UNet-Plus: Hybrid Segmentation Models for Oral Cancer Imaging

Overview

The central thrust of this project is the application of image segmentation techniques to a dataset of oral cancer images sourced from Kaggle. The project employs Convolutional Neural Networks (CNNs), specifically utilizing the U-Net architecture. Unlike typical CNNs, which are employed for image classification tasks, U-Net architectures are particularly potent for image segmentation problems.

To elevate the performance and efficiency of the U-Net models, various pretrained backbones, namely RESNET34, RESNET101, InceptionNETV3, and EfficientNet, were integrated into the architecture. These pretrained models serve as the encoder in the U-Net architecture, aiding in the extraction of essential features from the images. The decoder then performs the actual segmentation based on these features.

The Intersection over Union (IOU) score is used as a key performance metric for evaluating the effectiveness of each segmentation model. Given its robustness in evaluating how well the predicted and actual segments overlap, the IOU score is highly valuable in this context. To further optimize the models, a weighted average approach is used to create hybrid models that combine the strengths of individual architectures, thereby aiming to achieve superior IOU scores.

Key Concepts

Image Segmentation: This technique involves partitioning an image into different segments or "labels" to simplify image analysis. In this project, segmentation plays a critical role in isolating and identifying oral cancer features. According to a review article by Haralick and Shapiro, image segmentation is often the first step in image analysis and understanding. Haralick and Shapiro, 1985

U-Net Architecture: A specialized Convolutional Neural Network originally designed for biomedical image segmentation. It uses a symmetric encoder-decoder structure and is particularly effective in tasks where the input and output are of the same dimensions. A research paper by Ronneberger et al. substantiates its effectiveness in medical imaging. Ronneberger, Fischer, & Brox, 2015

Pretrained Models: These are neural network models initially trained on a large, general dataset and then fine-tuned for a specific task. They provide a shortcut to achieving high performance without requiring extensive computational resources. A study by Yosinski et al. discusses the transferability of features in deep neural networks, reinforcing the value of pretrained models. Yosinski et al., 2014

Backbone models RESNET34, RESNET101, InceptionNETV3, EfficientNet: These are the different pretrained neural network architectures that were harnessed as the encoder (the feature extraction part) in the U-Net models used in the project. Each architecture brings its unique advantages to the table, such as RESNET's residual connections that help in training deeper networks and InceptionNET's inception modules that allow for more efficient feature extraction.Their applications and architectures are discussed in various papers He et al., 2016, Szegedy et al., 2016, Tan & Le, 2019

Intersection Over Union (IOU) Score: This metric quantifies the overlap between the predicted segmentation and the ground truth. It calculates the ratio of the area of overlap to the area of the union of the two sets. It has been highlighted as an important measure for evaluating the quality of image segmentation tasks in various studies.

Weighted Average Approach: This technique involves fusing multiple models to create a single, more robust model. The "weights" are determined based on each model's performance, particularly their IOU scores in this project. This way, each contributing model has an influence proportional to its accuracy. The concept is closely related to ensemble learning methods, which leverage multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

Additional Notes on Project Environment and Code

Google Collab as the Development Environment

For this project, Google Collab was chosen as the development environment due to its ease of use and availability of substantial computational resources, including GPUs. The cloud-based setting made it convenient to access and run the code from different locations and devices, offering flexibility during the development phase.

Data and Model Availability

It's important to note that the GitHub repository hosting this code does not contain the dataset used for the project, nor does it host any of the trained models or predictions. While the code serves as a reference, it's not intended for public use of models or predictions. However, some example outputs and graphs are provided both in the repository and in the document to offer a level of understanding of the work conducted.

Code Documentation

The code lacks extensive documentation, and some sections are deprecated or not in use as the project evolved over time. While comments have been added to facilitate understanding, the code itself is not as polished or well-documented as one might expect for a public release. This makes the present documentation all the more crucial for anyone looking to understand the procedures, methodologies, and outcomes detailed in this project.

In essence, this documentation serves as a comprehensive guide to understanding the segmentation models used for oral cancer imaging, offering clarity where the code may lack immediate comprehensibility.

Code Overview

The code implementation for the project "UNet-Plus: Hybrid Segmentation Models for Oral Cancer Imaging" can be broadly segmented into initial setup, data preparation, and model training phases. The key steps in each of these phases are outlined below: ---------- nedd to – rewrite ----------

Initial Setup

Library Importation and Environment Configuration: The first step involves importing all necessary libraries and modules for the project. This includes utilities for image processing such as skimage.io, deep learning frameworks

like Keras, and various other packages for data manipulation and visualization. A GitHub repository is cloned which contains pretrained segmentation models, facilitating a quicker start to the implementation.

Data Preparation

Data Loading: A NumPy array is initialized to store the images that form the dataset. Google Colab's drive environment is mounted for accessing the dataset stored in specific folders. This provides the necessary data accessibility required for image processing tasks.

Image Resizing: Due to varying dimensions of the raw images, all are resized to a common size of 224×224 pixels. These resized image arrays are stored back in a separate folder within the drive. This approach significantly enhances computational efficiency, ensuring that resizing doesn't have to be repeatedly performed.

Ground Truth Labeling: Ground truth labels for each of the 180 images are manually created using a specialized segmentation image website. This platform allows the user to upload the dataset, define labels, and then manually annotate specific regions of each image. In this case, cancerous tissues are marked as 1, while non-cancerous areas are designated as 0. The resultant ground truth label for each image thus has dimensions 224×224×1.

Ground Truth Storage: These ground truth labels are stored in a specific folder within the drive and are then loaded into a NumPy array. This prepares the labels for the subsequent training and testing splits.

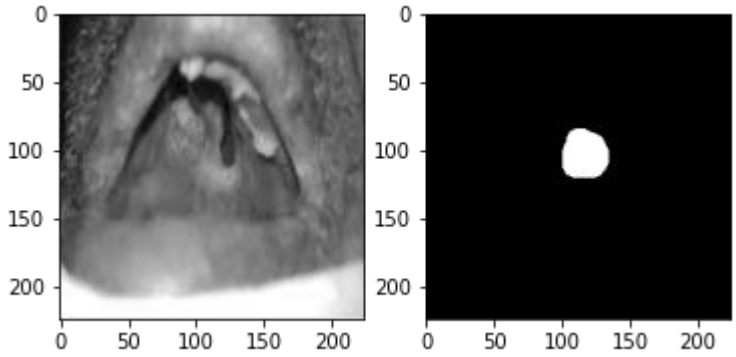

Fig1: Example of resized oral cancer image and its ground truth.

Model Training Preparation

Data Splitting: The image and label arrays are divided into training and testing sets. The train_test_split function from sklearn.model_selection is used for this purpose. This ensures that the model can be rigorously evaluated on unseen data, providing a more reliable measure of its performance.

U-Net Model Initialization

Parameter Configuration: At the start, shared parameters across all models are set. This includes setting the number of classes (n_classes) to 1, choosing the activation function as 'sigmoid', and defining a learning rate (LR) for the Adam optimizer.

Loss Function Design: For evaluating the segmentation performance, a customized loss function is formed by combining Dice Loss and Categorical Focal Loss. Dice loss, in particular, is weighted across classes. The total loss function (total_loss) is then formulated as the sum of Dice Loss and Focal Loss.

Metric Configuration: Intersection Over Union (IOU) Score and F-Score with a threshold of 0.5 are set as metrics for model evaluation. These are common metrics used for segmentation tasks.

Preprocessing: The input data is preprocessed using the specific preprocessing function (preprocess_input1) for ResNet34.

Model Architecture: The U-Net model is initialized using a ResNet34 backbone, and it's compiled with the chosen optimizer, loss function, and metrics.

Training: The model is then trained on the training data using a batch size of 16 for 50 epochs. The term "epoch" refers to one complete forward and backward pass of all the training examples. The model's performance is validated using a validation set (X_test, y_test).

Model Assessment and Storage

History: All the metrics and loss values during training are stored in a Pandas DataFrame for future analysis.

File Storage: This DataFrame is saved as a CSV file, and the trained model is also saved in the HDF5 file format in Google Drive.

Performance Visualization: A graph is generated to show the IOU Score of the model on both the training and validation sets across epochs.

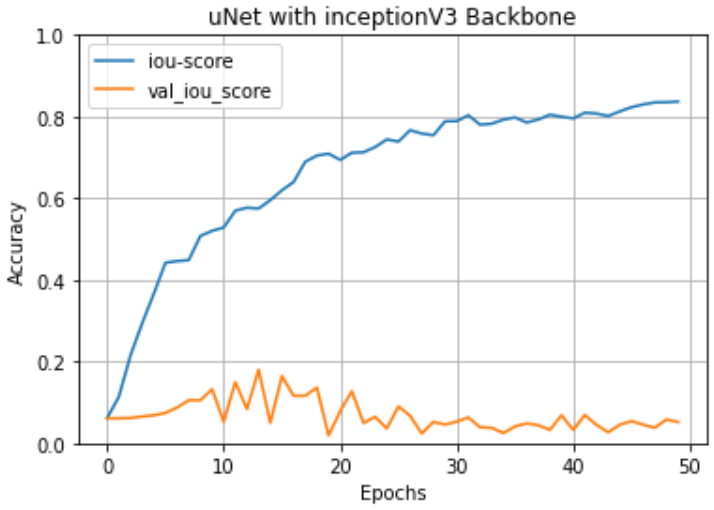

Fig 2: Example of comparative IOU score for each epochs of training

Multiple Models

This process is repeated for other pre-trained models like InceptionV3, ResNet101, ResNeXt, and EfficientNet. Given that training each model on the dataset takes around 5-6 hours, the trained models and their evaluations are saved in Google Drive for future use.

Model Retrieval

The pretrained models, including ResNet34, InceptionV3, ResNet101, ResNeXt, and EfficientNet, are subsequently loaded from Google Drive for a more detailed evaluation. These models were previously trained for a period of around 5-6 hours each and their configurations, alongside their training history, were saved in HDF5 and CSV formats, respectively. The aim here is to eschew retraining in future sessions and directly employ these models for inference tasks.

Intersection Over Union (IOU)

IOU is employed as a key metric for semantic segmentation tasks. It provides a robust measure to compare the overlap between the predicted segmentation and the actual ground truth. This metric ranges from 0 to 1, where a higher value indicates better model performance.

For our purpose, we use the Keras built-in MeanIoU class for this calculation. We also write custom code to calculate IOU by taking logical AND and OR operations between the predicted and ground truth arrays. Both the built-in and custom methods are utilized for cross-verification.

Thresholding and Binary Conversion

Predictions from the models are also subjected to a thresholding operation, which enables us to convert the model's output to a binary format. A predetermined threshold of 0.5 is used for this purpose. Values above this threshold are set to 1 and those below are set to 0, simplifying the subsequent calculation of IOU.

```
# Code snippet for Thresholding

pred_threshold = 0.5

pred1_b = np.where(pred1[:,:,:,:] > pred_threshold, 1, 0)
```

Comparative IOU Scores

The IOU scores for each model are calculated on the same set of images to ensure a consistent comparison. The scores reveal the efficacy of each model in the segmentation task, providing a clear landscape for choosing the most effective architecture for further investigation. By closely observing the IOU scores, it becomes evident which models are better suited for this specific task, thereby aiding in the decision-making process for subsequent steps in the research pipeline.

Prediction Normalization

To obtain a standardized output across various models, the predictions are normalized by setting a fixed threshold, hereby denoted as $\alpha=0.5$. This ensures that the predicted values are discretized into binary format, with values greater than α set to 1 and those lesser set to 0. The resultant predictions pred array are binary masks, which can now be directly compared with ground truth masks.

Logical Operations for IOU

Intersection Over Union (IOU) is computed using the logical AND (*) and OR (+) operations between the ground truth masks and the normalized predictions. For the AND operation, a threshold of $\beta=0.5$ is set to define the overlapping regions, while a threshold of $\gamma=0.99$ is used for the OR operation to identify the union regions.

The IOU score for each model is then calculated as:

$$IOU_i = union_i / overlap_I$$

Upon applying the above-described methodology, the IOU scores obtained are as follows:

IOU Score for Model 1: 0.6544

IOU Score for Model 2: 0.6686

IOU Score for Model 4: 0.6328

Weighted Average for IOU Optimization

Weighted averaging is employed to improve the Intersection Over Union (IOU) score by combining predictions from multiple models. This involves assigning different weights to each model's output and summing them up to produce a composite prediction. The aim is to maximize the IOU score by tuning these weights. Initially, a predefined set of weights was used to calculate the IOU to evaluate the efficacy of the ensemble method.

Subsequently, a grid search algorithm was applied to discover the optimal set of weights that would yield the highest IOU score. This algorithm iteratively tested different combinations of weights, storing the resulting IOU scores in a Pandas DataFrame. This provided a mechanism to identify the best-performing weight set by querying the DataFrame for the maximum IOU score.

The results indicated an enhancement in the maximum IOU score when employing the weighted average approach, as compared to the scores from the individual models. This highlighted the effectiveness of ensemble methods in improving model performance in tasks such as semantic segmentation.

The methodology followed—beginning with the calculation of individual model IOU scores, transitioning to a weighted average ensemble approach, and then optimizing the weights through grid search—offers a comprehensive and robust procedure for performance optimization.

The detailed methodology extends to applying the model ensemble to real-world data, specifically for oral cancer images. The optimal weights obtained from the grid search, [0.6,0.9,0.4][0.6,0.9,0.4], were incorporated into the

ensemble prediction formula. This weighted prediction formula was then applied to new oral cancer images as an ultimate test of the model's capabilities.

The images, initially in a default size, were resized to 224×224224×224 pixels to be compatible with the model's input dimensions. After resizing, these images were subjected to the ensemble model for prediction. The model's output was then binarized based on a threshold of 0.5 to obtain the final prediction mask.

Finally, the resulting predictions were visually assessed by overlaying them on the actual input images. Two overlaying sets were created for each image: one with the ground truth and another with the model's prediction. This direct visual comparison enables a qualitative evaluation of the model's performance, supplementing the quantitative IOU scores previously calculated.

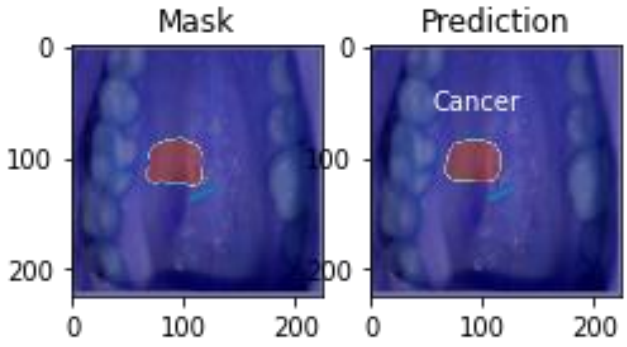

Fig: Example of hybrid model predicted image alongside ground truth

The outlined procedure not only validates the model on new data but also provides a mechanism to visually interpret the model's capabilities, thereby achieving a comprehensive performance assessment.

Confusion Matrix

A Confusion Matrix is a table that is used to evaluate the performance of a classification model. It contains information about actual and predicted classifications, facilitating the computation of various performance metrics like accuracy, precision, recall, etc. In the binary segmentation scenario, four categories are of primary interest:

- True Positives (TP): The number of pixels correctly predicted as belonging to the target class.
- True Negatives (TN): The number of pixels correctly predicted as not belonging to the target class.
- False Positives (FP): The number of pixels wrongly predicted as belonging to the target class.
- False Negatives (FN): The number of pixels wrongly predicted as not belonging to the target class.

In this case, a seaborn heatmap visualizes the Confusion Matrix, providing a bird's-eye view of the model's prediction effectiveness. Additionally, the False Positive Rate (FPR) and True Positive Rate (TPR) are computed from the Confusion Matrix to prepare for ROC Curve plotting.

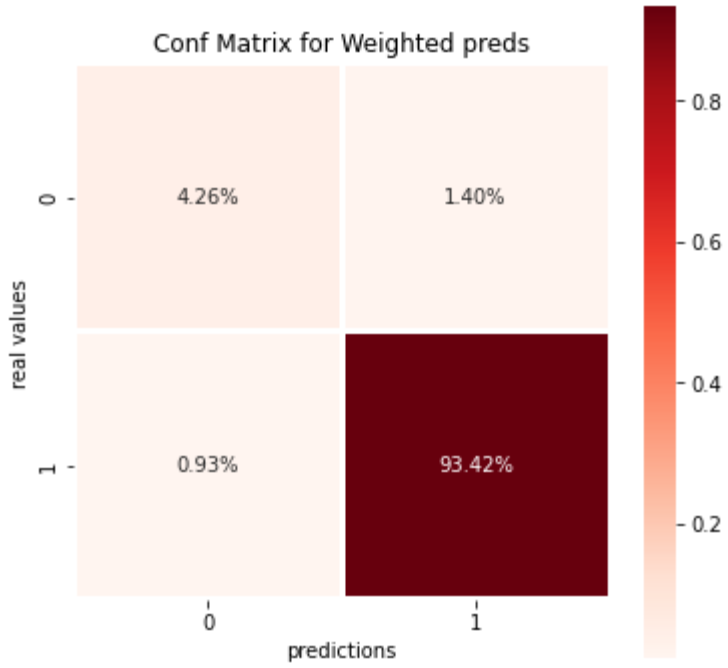

Fig: Confusion matrix for weighted prediction after grid search

ROC Curve

A Receiver Operating Characteristic (ROC) Curve is a graphical representation that illustrates the diagnostic ability of a binary classifier as its discrimination threshold varies. The ROC Curve is plotted with TPR against FPR for the various threshold points. The area under the ROC Curve, known as the AUC (Area Under the Curve), serves as an aggregate measure of performance across all possible classification thresholds.

In this specific implementation, ROC Curves are plotted for each pre-trained model (ResNet34, InceptionV3, ResNet101) involved in the ensemble. These are then collectively displayed to facilitate a comparative analysis of the individual models' effectiveness in distinguishing the target class.

To sum up, these additional performance metrics, especially when examined collectively, present a comprehensive understanding of the model ensemble's efficacy and reliability.

Conclusion

In summary, the documentation presents a comprehensive approach to tackle oral cancer image segmentation using an ensemble of pre-trained neural networks. It elaborates on the methodology from data preprocessing to model architecture selection and, ultimately, to performance evaluation.

The project makes use of ensemble learning to combine the strengths of different pre-trained models, namely ResNet34, InceptionV3, and ResNet101. By employing these diverse architectures, the ensemble model aims to leverage their individual strengths, thereby improving overall prediction accuracy. The weighted average method

is utilized to consolidate predictions from these models, and a rigorous grid search is conducted to find the most optimal set of weights that maximize the Intersection Over Union (IOU) score.

Evaluation metrics, including the IOU, Confusion Matrix, and ROC Curve, are meticulously explained and implemented. The Confusion Matrix provides a granular understanding of the model's true and false predictions, which is essential for diagnostic applications. The ROC Curves for each backbone model offer an insightful comparative analysis, revealing the model's capabilities across various discrimination thresholds.

The ensemble method successfully amalgamates the advantages of multiple pre-trained models, resulting in a robust and reliable system for oral cancer image segmentation. This approach shows promise for future enhancements and applications, positioning it as a noteworthy contribution to the field of medical image analysis.

References

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention* – MICCAI 2015.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning* (Vol. 97, pp. 6105-6114).