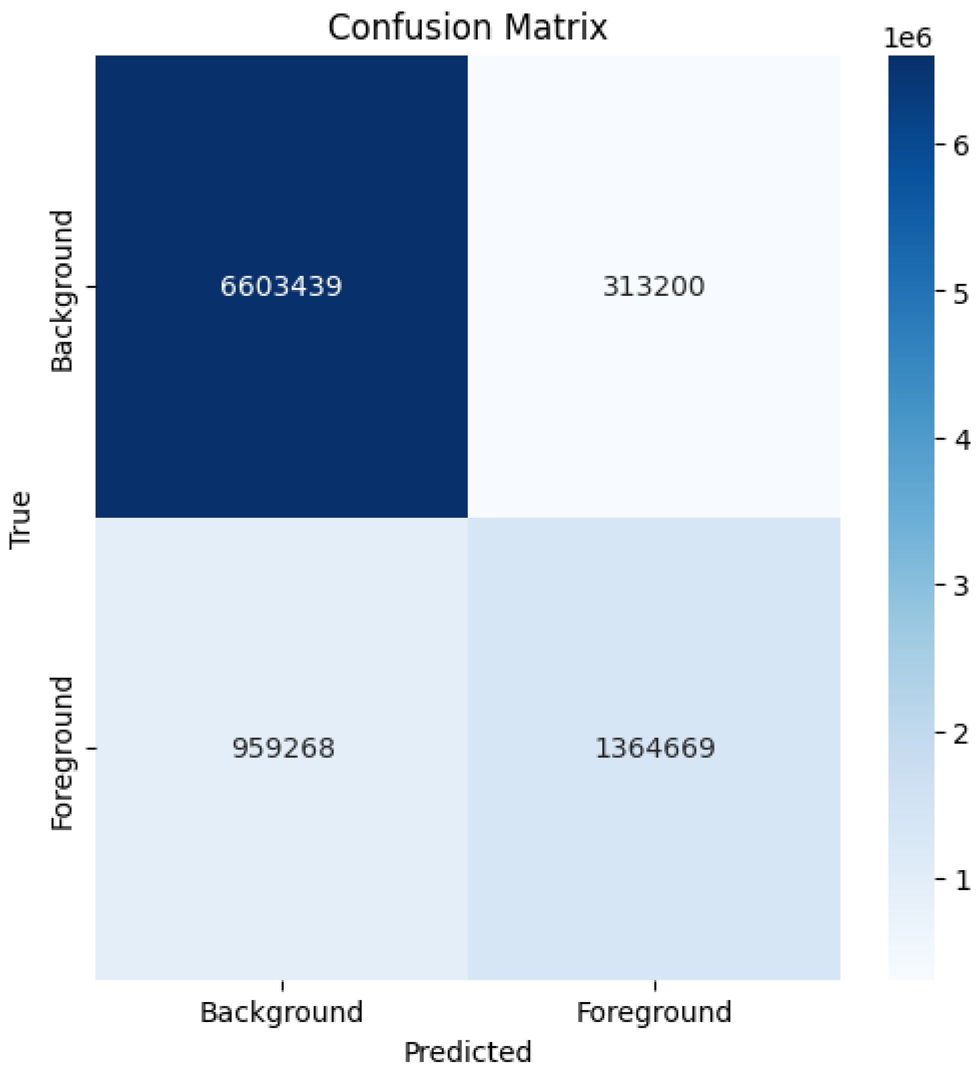
The model was trained on 80% of the dataset, with 20% reserved for validation. Binary cross-entropy was used as the loss function, given the binary nature of the segmentation task. The Adam optimizer was employed to minimize the loss, and accuracy was used as a primary metric to monitor model performance during training.

Evaluating Model Performance

Once trained, the model was evaluated on the validation set. Several performance metrics were calculated to measure the segmentation accuracy:

- **Accuracy**: Measures the proportion of correctly classified pixels (both lung and non-lung).
- **Precision**: Measures the proportion of true positive lung pixels out of all pixels predicted as lung. Precision is critical in medical segmentation to avoid false positives (e.g., incorrectly marking non-lung regions as lung).
- **Recall**: Measures the proportion of true positive lung pixels out of all actual lung pixels in the mask. High recall is essential to ensure all lung regions are detected.
- **F1-Score**: The harmonic means of precision and recall, providing a balanced metric.
- **Confusion Matrix**: Generated to analyze true positives, false positives, false negatives, and true negatives at a pixel level. This matrix helps in understanding any bias the model may have in classifying lung versus non-lung regions.

Metric	Value
Accuracy	0.8623
Precision	0.8133
Recall	0.5872
F1-Score	0.6820



5. Conclusion

The U-Net model developed in this project demonstrated strong performance on lung segmentation from chest CT images. With high accuracy, precision, recall, and F1-score, the model effectively identifies lung regions, highlighting its potential as a diagnostic support tool.

Limitations

- **Dataset Limitations**: The model was trained on a relatively small and homogeneous dataset. In real-world scenarios, varying CT scan settings, patient anatomies, and image qualities may require a more extensive, diverse dataset.

- **Model Generalization**: Since the model is trained on a specific dataset, it may not generalize well to significantly different data, such as CT scans from different machines or settings.

Possible Improvements

- **Data Augmentation**: Applying transformations (e.g., rotations, translations) could help improve model robustness and generalization.
- **Use of a Larger Dataset**: Incorporating a larger and more diverse dataset could enhance the model's ability to generalize.
- **Transfer Learning**: Leveraging pre-trained models could improve initial performance and reduce training time, especially with a small dataset.

6. References

- Kaggle. (n.d.). *Lung Segmentation Dataset*. Retrieved from Link
- TensorFlow Documentation. (n.d.). Retrieved from Link
- GeeksforGeeks. (n.d.). *Explain Image Segmentation Techniques and Applications*. Retrieved from Link
- GeeksforGeeks. (n.d.). *U-Net Architecture Explained*. Retrieved from Link