JOURNAL OF
EMERGING INVESTIGATORS

Article

Unlocking robotic potential through modern organ segmentation

Ansh Chaudhary[1], Robail Yasrab[2]

[1] Saratoga High School, Saratoga, California

[2] School of Clinical Medicine, University of Cambridge, Cambridge, United Kingdom

SUMMARY

Deep learning has revolutionized the approach to complex, data-driven problems, specifically in medical imaging, where its techniques have significantly raised efficiency in organ segmentation. Enhancing the depth and precision of organ-based classification is an essential step towards automation of medical operations and diagnostics. Our study aimed to investigate the effect and potential advantages of different models using Binary Semantic Segmentation. We chose to employ the SegFormer model as our primary deep learning model because of its lightweight architecture, alongside different Unet variations. We hypothesized that the performance of the SegFormer model would surpass the different Convolutional Neural Networks (CNN) models. We assembled a custom 2D computerized tomography (CT) scan dataset CT-Org2D, through conversion from 3D volumes and placing them in their respective folders. In contrast to the selected models, several experiments showed the task's simplicity required a redesigned Unet architecture with reduced complexity. This redesigned model yielded impressive results: the precision, recall, and Intersection over Union (IoU) scores were 0.91, 0.92, and 0.85, respectively. Our research could be improved upon by utilizing more diverse datasets, optimizing the model's architecture, and conducting additional experiments with more advanced resources.

INTRODUCTION

Around 310 million major surgeries are performed each year, all of which carry a substantial risk of unfavorable results. Several major issues may arise due to decreased accuracy and precision during these operations. For example, larger incisions and a surgeon's limited range of motion when accessing certain areas can result in more painful scarring (1). Besides the potential for mistakes, these surgeries are also very expensive, not only for the average patient but also for hospitals. By 2025, hospital expenses are projected to average $40 million annually when accounting for medical staff, equipment, and other costs (2). AI-based methods have recently enhanced human work at the intersection of medical and AI fields. The use of automation in surgery, including robotic assistance, has the potential to significantly benefit medicine by ensuring consistent precision and lowering surgical expenses (3).

Convolutional Neural Networks (CNN) are advanced networks designed to effectively recognize patterns across different parts of an image in the form of features (4). Unlike traditional neural networks, which are slower and less effective at capturing these features, CNNs show promising results because of their three layers: convolutional, pooling, and fully connected layers. The convolutional layer applies filters to scan the image and detect certain features (4). The pooling layer then reduces the spatial dimensions of these features, reducing overall complexity (4). Lastly, the fully connected layer connects each neuron from one layer to every neuron of the next layer, resulting in a dense network that allows for complex pattern recognition (4). However, CNNs have their limitations, especially in the task of organ segmentation. One study attempting to use CNN for radiology classification noted that the model demanded a large pool of annotated data for proper training (5). However, this is a common problem in medical tasks as there is little annotated medical imaging data available to the public to develop a classification tool. Another noted challenge was overfitting, which occurs when the model begins to learn irrelevant patterns, impairing accuracy when the model is applied to a new dataset.

To address these challenges, researchers employed the transformer architecture, which has been primarily used in Natural Language Processing (NLP). This architecture has two important features: self-attention mechanisms and parallel processing. The self-attention component assigns different weights of importance to various features in an image, also known as long-range dependencies (6). Thus, the model focuses on more important parts, allowing it to capture relationships much more efficiently from less data while preventing overfitting. The second feature, parallel processing, allows the model to simultaneously work on various pieces of information, similarly to how multiple people working on different parts of a project can speed up the process. Because of the huge size of transformers, their training times are typically longer than CNNs (6). Despite this, we hypothesized that transformers we used would achieve a higher accuracy than the CNNs due to their intricate and robust architecture.

In our study, we implemented two types of models: Unet, a type of CNN, and SegFormer, a type of transformer (**Figure 1**). Unets are characterized by a U-shaped structure with encoder and decoder paths (7). Encoders are analogous to a magnifying glass that zooms in and focuses on certain parts, while decoders zoom out to synthesize these details together and capture their essential parts. Because of its efficient features, Unet is used in a variety of biomedical applications such as MRI reconstruction and monitoring chronic wounds (8, 9). Conversely, instead of convolutional layers, SegFormer uses transformer blocks, which are fundamental components of its architecture that facilitate input data. These blocks prevent

https://doi.org/10.59720/23-323

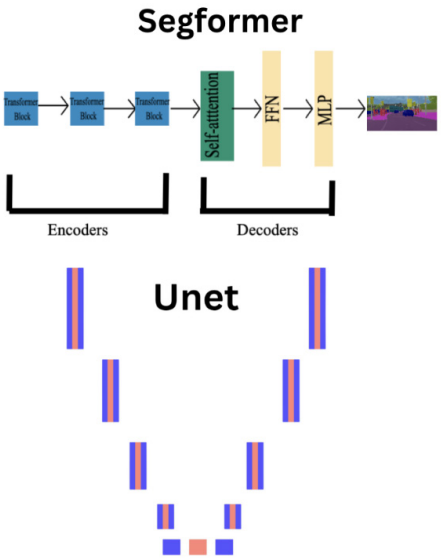

Figure 1. A visual representation of the Segformer and Unet model architectures. The Unet decreases the image size by half, affecting the resolution, but restores it to its original size through upsampling and downsampling functions. The Segformer uses of encoder and decoder components alongside its transformer blocks to predict the output.

the model from using higher-complex decoders, resulting in a more lightweight model. Instead, SegFormer uses multi-layer perceptron (MLP) decoders that are used for simpler feature relationships to combine information from different layers in the model (8). SegFormer was chosen for this task because of its successful functionality in semantic segmentation (10). Even though SegFormer is five times smaller compared to a typical CNN, Xie *et al.* observed notably improved performance in its ability to combine both local and global attention (10). Through local and global attention, the model can create efficient representations for segmentation.

We added four different types of backbones onto the Unet: Vgg16, Vgg19, Resnet50, and a customized Unet without a backbone. Backbones are the basic architectures of models that decide the structure, complexity of arrangement, and number of layers. The Vgg architecture captures detailed features through its convolutional layers at different levels of abstraction (11). Vgg16 and Vgg19 differ in the number of layers present. Resnet includes skip connections between layers that enable a deeper architecture, since the skip connections help prevent vanishing gradients, the signals used to adjust the weight becoming too small (12). Additionally, we made a customized Unet that was not dependent on any pre-existing backbone architecture. We chose to study many Unet backbones to enable a thorough assessment of this as opposed to only one.

Considerable research has explored the effectiveness of different models in a variety of tasks: Chen *et al.* propose TransUnet, a hybrid architecture combining a Unet and a transformer and taking advantage of the benefits of each architecture for medical imaging (13). The TransUnet showed promising results on organ segmentation in CT scans with a high average Dice Score Coefficient (DSC), demonstrating the effectiveness of a hybrid mode (13). However, this study could not explore of each architecture's strengths and weaknesses, as it combined the Unet and transformer into a single model. Hence, a comparative analysis is necessary to identify which model works best for each use case and how each model can be improved. Additionally, their data lacked variations in image quality and conditions including the resolution, lighting, and clarity. Moreover, Zettler *et al.* showed that 2D Unet models are more effective than 3D Unet models in terms of speed and low memory costs (12). Although the 3D model had slightly more favorable results than the 2D model, the authors concluded that this could not justify the additional computer resources needed (14). The 3D model also required 3D image data, which further increased memory usage and rendering time (14). Thus, we decided it was best to work with 2D models. Similarly, Dia *et al.* used transformer-based architectures for multi-modal medical imaging classification, which involved labeling data from multiple sources of information (15). The authors discussed their struggle to solve the lack of sufficient medical imaging data, which is needed to work with transformers (15). In the end, they opted to address this problem by creating a new model that works with the data they found (15). Here, we chose to directly modify the dataset instead of the models for our project for our project.

Automating medical procedures requires organ segmentation, separation, and identification. This project used binary semantic segmentation, in which each pixel is labeled according to two classes: either an organ or the background. The goal in this task is to create a binary mask that separates the target area from everything else. To simplify this process, the method uses downsampling to reduce the image size and thus the amount of data. While at the end, upsampling is used to restore the image to its original size. This is done through interpolation which creates new pixels with values estimated from neighboring pixels to restore the original image size.

Transformers have recently enabled improvements in computer vision with their ability to execute tasks more efficiently than a CNN (6). This motivated us to research the strengths and weakness of transformers on semantic segmentation and hypothesize that the transformer would produce the most efficient results in this organ segmentation task. However, the data shown in this study does not support this hypothesis, as a simpler model outperformed all other transformer and CNN models we tested. Thus, for certain tasks, simpler models may offer significant advantages in terms of efficiency and performance, showing the potential in medical imaging analysis.

RESULTS

To compare the performances of the transformer and Unet models on segmenting organs, we utilized several key metrics: precision, recall, and intersection over union (IoU) score. Due to the task's focus on binary segmentation, the IoU score served as the benchmark metric for comparison. The values for these metrics range from 0 to 1, representing the probability in different aspects of the task. For example, precision is the number of true predictions relative to the entire set of outcomes. To guarantee that non-organ regions are excluded from the segmentation, precision is essential. Recall is the ratio of actual positive results that the model accurately predicts. High recall is essential for segmenting organs since it guarantees thorough coverage of the entire organ. Insufficient recollection may result in insufficient segmentation, which may produce incorrect representations of the organ's

https://doi.org/10.59720/23-323

size and shape. The IoU score is the overlap between the predicted regions and the ground truth. It is the most reliable metric for accurate binary segmentation (16).

We believed that the SegFormer model would yield the highest results in the task of binary segmentation of organs due to its intricate architecture. We tested this by recording the results of each model through different metrics. For the SegFormer, we observed relatively low scores for precision (0.56), recall (0.33), and IoU (0.26). Comparatively, the Unet models with backbones had scores around 0.95 for precision, 0.38 for recall, and 0.37 for IoU. Thus, we inferred these bigger models had trouble adjusting to this task, as seen through the successful scores from the custom Unet. Alternatively, the data itself could've been the reason for these models performing worse **(Table 1)**.

Looking at the models, the SegFormer model had the largest size (180 MB), with the greatest number of parameters (47 million) relative to the other Unets (2.6 million) **(Table 2)**. Additionally, it had the fewest layers (5 layers), validating its efficient architecture **(Table 2)**. Therefore, the model's poor performance in precision could be attributed to the lower number of layers. The Unets with backbones were only successful in the precision category, suggesting that they failed to localize many of the objects. On the other hand, the custom Unet was the smallest in size (2.2 MB). Yet, it had the highest balance of all three matrices **(Table 2)**. Additionally, all the models ran for 20 epochs, but the average scores of the models on training data usually converged after 10-15 epochs (**Figure 2**). Therefore, we ran the models for around this range of epochs, as running them longer would prove unnecessary.

Model	Precision	Recall	IOU Score
UnetVgg16	0.96	0.41	0.37
UnetVgg19	0.95	0.35	0.36
UnetResnet50	0.96	0.37	0.39
SegFormer	0.56	0.33	0.26
Custom Unet	0.91	0.92	0.85

Table 1. Various evaluation metric results for all models. Highest precision, recall, and IOU score metrics are presented for all five models. The custom Unet preformed the highest for all three metrics. Conversely, the SegFormer had the lowest performance.

Model	Parameters	Model Size	# of Layers
Unet Vgg16/19	31 million	109 mb	51
UnetResnet50	20 million	79.4 mb	176
SegFormer	47 million	180 mb	5
Custom Unet	550,000	2.2 mb	51

Table 2. Comparative characteristics of the chosen models. The SegFormer model has the greatest number of parameters, resulting in its huge model size. Conversely, the custom Unet had the lowest number of parameters and the smallest model size.

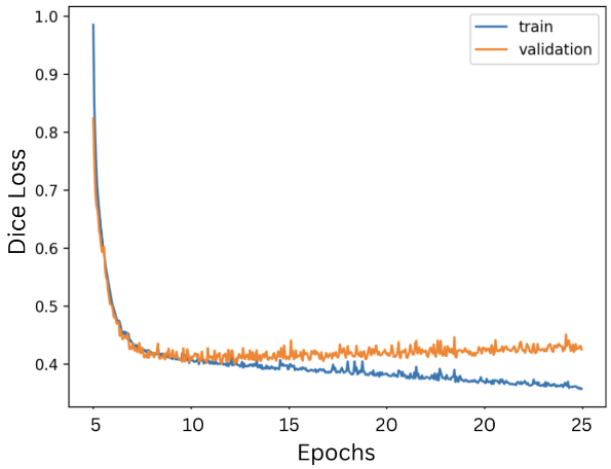

Figure 2. Average convergence rates of all the models across several epochs using a Dice Loss Function. The models converged to a constant loss score by the 10th-15th epoch.

DISCUSSION

We aimed to evaluate the effectiveness of a transformer model, specifically SegFormer, on segmenting organs correctly. We compared the performance of the model to other various Unet models using a custom 2D CT scan dataset. Our experiments showed that the custom Unet, without a backbone, outperformed all the other models through higher precision, recall, and IoU score. The results can suggest that simpler models sometimes offer better performance in less-complex tasks.

We compared the top-performing models from each approach in this segmentation task using the accuracy scores from each metric. However, because of the low accuracy we observed in both the CNN and transformer models, we created a customized Unet with fewer layers and nodes. The model's low complexity was well-suited to the simplicity of the binary segmentation task. Thus, we utilized a Unet lacking a backbone to prevent additional complexity. We also compared the scores from our task to other tasks in medical imaging. Chen *et al.* used a DSC function to find that their project yielded a DSC of 89.71% on average with multiple organs (13). The DSC function is like our IoU score because both measure the degree of overlap, from 0 to 1, between a predicted and truth set. Similarly, we converted our IoU score from our best model of 85% into a dice score, resulting in 83.4%. The slight discrepancy between scores may be attributed to the lack of resources, as higher-performing GPUs and memory may improve model training.

Another aspect discovered in this study is the issue of overfitting, shown through the CNN and transformer models. We found out it was occurring as the model showed a remarkably lower loss and higher accuracy on the training set compared to the validation. A possible explanation could be the tendency of the models to memorize the irrelevant patterns present in the initial and ending segments of each volumetric image. As the 3D volume was converted to 2D slices, these slices represented varying depths in the volume. Consequently, the upper and lower sections of the volume were blank because there were not any organs present in either of these region depths. When the model repeatedly encountered these blank images, it is possible that the model allocated importance and

https://doi.org/10.59720/23-323

memory to them, rather than focusing on the meaningful data. Another reason could be the dataset itself. As we built it, we lacked the time to address class imbalance and the bias that comes as a result. Class imbalance occurs when one class, such as a specific organ, appears more often than others. As a result, the model does not receive enough data on the less-represented classes and may be overfitted on the more common class. In our dataset, the kidney was more common in the slices than the brain or liver. Hence, these organs may have been the most affected by the class imbalance.

Throughout the project, several limitations influenced the research process and results. One significant constraint was the limited computational resources in our work environment, including available RAM and software. The large size of the dataset exacerbated these problems, leading to frequent timeouts and extended execution delays during debugging phases. Therefore, we created a more manageable subset by shortening the number of slices notably to allow for troubleshooting and debugging only, while using the original dataset for training and testing.

Future work could broaden the scope of this study to include multi-segmentation. This would open the doors to more model options, including intricate transformer architectures that may better fit the complexity needs of the task. Another option is the addition of the third dimensions to the task, which could allow the use of 3D datasets. The room for improvement is endless in the field of organ segmentation. The study has the potential to impact the medical field by leading to more effective diagnosis and treatment planning through a custom model and dataset.

MATERIALS AND METHODS

The 3D computed tomography (CT) volume dataset with multiple organ segmentations (CT-ORG) was chosen due to its large amount of data of multiple organ classes and the lack of 2D data online (17). CT-ORG contains six organ classes for segmentation: liver, lung, bladder, kidney, bone, and brain (17). The data was gathered from CT scans of patients who had conditions or lesions in one or more of the organs specified. The scans had a wide variety of parameters, including abdominal and full-body, contrast and non-contrast, and low-dose and high-dose. The front view shows the greatest number of organs with the best definition **(Figure 3)**. In contrast, the top view hides many of the organs, not revealing their true shape. The side view emphasizes the bones, not giving importance to any of the other organs. Thus, we used the front view, as it was the most useful in our case.

To convert the volumes into slices, we condensed the original scans from 512 × 512 with varying depths, ranging from 74 to 987 slices down. We reduced these slices to 285 × 277 by rotating the view from top (axial) to front (coronal) for better organ visibility. This resulted in 512 slices per volume, creating a total of 71,680 slices. The data was split into training, validation, and testing sets (60-15-25%). The reason for this specific data split was to give more data to the training set, allowing the model to learn better with more practice. Periodically, there was a validation set which would test the model using the training data for each epoch. This helps by monitoring the current accuracy while adjusting to the hyperparameters, which leads the model to generalize properly. Finally, the last set was the testing set so the model could be tested on unseen data. Notably, the data was split by volume instead of by individual slice, meaning that all slices for each volume were together in a certain data set (**Figure 4**). The slices were sorted and identified using a specific naming convention (e.g., Volume0_Slice-0).

We simplified the custom Unet by reducing the number of convolutional layers from 23 in its original layer to just 11, which lowered the number of encoder and decoder stages. We also used a uniform number of filter counts, instead of the filters doubling or halving, to minimize the model's size. Lastly, we selectively picked several skip connections, as they were essential to retain spatial information. These changes ensured the Unet could effectively handle this task while lowering its computational resources.

In the new dataset, Ct-Org2D, the features (input) were grayscale, but the labels were RGB. We duplicated the grayscale channel twice so we could work with different models, then we combined all the outputs to get a three-channel image. This allowed us to use the three-channel features and labels with ease while maintaining grayscale images. Alternatively, we could have customized the model's layers to meet the singular channel requirement. The learning rate and batch size differed between the Unets and SegFormer (**Table S1**). After testing many different values for these factors, we were able to identify the optimal settings that maximized performance. In the Unets, the Adam optimizer was chosen over Stochastic gradient descent (SGD) for its adaptiveness to adjust the weights better. However, SGD was used in the SegFormer for its better generalization patterns, improving the accuracy of the model **(Table S1)**.

Google Colab Pro was used for running the code. The software gave access to efficient GPUs such as A100 and NVIDIA Tesla T4 for faster runtimes. With higher memory settings, we had access to 25.5 GB of RAM. For the first half of the study, the PyTorch framework was employed for data collecting and analysis. Pytorch outperformed Keras and TensorFlow in terms of handling huge datasets and overall flexibility. Pytorch, however, demonstrated less incompatibility with debugging, displaying images, and handling complex models, especially transformers (18). Hence, for the training, testing, and visualization phases, we utilized Keras/TensorFlow 2.13.

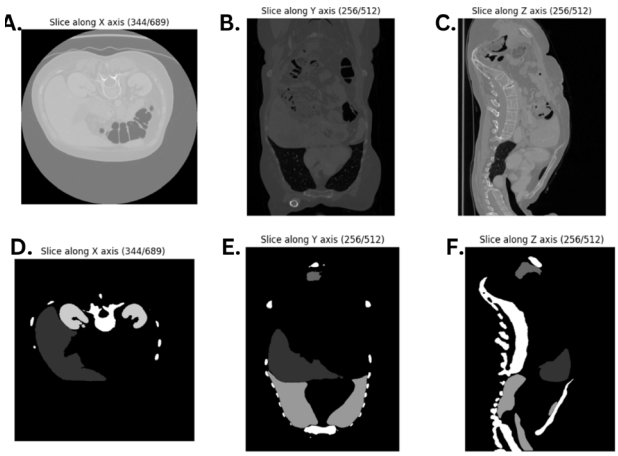

Figure 3. Sample slices from all three axes with labels. A) Feature of a sample slice, and **D)** shows the segmented mask from the top view. **B and E)** Corresponding images from the front view. **C and F)** Corresponding images from the side view.

https://doi.org/10.59720/23-323

Figure 4. Sequential slices in a 3D labeled image. The slices go from the uppermost (left) to the bottom (right) of a 3D labeled image. The middle slices contain more organs, as they reflect the center of the human body. For example, the slice numbers from left to right are 1, 52, 104, 253, 369, 427, 511, respectively. One volumetric image consisted of 512 such slices.

Google Drive was used to store the data and to help perform functions on the dataset by allowing Python libraries access to it. The SimpleITK library extracted each slice and different characteristics (e.g., shape, views) from various volumes. Matplotlib visualized the predicted masks and input/output after being converted to a NumPy array. The Shutil library relocated thousands of images to their respective training/testing drive folders. Finally, the TQDM library, although not necessary, was used to generate a progress bar when running a code block. This worked well to display the percentage left in the completion of a code's runtime, especially when it came to training the huge models.

Received: November 25, 2023
Accepted: July 1, 2024
Published: Novemeber 12, 2024

REFERENCES

1. Dobson, G. P. "Trauma of major surgery: A global problem that is not going away." *International Journal of Surgery*, vol. 81, Sep. 2020, pp. 47–54. https://doi.org/10.1016/j.ijsu.2020.07.017.
2. Shepard, M. "A look at the actual device costs for hospitals." Medical Product Outsourcing. bit.ly/41nRZJv. Accessed December 13, 2023.
3. Deo, N., & Anjakar, A. "Artificial Intelligence With Robotics in Healthcare: A Narrative Review of Its Viability in India." *Cureus,* May 23, 2023. https://doi.org/10.7759/cureus.39416.
4. Mishra, M. "Convolutional Neural Networks, Explained." Medium, www.towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939. Accessed December 13, 2023.
5. Yamashita, R., et al. "Convolutional Neural Networks: An overview and application in Radiology." *Insights into Imaging,* vol. 9, no. 4, Aug. 2018, pp. 611–629. https://doi.org/10.1007/s13244-018-0639-9.
6. Vaswani, A., et al. "Attention Is All You Need." *arXiv*, 12 June 2017. https://doi.org/10.48550/arXiv.1706.03762.
7. Ronneberger, O., et al. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *arXiv,* 2015. https://doi.org/10.48550/arXiv.1505.04597.
8. Van Lohuizen, Q., et al. "Assessing deep learning reconstruction for faster prostate MRI: Visual vs. Diagnostic Performance Metrics." *European Radiology*, 2024. https://doi.org/10.1007/s00330-024-10771-y.
9. Alabdulhafith, M., et al. "Automated wound care by employing a reliable U-Net architecture combined with ResNet feature encoders for monitoring chronic wounds." *Frontiers in Medicine,* vol. 11, 2024. https://doi.org/10.3389/fmed.2024.1310137.
10. Xie, E., et al. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers." *arXiv,* 31 May 2021. https://doi.org/10.48550/arXiv.2105.15203.
11. Simonyan, K., et al. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv,* 2015. https://doi.org/10.48550/arXiv.1409.1556.
12. He, K., et al. "Deep residual learning for image recognition." *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016. https://doi.org/10.1109/cvpr.2016.90.
13. Chen, J., et al. "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation." *arXiv,* 2021. https://doi.org/10.48550/arXiv.2102.04306.
14. Zettler, N., & Mastmeyer, A. "Comparison of 2D vs. 3D U-Net Organ Segmentation in abdominal 3D CT images." *arXiv*, 8 July 2021. https://doi.org/10.48550/arXiv.2107.04062.
15. Dai, Y., & Gao, Y. "TransMed: Transformers Advance Multi-Modal Medical Image Classification." *arXiv,* 10 March 2021. https://doi.org/10.48550/arXiv.2103.05940.
16. Huilgol, P. "Precision and Recall." Analytics Vidhya. short-url.at/jqv49. Accessed December 13, 2023.
17. Nolan, T. "CT volumes with multiple organ segmentations." Cancer Imaging Archive Wiki. Accessed December 13, 2023.
18. Terra, J. "Keras vs Tensorflow vs Pytorch: Key Differences Among Deep Learning." Simplilearn. www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article. Accessed December 13, 2023.

https://doi.org/10.59720/23-323

APPENDIX

The dataset, CT-Org2D, and code used in the study can be found at: github.com/MrCarry123/OrganSegmentation

Model	Batch Size	Learning Rate	Epochs	Optimizer
Unets	64	0.0001	20	Adam
Segformer	32	0.001	20	SGD

Table S1. Comparison of training parameters across different models. The unets were trained differently than SegFormer, but the epochs remained consistent throughout each model.