U-Net for Retinal Segmentation Using FIVES & RETA Datasets.

By: Hesham Zaky

Datasets

Fives Dataset

The FIVES dataset is a collection of fundus photographs that has been made available on Figshare. It contains a total of 800 images, out of which 600 images are recommended for training and the remaining 200 for testing. The dataset is divided into four categories: AMD, DR, Glaucoma, and Normal. Each image is accompanied by a corresponding ground truth image, and quality assessment scores are provided in a Microsoft Office Excel list.

RETA Dataset

The RETA benchmark dataset was utilized as a part of the project, consisting of 81 retinal fundus images and corresponding pixel-level blood vessel masks. This dataset was drawn from the first subset of Indian Diabetic Retinopathy Image Dataset (IDRiD) and constructed using the defined benchmark build protocols. The dataset was annotated using the Computer Aided Retinal Labelling (CARL) tool, which involved pixel-level, structure-level, and network-level annotation stages. The RETA benchmark dataset has been applied to develop and train an AI model according to U-Net architecture for retinal segmentation.

Methodology

Dataset augmentation

Data augmentation and manipulation was performed in Python using OpenCV and Albumentation libraries. A variety of transformations such as horizontal and vertical turnover, elastic transformation, grid deflection, and optical distortion were applied to the images. The original ground truth labels were preserved while creating new images from existing ones. The augmented dataset was saved as image and mask pairs and the images were resized to a specific height and width. The process was conducted on the training and testing datasets for RETA, which contained only 81 images, a relatively small number for developing an AI model. The main objective was to increase the diversity in training data to enhance the robustness and

generalization of the model, as it is a common approach to mitigate overfitting and improve model performance, especially when working with limited data.

U-Net architecture implementation

U-Net is a popular architecture for semantic segmentation, consisting of a contracting path and an expansive path. The contracting path consists of repeated 3x3 convolutions with ReLU activation, followed by 2x2 max pooling with stride 2 for downsampling. The expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution"), concatenation with the corresponding feature map from the contracting path, and two 3x3 convolutions with ReLU activation. The final layer uses a 1x1 convolution to map each feature vector to the desired number of classes. The network has 23 convolutional layers in total.

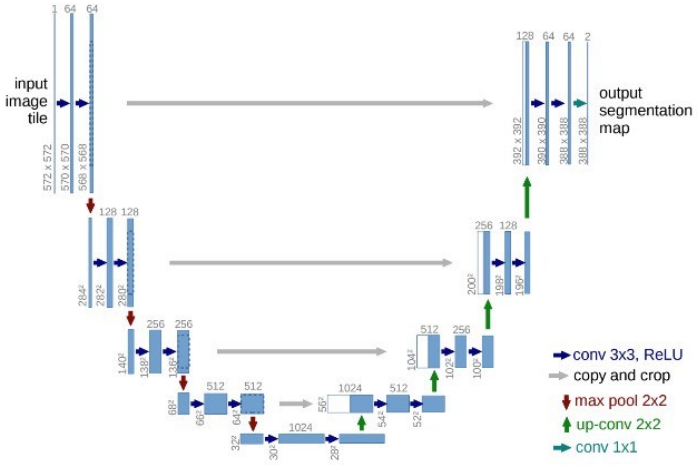

Figure 1 U-Net architecture

U-Net architecture was implemented in Python using the Keras library. Along with several functions that represent the building blocks of the U-Net architecture, including convolutional blocks, encoder blocks, and decoder blocks. See attached code **model.py** for Model implementation.

Training Model

In the training phase, a Python script was used to train the model using the U-Net architecture by importing libraries like OpenCV and TensorFlow and defining functions for creating directories, loading data, and shuffling data. The loaded image and mask data are preprocessed by resizing and scaling the pixel values. These processed images and masks are then used to train the model using RETA or FIVES dataset. The model is compiled with a custom loss function and evaluation metrics like Dice Coefficient, Intersection Over Union, Recall, and Precision. The script

also includes callback functions for saving the model, reducing the learning rate, logging data, and early stopping if validation loss doesn't improve. The model is trained with specified hyperparameters like learning rate, batch size, and number of epochs, and evaluated on a validation set after every epoch. The best model is saved in a CSV file.

Testing

In the testing phase, the model was loaded from the file "model.h5" using the Keras API. The test dataset was loaded from the "new_data/test" directory using the "load_data" function. The model was used to make predictions on each test image and the results were saved to the "results" directory using the "save_results" function. The performance of the model was evaluated by calculating several metrics including accuracy, F1-score, Jaccard similarity score, recall, and precision using the ground truth masks and the predicted masks. The metrics were calculated for each test image and the mean values were reported. Finally, the performance metrics were saved to the "files/score.csv" file for further analysis. ***Result***

Results have been analyzed for the RETA and FIVES datasets to collect these metrics.

Table 1 Results for U-net for both RETA and Fives Datasets

DATASET	ACCURACY	F1 SCORE	JACCARD	RECALL	PRECISION
RETA	0.94966	0.58555	0.4144	0.86872	0.44216
FIVES	0.96415	0.65845	0.49434	0.88482	0.52848

The model is performing well on both datasets in terms of accuracy from this table. The model got an accuracy of 0.94966 for the **RETA** sample; in that case, it accurately identified 94.97% of the test set. According to the F1 score of 0.58555, the model balanced accuracy and recall, with a tendency towards recall. The Jaccard's score of 0.40144 indicates a slight overlap between the prediction label and the actual, but it does not seem too great. The recall score of 0.86872 is very high, indicating that the model could identify true positives. However, the precision score of 0.44216 is very low, indicating that there were many false positives.

The model was also more accurate, with a score of 0.96415, and correctly determined 96.4% of the test set in the **FIVES** dataset. The F1 score of 0.65845 suggests that the model achieved a better balance between precision and recall compared to the RETA dataset. The Jaccard score was higher than that of the RETA dataset by 0.49434, indicating a better correlation between predicted and accurate labels. The recall score of 0.88482 is again high, and the precision score of 0.52848 is also higher than that of the RETA dataset, indicating a lower number of false positives.