2023.12.10.

Model Ensemble for Medical Image Segmentation

Deep learning project work

Team:
Zsombor Bánfi
Botond Klenk
Bence Benyák

Project Description

In this project, we dived into the idea of using multiple models together, known as model ensembles, to make our deep-learning solutions more accurate. They are a reliable approach to improve the accuracy of a deep learning solution for the added cost of running multiple networks. Using ensembles is a trick that's widely used by the winners of AI competitions. We explored approaches to model ensemble construction for semantic segmentation. Trained multiple models and constructed an ensemble from them. Then analysed the improvements, benefits, and added costs of using an ensemble.

Methods of training

For the project, we tested multiple models, with different architectures and complexity

SimpleSegmentationModel

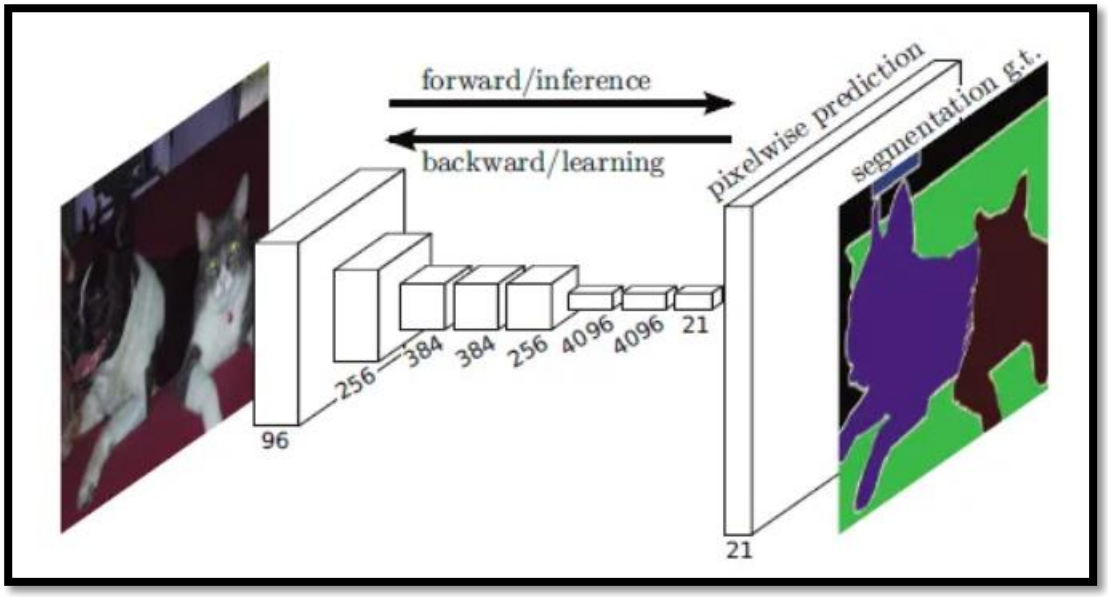
This model, created using the GitHub Copilot LLM, was initially designed as a baseline for segmentation tasks. It comprises two main submodels: the Upsampling and Downsampling components. However, during the training process, it became evident that the model's complexity was insufficient for the project's requirements.

An issue arose where the model tended to classify entire images as background, particularly because a significant portion of the pictures predominantly featured background elements. This limitation hindered the model's ability to accurately segment foreground objects.

Fully Convolutional Network (FCN) - Resnet

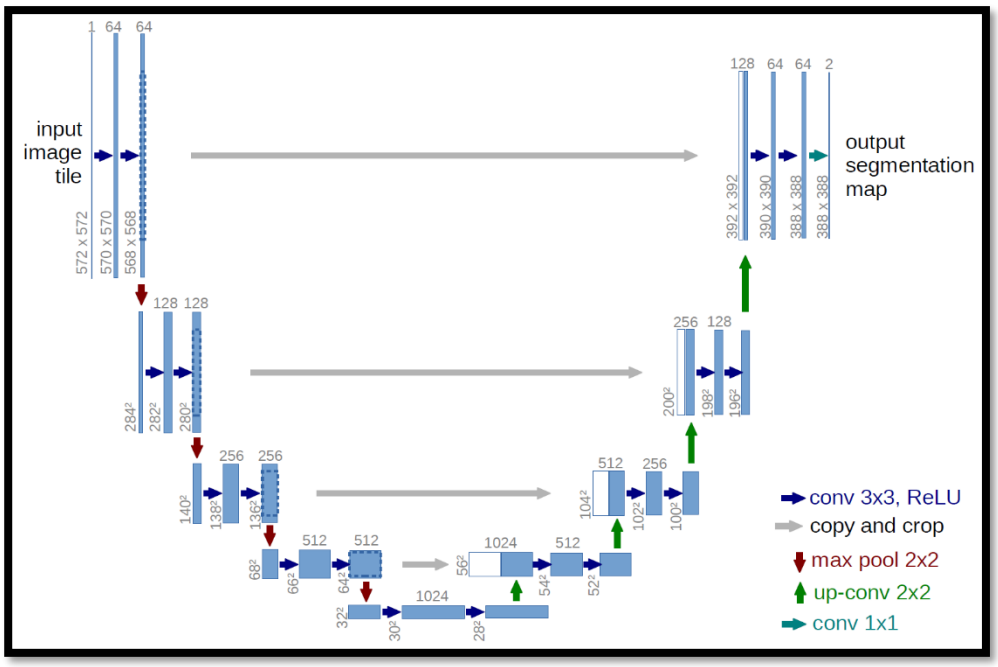
We imported an FCN model segmentation model with a resnet50 backbone from pytorch. This model needed some modification as it was designed to handle RGB channel pictures instead of the grey scale images found in our dataset. In FCNs, instead of gradually reducing spatial dimensions through pooling layers, the network employs transposed convolutions or upsampling layers to maintain the spatial resolution of the input. FCNs also feature skip

connections or skip architecture, allowing the network to merge information from different levels of abstraction.



Unet

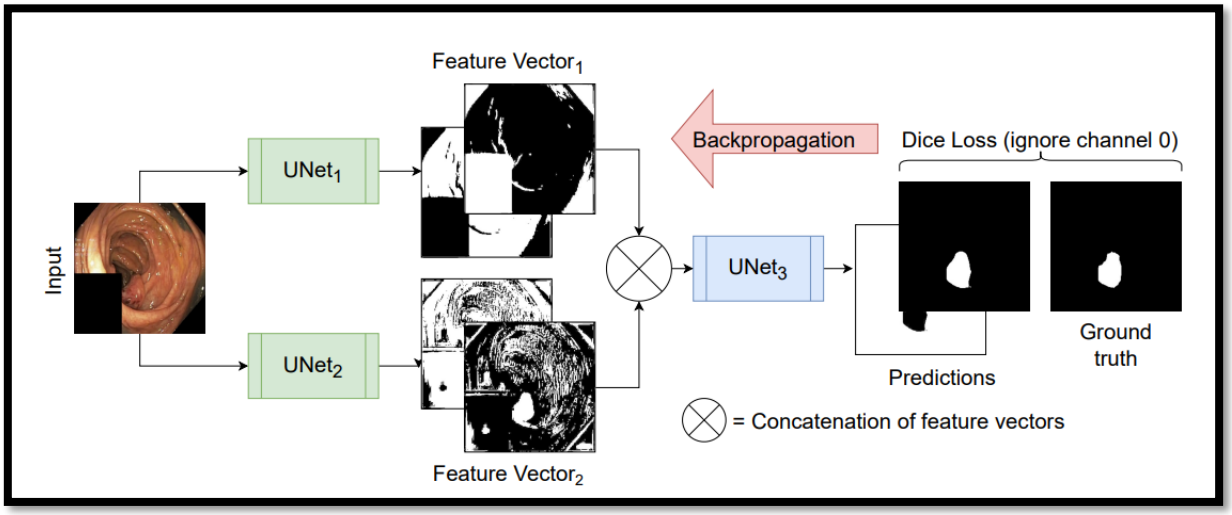
In contrast to the architectural paradigm of Fully Convolutional Networks (FCNs), U-Net deploys a distinctive U-shaped structure, comprising contracting, bottleneck, and expansive pathways with direct skip connections. This architectural uniqueness positions U-Net as a specialized solution for image segmentation tasks, particularly excelling in contexts demanding precision, such as biomedical image segmentation. The incorporation of direct skip connections facilitates the fusion of high-level semantic information with nuanced details, underlining U-Net's efficacy in preserving spatial resolution and intricate features. This architectural emphasis renders U-Net notably advantageous in scenarios where the preservation of fine-grained information holds significant importance.



TriUnet

The TriUNet is an ensemble architecture, which means that it combines the outputs of multiple different models to produce a final segmentation mask.
The architecture is composed of three main components: two separate UNet models and a third fusion component. The input image is passed through the two UNet models in parallel, each with different randomized weights, which helps to reduce overfitting and improve the generalizability of the model. The output of each UNet model is then concatenated together and passed through the fusion component. The fusion component is a third UNet model that takes the concatenated feature vectors from the two input UNets and predicts the final segmentation mask.

To further explore the potential of the triangular architecture, we investigated various model combinations. One iteration involved employing three fully convolutional networks (FCNs), while another configuration combined two FCNs for feature extraction and a U-Net for prediction. Additionally, we extended the TriUnet architecture to MultiUnet, enabling the utilization of more than two models for feature extraction. However, executing multiple models concurrently proved challenging within the PyTorch framework. To address this issue, we experimented with a single U-Net model for feature extraction, but with double the original channel count, and employed depthwise convolutions (with two groups). With these modifications and the input replicated across the channel dimensions, the model could mimic the behavior of two parallel U-Net models. Because this approach utilized the original U-Net implementation, it exhibited significantly slower performance compared to the segmentation_models_pytorch library, prompting the discontinuation of our experiments.



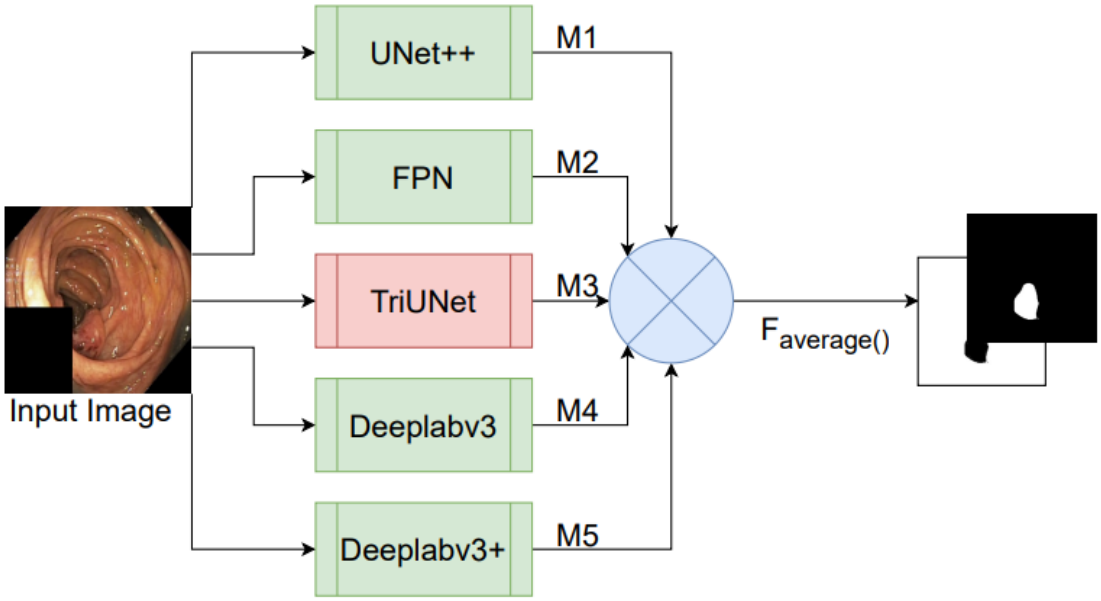
DivergentNets

DivergentNets is a novel ensemble architecture for medical image segmentation, inspired by the success of combining multiple high-performing segmentation models. It builds upon the TriUNet architecture, incorporating additional segmentation models into its design to enhance accuracy, robustness, and generalizability.

The Divergent Nets architecture consists of multiple parallel paths, each containing a combination of the TriUNet and other segmentation models, such as UNet++, FPN, DeepLabv3, and DeepLabv3+. This diverse set of models allows the network to explore various representations and learn from the strengths of each approach.

As the input image is processed through each path, the outputs are passed through a fusion module. This module combines the information from the different paths, employing techniques like weighted averaging or consensus voting to generate a final segmentation mask. The final mask reflects the collective decision of the ensemble, leveraging the strengths of multiple models to achieve superior performance.

Our experiments did not involve the models shown in the provided figure, primarily due to computational power constraints, instead, we employed a single U-Net, an FCN, and a TriUnet model.



Training

We implemented our training pipeline using Pytorch and Pytorch Lightning framework.
We also used other widely used python libraries such as segmentation_models_pytorch and torchmetrics.

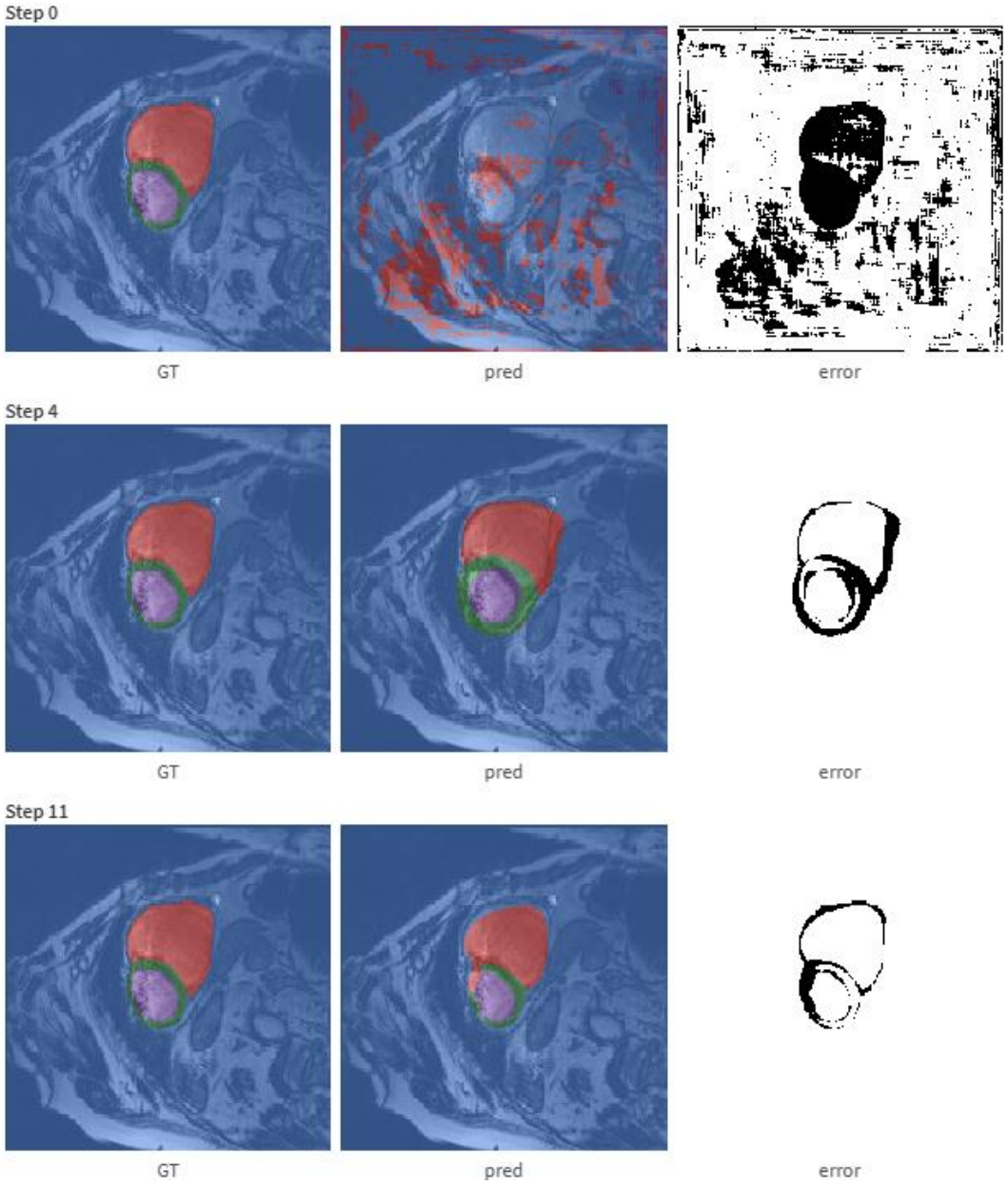
Evaluation

As the used ACDC dataset was originally a Challange (hence the name) where one of the main metrics was the Dice coefficient, therefore we used this metric as our loss function. It measures the similarity between the predicted and target segmentation masks. Dice Loss provides a differentiable and smooth measure of segmentation accuracy that is also highly interpretable, robust to noise and computationally efficient.

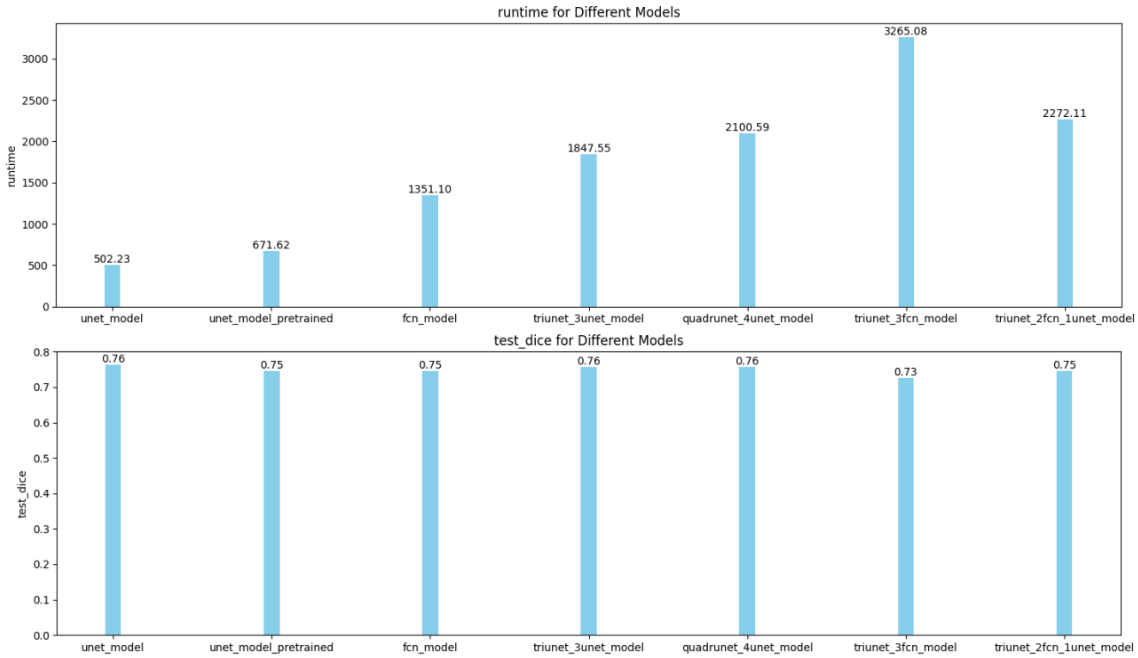
$$\text{Dice} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}}$$

To more fully assess the segmentation performance of both ensemble and non-ensemble methods, we measured their performance using multiple metrics, including Dice Loss and pixel-wise accuracy. Additionally, we recorded training time to compare the computational cost of each approach.

For logging we used WandB. Besides the evaluation metrics we also uploaded the actual prediction pictures for each epoch:



We downloaded the runs from WandB so we can also visualize some data, for example the runtime and the test's dice metric:



Conclusion

Our primary objective centered on establishing an efficient pipeline, within which we investigated the capabilities and practicality of ensemble models. We successfully constructed a fundamental framework to train and test these models. Evaluation of model performances was conducted to select the optimal among them.

References

Unet: https://arxiv.org/abs/1505.04597
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* (pp. 234-241). Springer International Publishing.

FCN: https://arxiv.org/abs/1411.4038
Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

DivergentNets, TriUnet: https://arxiv.org/abs/2107.00283
Thambawita, V., Hicks, S.A., Halvorsen, P. and Riegler, M.A., 2021. Divergentnets: Medical image segmentation by network ensemble. *arXiv preprint arXiv:2107.00283*.

ACDC Dataset: https://ieeexplore.ieee.org/document/8360453
Bernard, O., Lalande, A., Zotti, C., Cervenansky, F., Yang, X., Heng, P.A., Cetin, I., Lekadir, K., Camara, O., Ballester, M.A.G. and Sanroma, G., 2018. Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: is the problem solved?. *IEEE transactions on medical imaging*, *37*(11), pp.2514-2525.

LLM usage

Throughout our project, we leveraged the most widely adopted LLMs (ChatGPT, Bard, and Copilot) to streamline our workflow.

Copilot was our primary tool because its integration with VS Code significantly boosted our productivity. We employed it to generate source code, explain code snippets from external repositories, and also answer theoretical questions about deep learning. By the end of the project, we had employed Copilot to generate the majority of our code documentation. While Copilot often produced accurate and functional code, occasionally it misinterpreted our prompts or delivered confidently incorrect responses.

ChatGPT and Bard primarily served as generators and polishers for our documentation and ReadMe files.