AUTOMATING THE SEGMENTATION OF X-RAY IMAGES WITH DEEP NEURAL NETWORKS

s231866, Yash Vagadia & s103629, Hans Christian Hansen

DTU
Anker Engelunds Vej 101, 2800 Kongens Lyngby

ABSTRACT

This project focuses on automating the segmentation of X-ray images using deep neural networks using the U-Net architecture. This work aims to address manual segmentation's time-consuming and error-prone nature by leveraging the power of convolutional neural networks. The U-Net model is known for its effectiveness in image segmentation[1] and therefore implemented to analyse tomographic X-ray datasets. The model's performance is evaluated using accuracy, precision, recall, and F1 score metrics. Additionally, a visual analysis is presented through a collage of images, showcasing the model's ability to identify and segment different regions of interest. The results indicate high accuracy and precision, demonstrating the model's proficiency in automating the segmentation process.

Index Terms— Deep Learning Neural Networks, X-ray segmentation, U-Net, Image Analysis, Convolutional Neural Networks, Automation, Medical Imaging, Tomographic X-ray Data, Image Segmentation, Performance Metrics, Precision, Recall, F1 Score, Accuracy

1. INTRODUCTION

The rapid development in X-ray has led to the analysis of tomographic X-ray data sets becoming increasingly critical in various fields like scientific, medical, and industrial applications. However, despite technological advances, the manual segmentation of raw data remains an issue. Not only is this task time-consuming, but it also introduces errors and subjectivity in the analysis.[1]
Automating the segmentation process will eliminate the errors and subjectivity introduced by human intervention, and it will also ensure that this process isn't stopping the development; instead, it will be able to keep up with the development of the X-ray technology and thereby continue to support scientific discoveries and insight for the industrial.
This project aims to take advantage of the power of deep neural networks to automate the segmentation process, thereby removing the manual work and speeding up the process. The project will take an advance on an existing architecture named U-Net, which is a convolutional neural network with a symmetric encoder-decoder structure, making it ideal for image segmentation. Its architectures enable it to handle spatial relationships and capture contextual information effectively, which is crucial for image segmentation in the medical industry. It can work with limited data, and its precise localisation makes it ideal. [2]
The report will go through the process of developing a deep neural network, from looking into the model description, design and construction of the code followed by results and comparison, a walk-through of the program architecture and finally, a conclusion of the project.

2. MODEL DESCRIPTION

The objective of the project is to develop and train a deep neural network based on the U-Net architecture for the segmentation of X-ray images. It was originally introduced in biomedical image segmentation. Still, U-Net has proven versatile and is now applied in various computer vision tasks - including the segmentation of psychographic X-ray images.[2]
The U-Net architecture is characterised by its U-shaped structure, consisting of a contracting path (encoder) and an expansive path (decoder), as depicted in Figure 1.

The contracting path captures context and extracts hierarchical features through convolutional and pooling layers, progressively reducing spatial dimensions while amplifying feature representation. The expansive path, responsible for precise localisation, employs transposed convolutions and skip connections, facilitating the reconstruction of high-resolution segmented output. The skip connection in the expansive part is a key feature of the U-Net. These connections allow the network to retain the fine-grained details from previous layers and combine them with broader contextual information in later stages. This results in more accurate segmentation results compared to conventional neural networks, which may lose spatial information during downsampling[4].

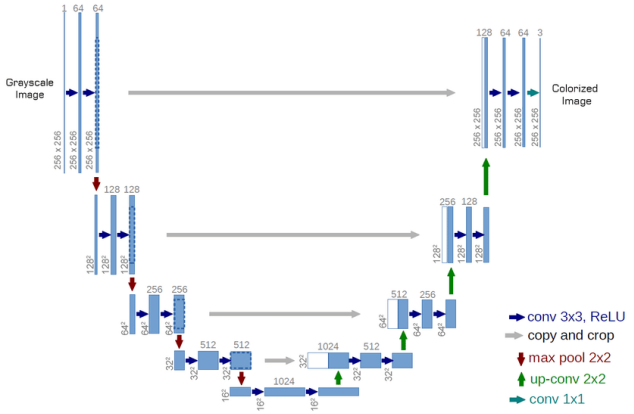

Fig. 1. Example architecture of U-Net.[3]

The real-world X-ray image datasets used in this project consist of raw and labelled images, where the predicted data from the U-Net will be benchmarked against manually labelled datasets to evaluate the performance of the trained UNet model. The benchmarking process is essential in assessing the model's accuracy and effectiveness compared to manual label images.
Figure 2 shows comparison between an example of a raw image that is used as input to the U-Net model, while the labeled image next to it shows what the result is benchmarked against.

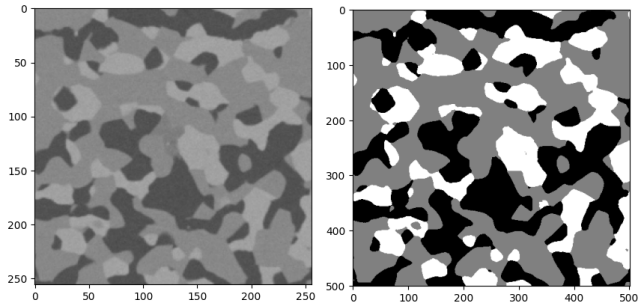

Fig. 2. Example of raw data vs. labeled data

3. DESIGN AND CONSTRUCTION

3.0.1. Model Configuration

The model is based on a U-Net architecture, a type of convolutional Neural Network (CNN) comprising an encoder (downsampling) and a decoder (upsampling). The encoder consists of several downsampling paths, each of which includes two convolutional layers followed by a max pooling layer. Each of the convolutional layers makes use of the ReLU activation, and the number of filters increases with each block. The bottleneck function, located at the end of the downsampling path, consists of two convolutional layers - this is the part of the model that allows for learning more complex features. The decoder, or the upsampling path, consists of 'up blocks', each of which includes an upsampling layer followed by two convolutional layers. The upsampling layer increases the size of the feature maps, and a concatenation operation combines these upsampled feature maps with the feature maps from the corresponding down block. This process of copying and concatenating feature maps from the downsampling path, also known as skip-connection, is a unique feature of the UNet and allows the network to use information from multiple resolutions. Lastly, the output layer is a convolutional layer with a sigmoid activation function, which maps the final feature representation to the desired output segmentation.
The model is compiled with the Adam optimizer and a custom Dice Loss function. The metrics used to evaluate the model's performance during the training are the Dice coefficient and a custom quantized mean squared error (MSE) function.

3.0.2. Loss Function and Metrics

The loss function used in this project is a custom Dice Loss, implemented as a class that inherits from the Keras loss library. The loss function measures the overlap between the predicted segmentation and the truth. It's a slightly modified version of the Dice lose, where the denominator includes sum of the squares of the predicted and true masks, raised to the power of gamma. This modification will penalize the model more heavily for incorrect predictions, which might lead to an improvement in the performance. The DiceLose class make use of a custom Mean Squred Error (MSE) metric, the function quantize the true and predicted into three levels: 0.0, 0.5 and 1.0. It will then calculate the MSE between the quantize true and predicted values. The result is then subtracted from 1 and multiplied by 100 to give a percentage score.

3.0.3. Data Preprocessing

In this project, a data loader is used which is a custom data generator that is designed to load batches of images and their corresponding masks from the disk during the training of the U-Net model. The images is resized in this process to a specified size in this case 256x256. Subsequently, normalization is performed where the pixel values of the images and masks are normalized to the range [0,1] by dividing the pixel values by 255, which is the maximum pixel value for an 8-bit image. For this specific task, no data augmentation has been deemed necessary. All images, both raw and labeled, are in the same format and correctly oriented, thereby eliminating the need for operations such as flipping.

3.0.4. *Hyperparameter Tuning*

The model's hyperparametrs - learning rate, batch size and number of epochs was tuned to optimize performance. Each of these parameters was found with trial and error, which was done because of problems with the resource allocation in co-lab.
A summary of the hyperparameter tuning results can be found in table 1

Learning rate	Batch size	epochs
XX	4	10

Table 1. Table for learning rate, batch size and epochs

3.0.5. *Training Process*

The training process involves feeding batches of images to the network, evaluating the output using Dice Loss, and adjusting parameters to minimize loss. Model weights are saved post-training for future use.

3.0.6. *Hardware and Software Environment*

Implemented in Python using TensorFlow and Keras libraries, the model runs on Google Colab, which provides a cloud-based Python notebook environment and access to GPU resources for training deep neural network. Data is stored on Google drive, and accessed directly from the Colab notebook.

4. RESULTS AND DISCUSSION

4.1. Results

This section covers the result and comparison for the model, utilizing visualizations of images, graph, and tables with metrics for the results. The visualization offer a direct insight into how the model handle the task, while the graph depicts the learning and Loss rate's. Finally, the metrics presented in the table serve to reinforce and substantiate the model's performance.

Figure 3 shows the effectiveness of U-Net in automating the segmentation of X-ray images. The left panel display the raw image, which serve as the input for the U-Net model. The middle panel is the segmented image produced by the U-Net. The last image to the right is the labeled image, which is used for comparison and evaluation of the model's performance.
As can be seen, theirs a close resemblance between the segmented image and the labeled images, which indicates the accuracy of U-Net in capturing the details precise segmentation.

The graph shown in Figure 4 shows the accuracy and loss for both training and validation over several epochs. The learning rate indicates the pace which the model is learning

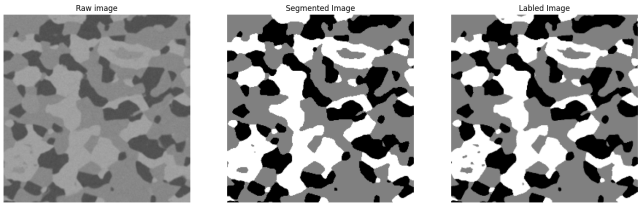

Fig. 3. Example of a result from the U-Net model, where a visual comparing for the raw image, segmented and the labeled image

during the training process. A high learning rate indicate the model's parameters is being updated rapidly, potentially leading to faster convergence, but with the risk of overshooting the optimal solution. A low learning rate implies more gradual, which could lead to more precise convergence but over a longer time. The loss rate graph shows how the model's performance, as measured by the loss function, improves over time. If the loss rate decreases it indicates that the model is making successful predictions and improving its performance on the training set. If the loss rate plateaus or increases it could indicate overfitting or that the model has reached its potential for learning from the given data.

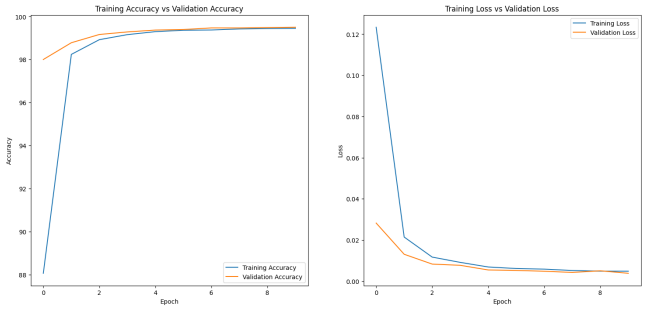

Fig. 4. Accuracy & Loss for the model

From the graph it can be observed that the training accuracy starts increasing rapidly and then follows a more steady increase, indicating effective learning from the training data. The vaidation data on the other hand, shows a stable and high accuracy, which could mean that the model is generalizing well to unseen data.
For the training loss, a consistent decrease is observed, indicating that the model is making successful predictions and is improving upon the training data. The same is true for the validation loss, its the same decrease that can be seen here, which indicates the the models performance on the validation data is improving.

Figure 5 presents a collage of nine images arranged in a 3x3 grid, with each column depicting different stages of the image analysis process. The first column displays the original

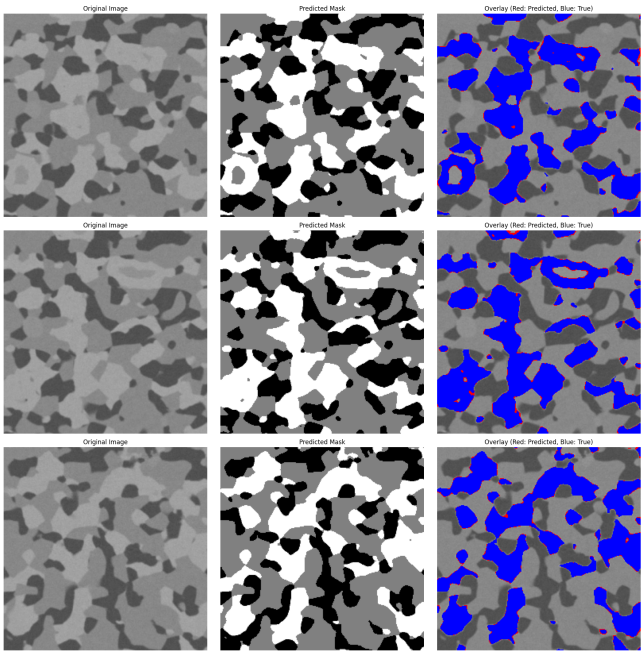

Fig. 5. Visualize images of three different images arranged in a 3x3 grid, where the raw data is to the left, the segmented/predicted data in the middle and labeled data to the right collected to show the predicted compared to true.

images, which serve as the raw input data for the model. The middle column, in black and white, represents the predicted mask that are the output after analysing the original image; the white areas are those marked by the model as areas of interest. The last column presents an overlay of the predicted (in red) and true (in blue) masks on the original images, providing a direct visual comparison between the model's predictions and the actual truth.

This visualization offers an intuitive way to see the model's performance and identify areas of improvement. As can be seen, the model's does well in identifying areas of interest.

The Confusion matrix[5] in table 2 give a detailed view of the model's performance. It can be seen that the model have identified a large number of positives and negative. However, there are also a significant number of false negative, where the model incorrectly predicted a negative outcome, when it was actual positive. This is further supported by the performance metric seen in table 3, here the *accuracy*, *precision*, *Recall* and *F1 score* have been calculated to provide additional info about the performance of the U-Net[5].

Total population	Positive (PP)	Negative (PN)
Positive (P)	941354	4331
Negative (N)	876650	1454465

Table 2. Confusion Matrix

The *accuracy* for the model is 0.7311, meaning that 73.11% of the prediction made by the model were correct. This is a descent score, but theres room for improvement. The *precision* for the model is quite high, a score of 99.70% which means that when the model predict a positive outcome its correct 99.70% of the time. *Recall* or *sensitivity* have a score of 0.6239, which means that the model correctly identified 62.39% of all actual positive outcomes. Compared to *precision*, then *Recall* have a relative low score, which could mean that the model is missing a significant number of positives outcome.

An F1 score of 76.75 indicates that the model have a reasonably good balance of precision and recall, but with room for improvement, especially in term of recall (reducing false negative). All in all the model shows a strong performance in terms of precision but falls in term of recall. The high number of false negatives suggest that the model might be to ó conservative in predicting positive outcomes.

Metric	Score	In percent
Accuracy	0.7311	73.11
Precision	0.9970	99.70
Recall	0.6239	62.39
F1 Score	0.7675	76.75

Table 3. Performance score for the U-Net model

5. CONCLUSION

In conclusion, the implementation of the U-Net architecture for automating the segmentation of X-rat images has shown good results. The model demonstrates its strong capability to accurately identify and segment regions of interest within the images. The visual comparisons and performance metrics, including the accuracy, precision, recall and F1 score, highlights the model's effectiveness in automating a manual and time-consuming process. This project shows the force of deep neural networks using U-Net.

6. REFERENCES

[1] Reuben Dorent Robert Bradford Shakeel Saeed Sotirios Bisdas Sebastien Ourselin Jonathan Shapey Tom Vercauteren Hari McGrath, Peichao Li, "Manual segmentation versus semi-automated segmentation for quantifying vestibular schwannoma volume on mri," 2020.

[2] the free encyclopedi Wikipedia, "U-net," 2018.

[3] Kamyar Nazeri and Eric Ng, "Image colorization with generative adversarial networks," 2018.

[4] Nikolas Adaloglou, "An overview of unet architectures for semantic segmentation and biomedical image segmentation," 2021.

[5] the free encyclopedi Wikipedia, "Confusion matrix," 2023.