**CHAPTER 3: METHODOLOGY**

**3.1 Introduction**

This chapter outlines the methodology employed in the segmentation of mammographic masses using deep learning techniques. The focus is on the data preparation, model selection, training, evaluation, and implementation phases of the research. The aim is to develop a robust and accurate model that can identify and segment masses in mammographic images, contributing to improved breast cancer detection.

**3.2 Research Design**

The research is structured around an experimental design, which is appropriate for the objective of developing a precise and reliable model for the segmentation of mammographic masses. According to Litjens *et al.* (2017), this approach allows for systematic experimentation with various model configurations and hyperparameters to optimize performance. The core of the methodology is a supervised learning approach, wherein labeled data consisting of mammographic images and their corresponding segmentation masks are used to train the model (LeCun, Bengio and Hinton, 2015). Supervised learning is particularly suitable for this task because it enables the model to learn directly from examples of the desired output, thereby improving its ability to accurately identify and delineate masses in new, unseen images.

Central to this design is the implementation of a Convolutional Neural Network (CNN) architecture, which is well-known for its capacity to handle image data efficiently (Krizhevsky, Sutskever and Hinton, 2012). Specifically, the U-Net architecture is employed, given its established effectiveness in biomedical image segmentation tasks (Ronneberger, Fischer and Brox, 2015). U-Net’s architecture, characterized by its encoder-decoder structure and the inclusion of skip connections, allows it to capture both local and global features of the image, which is crucial for the accurate segmentation of complex structures such as mammographic masses. The choice of U-Net is motivated by its ability to produce high-resolution segmentation maps while maintaining a high degree of accuracy, which is essential for clinical applications where precision is paramount. This experimental design, therefore, aligns with the overarching goal of achieving a model that can reliably and accurately segment masses in mammograms, ultimately contributing to the early detection and treatment of breast cancer (Szegedy *et al.*, 2017).

**3.3 Data Collection**

The dataset utilized in this research is composed of mammographic images that are categorized into two primary classes: benign (1,945) and malignant (1,976). Each mammographic image is accompanied by a corresponding mask, which delineates the segmented area of the mass. This dataset forms the foundation for training and evaluating the segmentation model, providing the necessary labeled data for a supervised learning approach.

*Data Characteristics:* The dataset is organised within a directory named data\_mass, which is structured into two subfolders: Benign and Malignant. Each of these subfolders contains .png format images, representing the mammograms, along with their corresponding mask files. The mask files follow a specific naming convention, with \_MASK.png appended to the base filename of the associated mammogram. This systematic organization facilitates the efficient loading and processing of data, ensuring that each image is correctly paired with its respective mask during the training and evaluation phases.

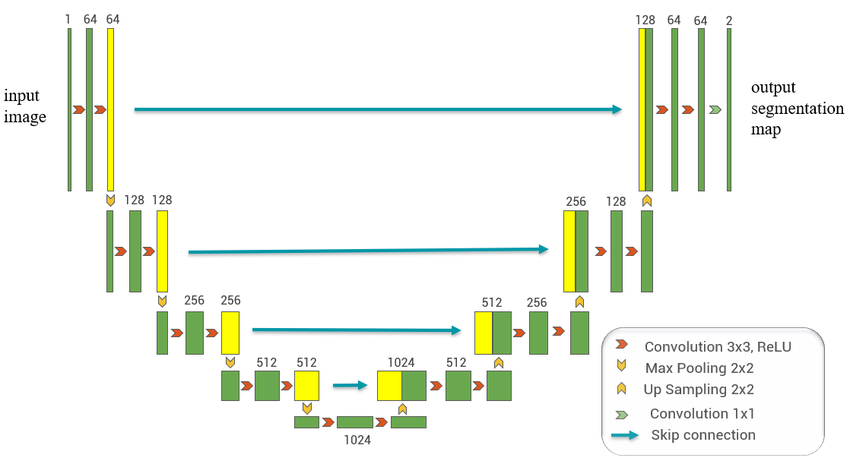
*Preprocessing:* To prepare the dataset for model training, several preprocessing steps are employed. Initially, all images are resized to a uniform dimension, which is crucial for ensuring consistency across the dataset and compatibility with the neural network input requirements. This resizing step standardizes the input size, thereby simplifying the model architecture and improving computational efficiency. Additionally, normalization is applied to the images, which involves scaling the pixel values to a common range. Normalization is essential for enhancing the model's convergence during training, as it helps to stabilize the learning process by reducing the variance in pixel intensity values.

*Data Augmentation:* data augmentation techniques are implemented to increase the diversity of the training data, which is particularly important in medical imaging tasks where the availability of labeled data can be limited. Augmentation strategies such as rotation, flipping, and zooming are applied to the images, generating additional samples that help the model generalize better to unseen data. These techniques not only expand the dataset but also introduce variability that simulates real-world conditions, thereby improving the robustness of the model.

*Data Splitting:* The preprocessed dataset is then partitioned into three distinct subsets: training, validation, and test sets. The training set is used to fit the model, the validation set is employed to tune hyperparameters and prevent overfitting, and the test set is reserved for the final evaluation of the model's performance. This stratification ensures that the model is trained and evaluated on separate data, providing an unbiased assessment of its generalization capabilities. By following these systematic data collection and preprocessing procedures, the research ensures that the model is developed and evaluated in a rigorously controlled and scientifically sound manner, laying the groundwork for accurate and reliable segmentation of mammographic masses.

**3.4 Model Architecture**

The U-Net architecture is selected as the foundational model for the segmentation of mammographic masses, primarily due to its encoder-decoder structure, which has demonstrated superior performance in biomedical image segmentation tasks. The U-Net architecture is particularly well-suited for this task because it effectively combines low-level feature information with high-level contextual information through the use of skip connections. These skip connections directly link corresponding layers in the encoder and decoder, allowing the model to preserve spatial information that is critical for accurate segmentation, especially in medical images where fine details can be diagnostically significant.



**Firgure1: U-Net architecture** (Guo *et al.*, 2020)

Given the specific challenges posed by mammographic mass segmentation, modifications to the standard U-Net architecture was necessary to enhance its performance. These modifications included adjusting the number of filters in each convolutional layer, altering the depth of the network, or incorporating additional regularization techniques such as dropout or batch normalization. The goal of these modifications is to tailor the U-Net architecture to the unique characteristics of mammographic images, thereby improving the model's ability to accurately segment masses, regardless of their size, shape, or location.

**3.5 Training the Model**

The training of the U-Net model is conducted using the preprocessed mammographic dataset, with careful consideration given to the optimization of key hyperparameters such as learning rate, batch size, and the number of epochs. The learning rate determines the step size at each iteration while moving toward a minimum of the loss function, and its optimization is critical for ensuring that the model converges to an optimal solution without overshooting. Batch size, which refers to the number of training samples utilized in one forward/backward pass, is also optimized to balance memory constraints and model performance. The number of epochs, representing the total number of passes through the entire training dataset, is adjusted to ensure sufficient training without overfitting.

The Adam optimizer, a widely-used optimization algorithm, is employed to minimize the selected loss function. Adam is chosen for its adaptive learning rate capabilities, which allow for faster convergence and better performance compared to traditional optimization algorithms. To further enhance the training process and prevent overfitting, early stopping and model checkpointing techniques are utilized. Early stopping monitors the model's performance on the validation set and halts training when no improvement is observed, thereby preventing the model from overfitting to the training data. Model checkpointing saves the best model weights during training, ensuring that the most effective version of the model is preserved for further evaluation and deployment.

***Validation***

**The model is validated using a dedicated validation set, which is not seen by the model during training. This validation set is crucial for fine-tuning the model's hyperparameters and for assessing its generalization capabilities. By evaluating the model on this separate dataset, overfitting can be monitored, ensuring that the model does not merely memorize the training data but instead learns to generalize to unseen data. The validation process involves continuous monitoring of the model's performance across the aforementioned metrics, allowing for iterative improvements and adjustments to the model architecture and training parameters.**

***Testing***

**After fine-tuning and selecting the best-performing model based on validation results, the final model is evaluated on the test set. The test set is entirely separate from both the training and validation sets, ensuring that the model's performance is assessed in a realistic and unbiased manner.**

**3.6 Evaluation Metrics**

**The evaluation of the U-Net model's performance in the segmentation of mammographic masses is conducted using a comprehensive set of metrics designed to assess various aspects of the model's accuracy and robustness. The primary metrics utilized include the Dice Similarity Coefficient (DSC) and Intersection over Union (IoU). Each of these metrics provides unique insights into the model's performance, allowing for a holistic evaluation of its segmentation capabilities.**

***Dice Similarity Coefficient (DSC):* The DSC is a crucial metric in medical image segmentation, as it measures the overlap between the predicted segmentation and the ground truth. It is particularly effective in evaluating the accuracy of the segmented regions, especially in tasks where the focus is on the precision of boundary delineation. A higher DSC indicates a greater degree of similarity between the predicted and actual segmented areas.**

***Intersection over Union (IoU):* IoU, also known as the Jaccard index, is another key metric that evaluates the accuracy of the segmentation by measuring the ratio of the intersection of the predicted and ground truth areas to their union. This metric is useful for assessing the overall coverage of the segmented region and complements the DSC by providing additional information about the spatial accuracy of the segmentation.**

**3.7 Implementation**

**3.7.1 Software and Tools**

**This project was developed using Python, with a focus on state-of-the-art libraries and frameworks tailored for deep learning and image processing tasks. The following software and tools were integral to the successful implementation of the U-Net model for image segmentation:**

***Python:* The primary programming language used for this project, chosen for its extensive support for scientific computing and machine learning.**

***TensorFlow/Keras:* TensorFlow served as the core deep learning framework, providing a robust platform for model development, training, and deployment. Keras, a high-level API within TensorFlow, was specifically utilized for designing and training the U-Net architecture due to its simplicity and flexibility in building complex neural networks.**

***NumPy:* NumPy was essential for handling numerical operations, particularly for the manipulation of large arrays of image data. It enabled efficient processing of images and masks, which are crucial inputs for the model.**

***Matplotlib:* Matplotlib was used for visualizing various aspects of the project, including data distribution, training progress, and model performance. The ability to plot images, training loss curves, and accuracy metrics helped in the interpretation and analysis of the model’s behavior.**

***OpenCV:* OpenCV, a powerful image processing library, was employed for preprocessing tasks such as reading images from disk, resizing, and performing image transformations. These steps ensured that the images were in the correct format and size required by the model.**

***Scikit-learn:* Scikit-learn provided tools for data splitting, model evaluation, and metrics calculation. Specifically, it was used to divide the dataset into training, validation, and test sets, as well as to compute metrics like accuracy and confusion matrices for model evaluation.**

**The entire implementation was carried out in a Jupyter Notebook environment, which is particularly suited for exploratory data analysis and iterative model development. This environment allowed for the seamless integration of code, visualizations, and documentation, facilitating a smooth workflow for the project.**

**3.7.2 Code Structure**

**The codebase of the project is methodically organized into several key sections, each dedicated to a specific phase of the implementation process. This structured approach ensures clarity and modularity, making it easier to understand and modify the code if needed.**

***Loading Libraries***

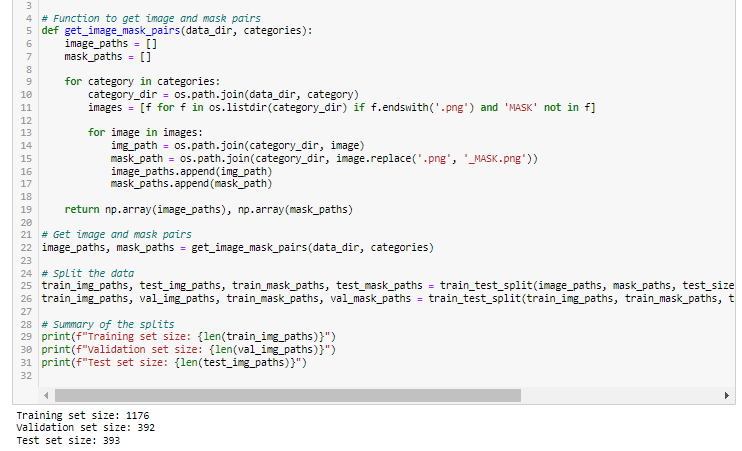
**The notebook begins with importing all necessary libraries, setting the foundation for the tasks ahead. Libraries such as TensorFlow, Keras, NumPy, Matplotlib, and OpenCV were imported in this section, along with any additional utilities required for data handling and visualization. The inclusion of these imports at the beginning ensures that all dependencies are loaded before any operations commence, avoiding potential interruptions in the workflow.**

***Data Preparation***

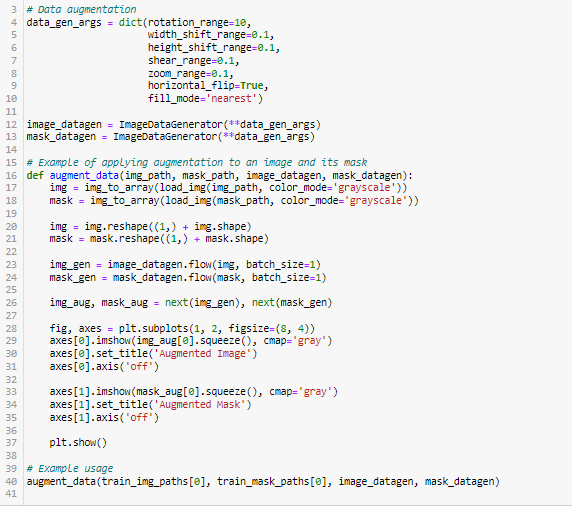
***Loading and Preprocessing Data:* The raw image data, along with corresponding masks, was loaded from the dataset using OpenCV. This step involved reading the images from disk, converting them to an appropriate format, and resizing them to match the input dimensions required by the U-Net model.**



***Splitting the Data:* The dataset was split into three distinct subsets: training, validation, and test sets. This was done using the train\_test\_split function from Scikit-learn, ensuring that the model was trained on one part of the data, validated on another, and tested on unseen data. The split was performed in a stratified manner to maintain the distribution of classes across the subsets.**

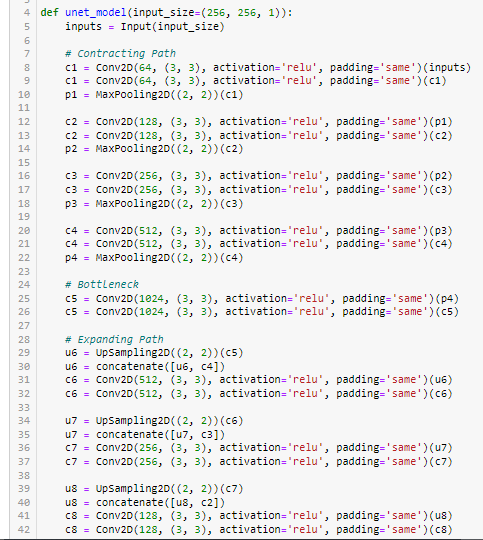


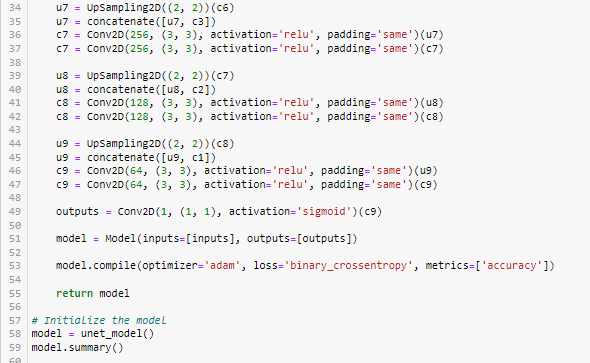
***Data Augmentation:* To address the challenge of limited training data and improve the model’s robustness, data augmentation was applied. This involved generating new training samples by applying random transformations such as rotations, translations, zooming, and horizontal flipping. TensorFlow's ImageDataGenerator was used for this purpose, allowing for real-time augmentation during model training.**



***Model Design***

***U-Net Architecture:* A U-Net model was designed for the task of image segmentation. The model architecture consisted of an encoder-decoder structure with symmetrical skip connections between corresponding layers of the encoder and decoder. The encoder was responsible for capturing contextual information through a series of convolutional and max-pooling layers, while the decoder reconstructed the image segmentation map using up-sampling and concatenation layers. The model was constructed using Keras' functional API, allowing for a flexible and modular design. This approach facilitated the addition of custom layers and ensured compatibility with various input sizes.**





***Model Training***

***Compilation:* The model was compiled with an appropriate loss function (binary\_crossentropy for binary segmentation tasks) and an optimizer (Adam) to minimize the loss during training. Metrics such as accuracy and Intersection over Union (IoU) were also tracked to monitor the model's performance.**

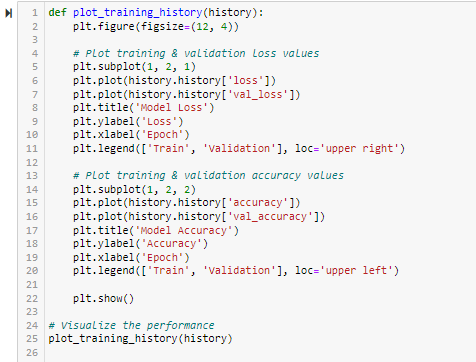
***Training Process:* The model was trained on the prepared dataset using the fit method in TensorFlow. The training process was enhanced by incorporating callbacks such as ModelCheckpoint and EarlyStopping. ModelCheckpoint was used to save the best-performing model during training, while EarlyStopping halted the training process when the validation loss ceased to improve, preventing overfitting.**



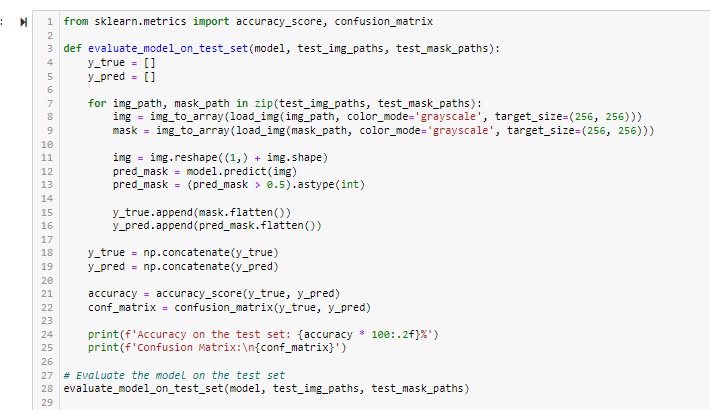
**The training history, including loss and accuracy for both training and validation sets, was recorded and later visualized to analyze the model’s learning curves.**

***Model Evaluation and Visualization***

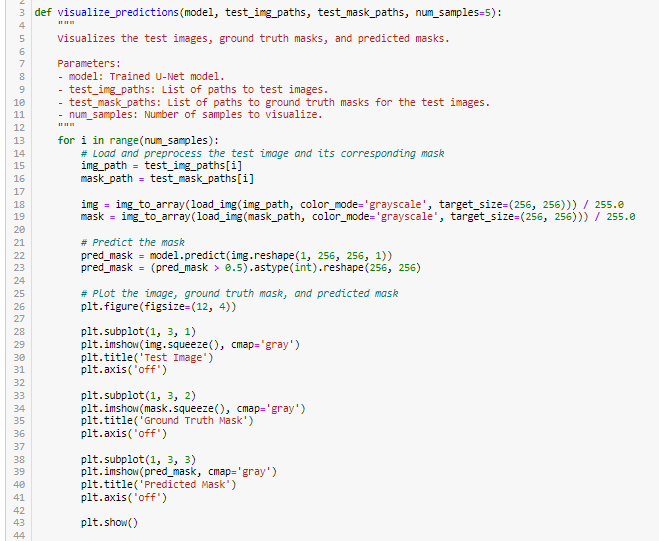
***Plotting Training History:* The training and validation loss curves were plotted to examine the model’s convergence. These plots helped in diagnosing issues such as overfitting or underfitting, guiding potential refinements to the model or training procedure.**



***Quantitative Evaluation:* Post-training, the model was evaluated on the test set. Metrics such as accuracy, confusion matrix, and IoU were computed to quantitatively assess the model’s performance. These metrics provided insights into the model’s ability to generalize to unseen data.**



***Qualitative Evaluation:* Beyond numerical metrics, qualitative evaluation was performed by visualizing the model’s predictions on test images. This involved overlaying the predicted segmentation masks onto the original images, allowing for a visual assessment of the segmentation quality.**



**3.8 Chapter Summary**

This chapter provided a detailed account of the methodology employed in the segmentation of mammographic masses. The next chapter will present the results of the model training and evaluation, followed by a discussion of the findings.

**REFERECE**

Guo, Y. *et al.* (2020) ‘Cloud detection for satellite imagery using attention-based U-Net convolutional neural network’, *Symmetry*, 12(6), p. 1056.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ‘Imagenet classification with deep convolutional neural networks’, *Advances in neural information processing systems*, 25.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) ‘Deep learning’, *nature*, 521(7553), pp. 436–444.

Litjens, G. *et al.* (2017) ‘A survey on deep learning in medical image analysis’, *Medical image analysis*, 42, pp. 60–88.

Ronneberger, O., Fischer, P. and Brox, T. (2015) ‘U-net: Convolutional networks for biomedical image segmentation’, in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, pp. 234–241.

Szegedy, C. *et al.* (2017) ‘Inception-v4, inception-resnet and the impact of residual connections on learning’, in *Proceedings of the AAAI conference on artificial intelligence*.