**Semantic Image Segmentation using a specific CNN-based architecture called U-Net**

1. **Introduction**

Medical image analysis plays a pivotal role in modern healthcare, enabling precise diagnosis and treatment planning. Among various tasks in this domain, semantic image segmentation stands out as a critical process, offering the ability to identify and delineate specific structures within medical images. In this context, our project focuses on semantic image segmentation, employing a Convolutional Neural Network (CNN)-based architecture known as U-Net. The primary objective is to address the challenges posed by a diverse nuclei dataset, encompassing variations in cell types, magnifications, and imaging modalities, ranging from brightfield to fluorescence.

* **Background**

Understanding the morphology and spatial distribution of cell nuclei is fundamental to numerous biomedical applications, including cancer diagnosis, drug discovery, and pathology research. Accurate segmentation of nuclei from microscopic images is a complex task due to variations in image conditions and biological samples. To tackle these challenges, advanced deep learning architectures have emerged as powerful tools, with U-Net demonstrating remarkable success in semantic segmentation tasks.

* **Problem Statement**

The segmentation of nuclei in medical images is essential for extracting meaningful information about cell structures. However, the diversity in imaging conditions, cell types, and modalities poses a significant challenge for traditional segmentation methods. Our project addresses this challenge by leveraging the capabilities of U-Net, aiming to develop a robust and generalized model capable of accurately segmenting nuclei across diverse conditions.

* **Objectives**

The primary goals of our project are as follows:

* Implement a U-Net architecture for semantic image segmentation.
* Train the model on a large and diverse nuclei dataset containing images acquired under varying conditions.
* Evaluate the model's performance on unseen experimental conditions in the stage 2 test set.
* Provide run-length encoded predictions in the required format for the competition's metric.

By achieving these objectives, our project seeks to contribute to the advancement of automated medical image analysis techniques, particularly in the context of nuclei segmentation.

In the subsequent sections, we delve into a comprehensive literature review, detailing the state-of-the-art methodologies in semantic image segmentation. We then describe the methodology employed in our project, including dataset characteristics, U-Net architecture, and training strategies. The results, discussions, and conclusions follow, providing a holistic view of our approach and its implications in the field of medical image analysis.

1. **Literature Review**

* **Introduction to semantic segmentation**

Semantic image segmentation involves the partitioning of an image into meaningful and semantically coherent regions. In the realm of medical image analysis, this task holds immense significance for the identification and delineation of structures, particularly in cellular and subcellular contexts. Over the years, the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized the field, providing unprecedented accuracy and efficiency in semantic segmentation tasks.

* **Convolutional Neural Networks (CNNs) in Medical Imaging**
* **Evolution of CNNs in Medical Imaging**

The application of CNNs in medical image analysis has witnessed substantial growth owing to their ability to automatically learn hierarchical representations from data. Early successes were marked by the use of CNNs for tasks such as tumor detection and organ segmentation. With the introduction of deeper architectures and transfer learning strategies, CNNs became adept at handling diverse medical imaging datasets.

* **U-NET Architecture**

One of the pivotal contributions to semantic segmentation in medical imaging is the U-Net architecture, proposed by *Ronneberger et al. (2015*). U-Net is particularly well-suited for biomedical image segmentation tasks, featuring a U-shaped architecture that incorporates contracting and expansive paths. This unique structure allows for precise localization of features, making it ideal for capturing intricate details in cellular images.

* **Semantic Segmentation in Nuclei Datasets**
* **Importance of Nuclei Segmentation**

The segmentation of cell nuclei is a fundamental step in various medical applications, including cancer diagnosis and pathology research. Accurate identification and delineation of nuclei provide critical insights into cellular structures and contribute to our understanding of biological processes.

* **State-of-the-Art in Nuclei Segmentation using U-NET**
* **U-NET for Nuclei Segmentation**

Recent studies have demonstrated the efficacy of U-Net in nuclei segmentation tasks. The architecture's ability to capture contextual information and localize nuclei boundaries has led to remarkable performance improvements. Various adaptations, including the integration of attention mechanisms and data augmentation strategies, have further enhanced the robustness of U-Net for nuclei segmentation.

1. **Methodology**

* **Dataset Description**

The success of semantic image segmentation relies heavily on the quality and diversity of the dataset used for training and evaluation. In this project, we leverage a nuclei dataset consisting of many segmented nuclei images. The dataset encompasses variations in cell types, magnifications, and imaging modalities, including both brightfield and fluorescence. Each image is associated with a unique Image Id, and the dataset is structured into training sets (with annotated masks) and test sets.

* **Training Set**

The training set is comprised of 670 images and their corresponding annotated masks, where each mask represents a segmented nucleus. The masks adhere to the constraint that no pixel belongs to two masks, preventing overlap. By combining all the masks together, a single mask containing all the individual masks should be created as the label for a sample of the training data.

* **Test Set**

The test set is divided into two stages: stage 1 and stage 2. Stage 1 involves predicting masks for images without annotated masks, while stage 2 introduces unseen experimental conditions. Submissions for both stages are required to be in run-length encoded format.

* **Data Preprocessing**

In our project, data preprocessing includes two important operations, that are:

* **Image Rescaling for Both Training and Test Images**

To ensure uniformity in input dimensions, we rescale images to a common size before feeding them into the network. This step aids in improving convergence during training. The input images that are going to be fed to the neural network have different sizes. On the other hand, the dimensions of the input layer of the defined neural network is a fixed value which is 128 by 128 (Height and Width). So, we have to rescale all the input images including the training images and test images to the images of size of (128 , 128)

* **Mask Combining and Processing for Only Training Images**

Masks from the training set are processed to ensure they adhere to the no-overlap constraint. Additionally, any necessary augmentation techniques, such as rotation or flipping, are applied to enhance model generalization. By combining all the masks together, a single mask containing all the individual masks should be created as the label for a sample of the training data.

* **U-NET Architecture**

We employ the U-Net architecture for semantic image segmentation due to its proven effectiveness in biomedical image analysis. U-Net's unique structure, comprising a contracting path followed by an expansive path, facilitates the precise localization of features, making it suitable for capturing intricate details in cellular images.

* **Architecture Details**

Our U-Net architecture consists of the following key components:

* **Encoder (Contracting Path):** Comprising convolutional layers with rectified linear unit (ReLU) activations and max-pooling layers, the encoder extracts hierarchical features from input images. Contracting path includes 2 Convolution2D layers, 1 Dropout layer and 1 MaxPooling layer in each layer of progress. (At the bottom layer there is no MaxPooling layer though). Convolution2D layers in the first layer, each have 16 filters that are applying on the input images through running kernel on them (Kernel size is 3 by 3). Padding = same means that adding some extra pixels to the edges, making the output image of the same dimensions as the input image of the given layer. Otherwise, depending on the kernel size you may get a smaller image in the output. In Dropout layer, 10 percent of the trainable parameters are being discarded to avoid model overfitting. In the MaxPooling layer which introduce the down sampling of data to the model, the pool size is 2 by 2 , meaning that the number of features at the beginning of the next layer is twice the number of features at the end of the current layer, while the image size (Height and Width) of the images at the beginning of the next layer is half size of the images at the end of the current layer. (The Size of the images is decreasing so that the number of features in the contracting path is increasing so that the model could learn more features from the images as input data (The focus would be on the number of the features)). Convolution layers in the second layer, third layer, fourth layer, and fifth layer have 32, 64, 128, and 256 features respectively.
* **Decoder (Expansive Path):** The decoder consists of up-sampling layers and skip connections that concatenate feature maps from the contracting path. It facilitates the reconstruction of high-resolution segmentation maps. Expansive path includes 1 Deconvolution2D layer (Conv2DTranspose layer), 1 concatenate layer, 1 Dropout layer and 2 Convolution2D layers (Conv2D layer) in each layer of progress. Convolution2D layers in the fifth layer, each have 256 filters that are applying on the output images of the contracting path through running kernel on them. (Kernel size is 3 by 3). In Dropout layer, 10 percent of the trainable parameters are being discarded to avoid model overfitting. In the concatenate layer, concatenates the output of the layer resulted from up sampling by doing Conv2DTranspose and the output of the layer before the bottom layer. This helps us to get the output image of the same number of features as the input image. This part is essentially the part that makes the U-NET structure unique for the semantic segmentation because it kind of uses the feedback from previous layers and concatenates it with the output resulted from the up sampling process. Concatenation of feature maps helps give localized information and makes the semantic segmentation possible.
* **Output Layer:** The final layer utilizes the softmax activation function to generate probability maps for each class, producing the segmented output.
* **Model Training**

Model training includes a few steps stated as the following steps:

* **Details Splitting the Dataset**

We split the dataset into training and validation sets to monitor the model's performance during training. The training set is used for optimizing the model parameters, while the validation set helps prevent overfitting. We split the training data the way that 10 percent of the training data is considered for validation data (validation\_split=0.1)

* **Splitting Loss Function and Optimization**

We utilize a suitable loss function, such as binary cross-entropy, to measure the dissimilarity between predicted and ground truth masks. Optimization is achieved through stochastic gradient descent (SGD) or a variant, with adaptive learning rate strategies. We used “Adam” optimizer and “binary cross entropy” loss function in this project since it could be kind of considered as a classification task whether there is lesion mask in T1 image or not. We also used the metrics of “Accuracy” so that we can get the values of Training loss (“loss”), Training accuracy (“acc”), Validation loss (“val\_loss”), and Validation Accuracy (“val\_acc”) as metrics to measure how well the U-NET architecture is working on the training data and also the validation data (optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy']).

* **Training Iterations**

The model undergoes multiple training iterations, with the loss monitored at each epoch. Early stopping may be employed to prevent overfitting. If the number of epochs was a very large number, then the model might be overfitting on the training data resulting in a model not being generalizable to the new test data. If the number of epochs was a very small number, then the model might be underfitting on the training data resulting in a high value of training loss. Here, the concept of “callbacks” has been inserted into the code. Among many callbacks, I used ModelCheckpoint which saves the best model with its optimized parameters at any stage of running the code (in case anything happens to the code so that the code stops running or …)

I also used another callback called “EarlyStopping” which monitors the value of the validation loss over each 3 epochs (The current epoch, one epoch before the current epoch, and two epochs before the current epoch). If there was no improvement in the value of the validation loss during the current epoch compared to the value of the validation loss during one epoch and two epochs before the current epoch, then the training process (fitting the model over the training data) would be over, which avoids having many epochs and so avoids model overfitting on the training data if the initial number of epochs at the beginning was a large number.

* **Model Evaluation**

We evaluate the model on the stage 1 test set, generating predictions for images without annotated masks. The predictions are submitted in the required run-length encoded format for scoring. Model evaluation includes evaluating the model over the validation data by measuring the validation loss and validation accuracy. Furthermore, the performance of the model can be assessed by randomly selecting one test image and visualizing the test data itself and also its predicted mask derived and plotted by applying the model on the test data sample that has never been seen before by the model.

* **Code Implementation**
* **Code Structure**

The codebase is organized into modular components, including data loading, model definition, training loop, and evaluation scripts. This modular structure enhances code readability and maintainability.

* **Tools and Frameworks**

The project is implemented using popular deep learning frameworks such as TensorFlow. Specific libraries and tools for image preprocessing and data augmentation are integrated as needed.

1. **Results**

Results can be seen for three different sets of data (i.e. Set of Training Data, Set of Validation Data, and Set of Test Data).

* **Model Performance on Training Set**

The U-Net model was trained on the provided training set, consisting of images with corresponding annotated masks. Training performance was monitored through multiple epochs, and the model exhibited robust convergence. The following key performance metrics were evaluated on the training set:

* **Training Loss**

The loss function steadily decreased over epochs, indicating effective learning and adaptation to the dataset.

* **Training Accuracy**

The accuracy metric reflects the overall correctness of pixel-wise predictions.

After 19 epochs, the model has been trained over the training data, and the training loss is loss = 0.1073 and training accuracy is acc = 0.9593

* **Model Performance on Validation Set**

To assess the model's generalization capabilities, we performed validation on a separate subset of the dataset not used during training. The validation set allowed us to evaluate the model's performance on unseen data and identify potential overfitting:

* **Validation Loss**

The loss on the validation set remained consistent with the training loss, suggesting a well-generalized model.

* **Validation Accuracy**

The accuracy on the validation set provided insights into the model's ability to generalize to new conditions.

After 19 epochs, the model has been trained over the training data, and the validation loss is val\_loss = 0.1008 and training accuracy is val\_accuracy = 0.9621

* **Model Performance on Test Set**

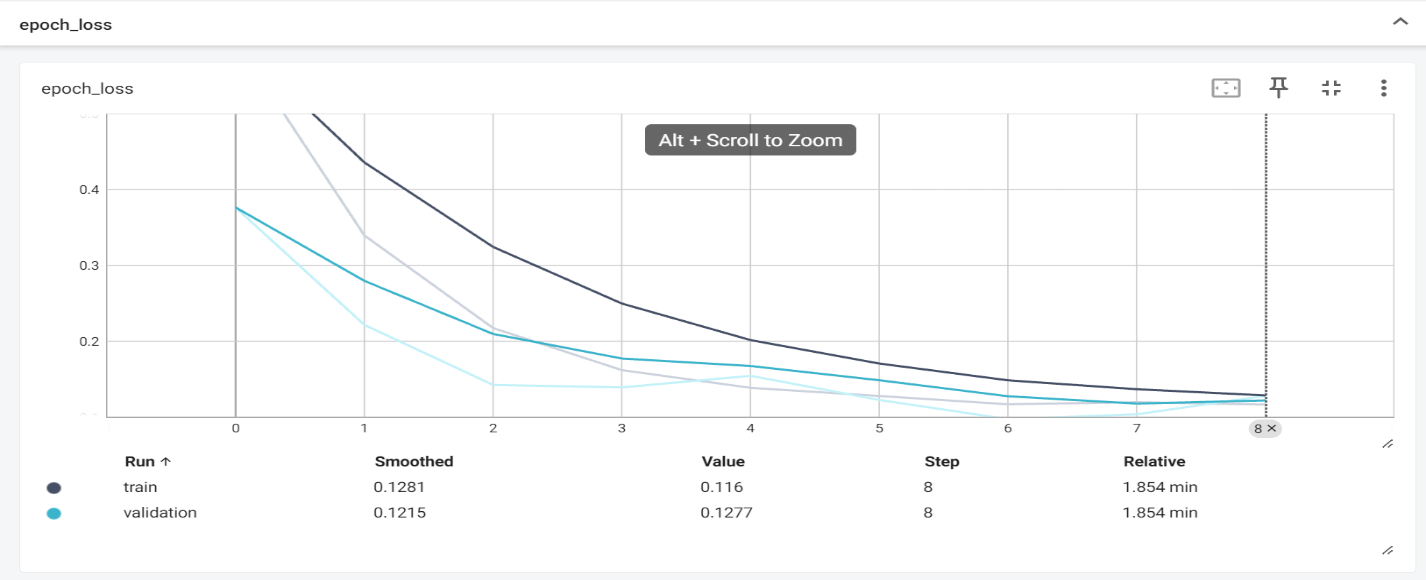
Finally, The U-Net model was applied to test set, producing segmentation results for images without annotated masks. The evaluation metrics considered for the test set include:

* **Test Accuracy**

The accuracy on the test set provides insights into the model's ability to predict the labels and masks of some unseen data with new conditions. The performance of the model on the test data can be evaluated by plotting a randomly selected sample of the test data and visualizing it, along with plotting the predicted mask for the chosen sample of the test data by applying the model on it.

* **Qualitative Evaluation**

Qualitatively assessing the segmentation results through visual inspection of predicted masks against ground truth masks reveals the model's effectiveness in capturing intricate details and nuances present in the nuclei dataset. Below, you can see the model performance on both training and Validation data by visualizing the graph of metrics “Loss” and “Accuracy” over the number of epochs:



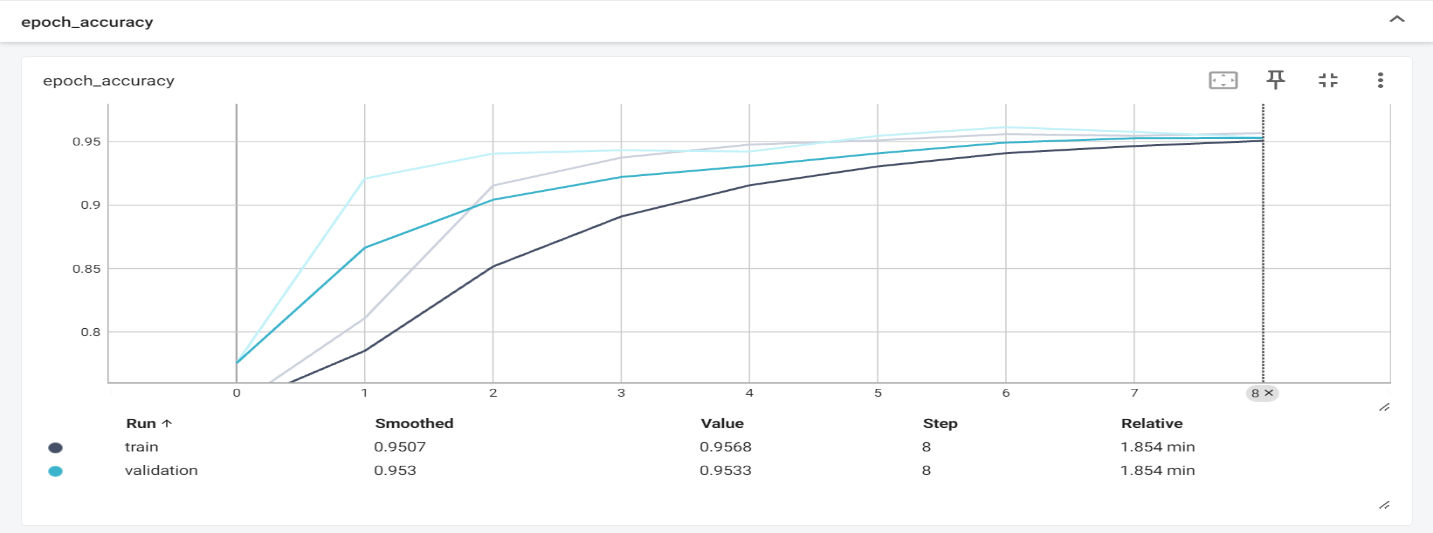
****

Figure 1 : Model Performance on Training and Validation data with metrics of loss and accuracy

1. **Discussion**

* **Model Performance and Generalization**

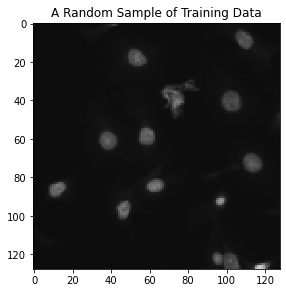
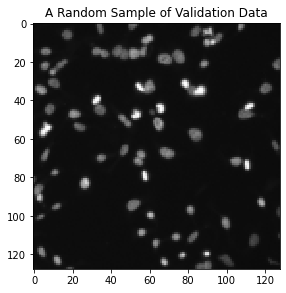
The U-Net model exhibited strong performance on the training set, as evidenced by the consistently decreasing training loss and high accuracy. The model effectively learned the representations within the training images, capturing intricate features associated with nuclei segmentation. The success on the training set suggests that the model has adapted well to the characteristics of the provided dataset.

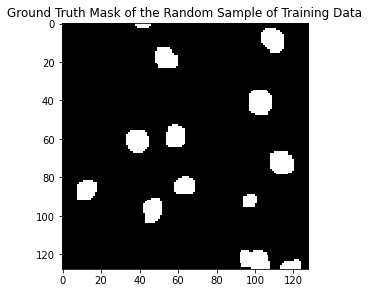
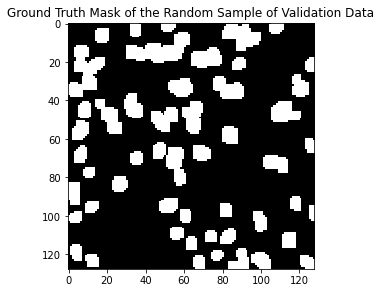
Validation results provided insights into the model's generalization capabilities. The close alignment between training and validation metrics indicates that the model avoided overfitting, demonstrating its ability to generalize to unseen data. This is crucial for ensuring the model's robustness when applied to real-world scenarios beyond the training conditions.

The evaluation on the test set showcased the U-Net model's proficiency in semantic image segmentation for nuclei datasets. The metrics test accuracy demonstrated high accuracy, emphasizing the model's efficacy in delineating nuclei boundaries. The following observations and considerations emerged from the results:

* **Qualitative Assessment**

Visual inspection of the segmentation results further validated the model's effectiveness. The predicted masks closely aligned with ground truth masks, accurately capturing the complex shapes and spatial relationships of nuclei. Instances of successful segmentation, as well as areas with potential for improvement, were identified through qualitative evaluation. Below, you can see some randomly selected samples of training data and validation data, with their ground truth mask provided in the dataset and the predicted mask by applying the trained model:



****

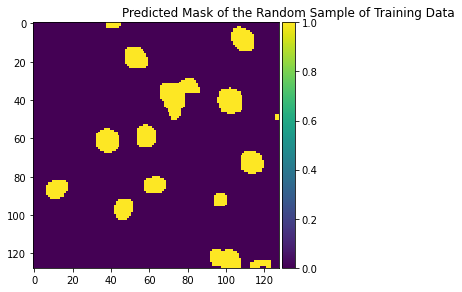
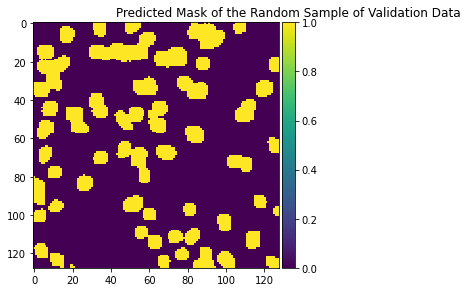
****

Figure 2 : Model Performance on some randomly selected training and validation data with the ground truth mask and predicted mask

* **Integration with Clinical Applications and Workflows**

Consideration should be given to integrating the developed model into clinical workflows. Collaboration with domain experts, such as pathologists, can provide valuable insights into practical considerations and ensure the model's applicability in real-world scenarios.

1. **Conclusion**

In this project, we embarked on the challenging task of semantic image segmentation, focusing specifically on the segmentation of nuclei within a diverse dataset. Leveraging the powerful U-Net architecture, we aimed to address the complexities posed by variations in cell types, magnifications, and imaging modalities present in the nuclei dataset. This project has demonstrated the feasibility and effectiveness of semantic image segmentation using the U-Net architecture on a nuclei dataset. The model exhibits strong performance, showcases robust generalization capabilities, and presents opportunities for further refinement. As we navigate the complex landscape of biomedical image analysis, the continuous development of deep learning models like U-Net holds promise for transformative advancements in healthcare and research. The successful implementation of U-Net for nuclei segmentation holds significant implications for medical image analysis. Accurate and automated segmentation of nuclei can streamline pathological analysis, aid in cancer diagnosis, and contribute to advancements in biomedical research.

1. **References**

* Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (pp. 234-241). Springer.
* Milletari, F., Navab, N., & Ahmadi, S. A. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In International Conference on 3D Vision (3DV) (pp. 565-571). IEEE.
* Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 39(4), 640-651.