Biomedical Computer Vision

Teka Kimbi Ntimanputu,
teka.kimbi@mail.polimi.it

I. Introduction

Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) are crucial in modern medical diagnostics, providing high-resolution images essential for diagnosing and treating various diseases. Accurate segmentation of MRI and CT images is vital for identifying and analyzing anatomical structures and pathological regions. Traditional segmentation methods, such as thresholding, edge detection, and region growing, often struggle with variability in image quality and anatomical diversity. Modern techniques leverage machine learning and deep learning algorithms to enhance segmentation performance. Convolutional Neural Networks (CNNs), U-Nets, and other deep architectures automatically learn features from large datasets of annotated MRI and CT images, leading to more precise and robust segmentation results. This work focuses on segmenting abdominal organs (liver, kidneys, and spleen) from MRI data using advanced deep learning techniques.

II. Dataset description

The dataset used in this work is provided by the CHAOS Challenge and comprises Abdominal MRI data. Specifically, 40 DICOM datasets from the T1-DUAL sequence (20 in-phase and 20 out-phase) are utilized. These images were retrospectively collected from the PACS of DEU Hospital using a 1.5T Philips MRI, resulting in 12-bit DICOM images. While the majority of the images have a resolution of 256 x 256 pixels, there are also some with a resolution of 288 x 288 pixels.

The datasets do not include tumors or lesions near the annotated organs of interest (liver, kidneys, spleen). Each dataset varies in inter-slice distance (5.5-9 mm) and x-y spacing (1.36-1.89 mm), with an average of 36 slices per dataset. Ground truth labels for the segmentation tasks are provided, detailing the annotated regions of the liver, kidneys, and spleen. Only the T1-DUAL in-phase sequence is employed for this work.

Several considerations are necessary when working with this dataset, including the average number of slices per dataset, inter-slice distance, and image resolution. Variability in contrast and pixel intensity can also be an issue. Consequently, a preprocessing pipeline is required

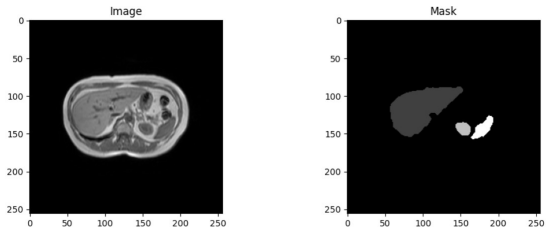

Fig. 1. Medical scan (left) and corresponding segmentation mask (right) highlighting regions of interest

to standardize the images, ensuring they are in the correct format for training the neural network.

The preprocessing pipeline is composed as follows:

- **Resampling:** Adjusting the voxel size to ensure uniform spatial resolution across all the datasets.
- **Resizing:** Scaling the images to a consistent resolution, in this case, 192 x 192 pixels.
- **Standardization:** Ensuring the image intensity values have a standardized range and distribution.
- **Normalization:** Scaling pixel intensity values to a common range, in this case, [0, 1].
- **Equalization:** Enhancing image contrast to improve the visibility of anatomical structures.

The pipeline is applied only to the DICOM images. For the ground truth labels, only the resizing process is applied, ensuring consistency with the processed DICOM images.

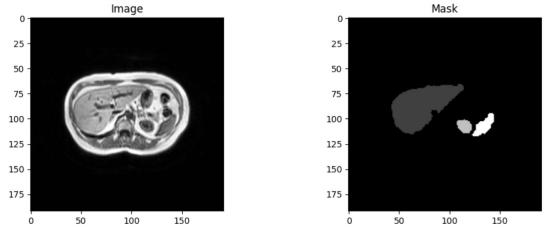

Fig. 2. Processed medical scan (left) and corresponding segmentation mask (right) highlighting regions of interest

After preprocessing, the dataset is divided into training, validation, and test sets with a ratio of 75% - 15% -10% (2880, 576, and 384 slices, respectively). Care is

taken to ensure that slices from the same dataset do not appear in more than one of the three sets.

III. Experiments

This section describes the steps, trials and implementations that led to the final implementation that is shown in Section IV.

A. Model Architectures

In modern deep learning, various architectures are available for image segmentation tasks. Among these, U-Net architectures stand out for their effectiveness in biomedical image segmentation. With skip connections, U-Nets preserve spatial resolution while capturing complex features, making them valuable for delineating anatomical structures and pathological regions.

Based on concepts introduced in lectures, the U-Net architecture was employed as a starting point for the analysis. The components of the U-Net model, featuring an encoder-decoder structure. The encoder consists of four blocks. Each block includes two 3x3 convolutions (ReLU, same padding) followed by 2x3 max pooling. The number of filters in the convolutions doubles with each block, starting from the 64 filters in the first block, 128 in the second, 256 in the third, and 512 in the fourth. Additionally, the fourth block is followed by a dropout layer with a rate of 0.5 to prevent overfitting. The bottleneck includes two 1024-filter 3x3 convolutions (ReLu, same padding), followed by a dropout layer with a rate of 0.5. The decoder also consists of four blocks. Each block includes 2x2 upsampling, a 2x2 convolution (ReLU) withe the same number of filters as the corresponding encoder block, concatenation with the output of the corresponding encoder block, and two 3x3 convolutions (ReLU, same padding). The number of filters decreases with each block, starting from 512 filters in the first block, 256 in the second, 128 in the third, and 64 in the fourth. The output layer consists of a 1x1 convolution with a number of filters equal to the number of classes being employed and softmax activation. The architecture is illustrated in Figure 3.

B. Performance Evaluation and Loss Function

In image segmentation evaluation, Mean Intersection over Union (IoU) and the Dice coefficient are pivotal metrics. Mean IoU assesses the overlap between predicted and ground truth masks, providing insights into the model's boundary delineation accuracy. Conversely, the Dice coefficient quantifies spatial overlap between these masks, ranging from 0 (no overlap) to 1 (perfect match). While IoU emphasizes the overlap relative to the total area, the Dice coefficient places more emphasis on the agreement between the masks. Both metrics range from 0 to 1, with higher values indicating greater spatial similarity. To handle imbalanced datasets during training, combining Dice loss with categorical sparse cross-entropy has become popular. Dice loss focuses on spatial similarity, while categorical sparse cross-entropy improves pixel-wise classification accuracy. Using these weighted loss functions together balances the optimization, boosting both segmentation precision and classification accuracy for better overall performance.

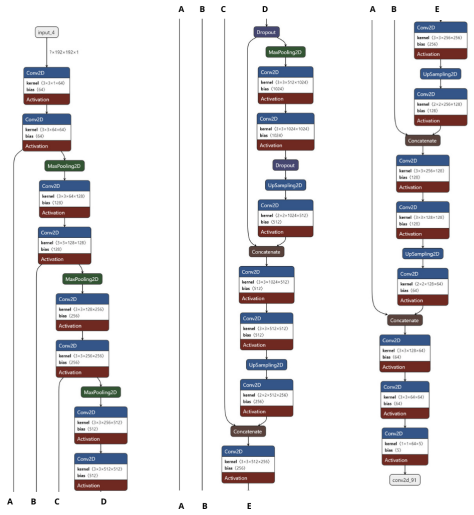

Fig. 3. Starting U-Net Model Architecture

C. Enhanced Architecture: Performance Boost and Overfitting Defense

The earlier versions of the trained model faced difficulty in achieving a mean Intersection over Union (IoU) value beyond 0.65. Additionally, a troubling trend emerged where the validation loss and training loss started to diverge, hinting at potential overfitting. These obstacles highlighted the necessity for the introduced improvements in both the model architecture and training process. To enhance performance and address the overfitting issues identified in initial trials with the architecture, batch normalization and dropout layers were incorporated. Batch normalization was added after each convolutional layer to stabilize training and improve generalization. Dropout layers, with rates up to 0.5, were introduced in deeper layers to prevent the model from becoming overly reliant on specific neurons. Various batch sizes (16 to 64) and learning rates (0.01 to 0.00001)

were tested, with the final configuration settling on a batch size of 16 and a learning rate of 0.0001. These changes significantly enhanced the model's robustness and reduced overfitting.

IV. Final Model

The final model, detailed in Section III, follows a U-Net-like setup with an encoder-decoder structure. In the encoder, there are four blocks, each with two 3x3 convolutional layers followed by ReLU activation and batch normalization. The number of filters increases from 64 to 512 across these blocks, and a dropout layer with a 50% rate is added after the fourth block to prevent overfitting. Moving to the bottleneck, it consists of two 3x3 convolutional layers with 512 filters each, followed by ReLU activation, batch normalization, and a 50% dropout rate. In the decoder, there are four blocks that mirror the encoder but in reverse. Each block has 2x2 upsampling, a 2x2 convolutional layer with ReLU activation, and a fusion step where the corresponding encoder block's output is combined. Additionally, two more 3x3 convolutional layers refine the feature representation. The number of filters decreases from 512 to 64 across these blocks. The output layer retains the same configuration as the initial model, remaining unchanged throughout the architectural refinement process. The final architecture is depicted in Figure 4, and the training procedure is detailed in Figure 5.

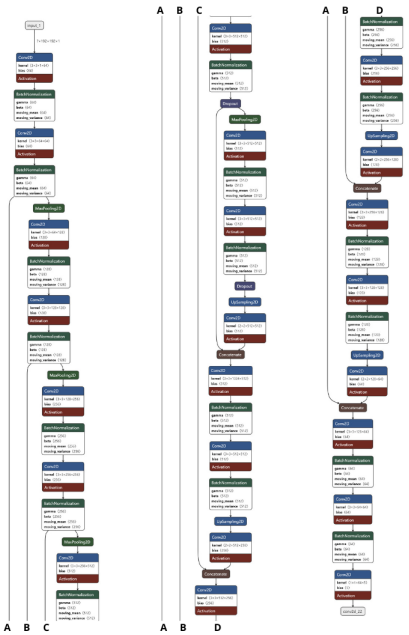

Fig. 4. Final U-Net Model Architecture.

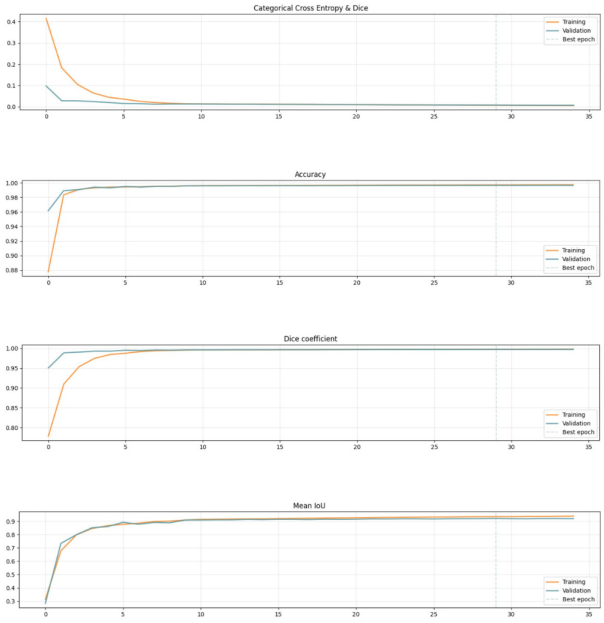

Fig. 5. Training and validation loss, along with performance metrics, monotored throughout the training process of the final model.

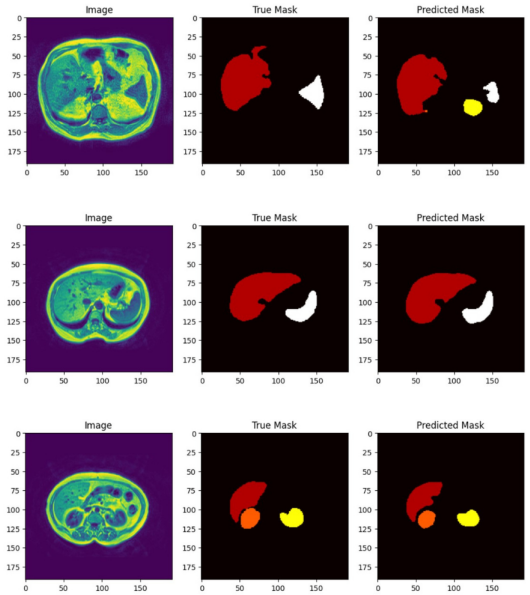

Fig. 6. MRI scan (left), True segmentation mask (center) and predicted segmentation mask (right).

This model has shown superior performance compared to the other attempts, exhibiting a Test Loss of 0.0410, Test Accuracy: 0.9864, Test Dice Coefficient: 0.9886, and Test Mean Intersection over union: 0.7557. The predictions are shown in Figure 6.

V. Conclusions

In this work, a model has been proposed for image segmentation on biomedical images. After evaluationg different appoaches and models, the configuration detailed in Section IV emerged as the most effective, leading to satisfactory. Specifically, the model achieved a Test Loss of 0.0410, Test Accuracy of 0.9864, Test Dice Coefficient of 0.9886, and Test Mean Intersection over Union of 0.7557. In conclusion, several routes can be explored to enhance performance in future rounds. Firstly, augmenting the training data with more samples can diversify the dataset and improve generalization. Secondly, fine-tuning hyperparameters, such as adjusting learning rates or exploring different optimization algorithms, may lead to improved model performance. Lastly, advanced techniques like ensemble learning or utilizing larger pre-trained models could further enhance segmentation accuracy. These strategies offer promising directions for refining the model and achieving better results in subsequent iterations.