PROJECT ABSTRACT

Retinal Vessel Segmentation using U-Net

Retinal blood-vessel segmentation with U-Net on the DRIVE dataset: patch training, BCE+Dice loss, Dice/IoU evaluation and an interactive segmentation viewer. Built-to-order with code, training notebook, report, PPT and viva kit.

01. Title

Retinal Vessel Segmentation using U-Net

02. Introduction / Background

The retinal vessel tree is one of the few places vasculature can be photographed directly, and its shape carries early signatures of diabetic retinopathy, glaucoma and hypertension. Manual tracing — the gold standard for training data — takes an expert nearly two hours per image, which is why automated segmentation has been a core medical-imaging problem for two decades. The DRIVE dataset (40 fundus images with expert manual tracings, Staal et al., 2004) is the field's standard benchmark, and U-Net (Ronneberger et al., 2015) remains the reference architecture: an encoder-decoder with skip connections preserving the fine detail thin capillaries demand. This project implements that pipeline faithfully, from CLAHE preprocessing to Dice-based evaluation.

03. Problem Statement

Vessel pixels are outnumbered ~10:1 by background, training images number only 20, and plain cross-entropy plus naive full-image training fails on both counts. The project addresses this with the literature-standard recipe — green-channel extraction, CLAHE, 48x48 patch training with augmentation, and a combined BCE+Dice loss — logging Dice, IoU, sensitivity and specificity per epoch, and wrapping the result in an interactive viewer (probability maps, threshold slider, overlays, ground-truth comparison) plus a STARE cross-dataset check.

04. Objectives

- Train a PyTorch U-Net on DRIVE (20 train images) with patch training and BCE+Dice loss.
- Reach a design target of ~0.79 Dice on the 20-image DRIVE test split, logging Dice, IoU, sensitivity, specificity and ROC-AUC per epoch.
- Build the segmentation viewer: probability map, live threshold slider, overlay and ground-truth-vs-prediction comparison.

- Measure cross-dataset transfer on STARE and deliver code, notebook, report, PPT and viva Q&A.

05. Proposed Methodology

Five stages. (1) Preprocessing: green-channel extraction, CLAHE contrast enhancement, field-of-view masks. (2) Patch sampling: 48x48 patches (10k/epoch) with rotation, flip, elastic deformation and gamma jitter. (3) Model: U-Net encoder-decoder with skip connections, single-channel input, sigmoid output. (4) Training: Adam 1e-3, ~60 epochs, combined BCE+Dice loss, best-checkpoint selection on validation Dice. (5) Evaluation: patch-wise inference stitched over the 20 test images, thresholded at 0.5, scored against expert tracings; STARE leave-one-out transfer measured.

06. System Architecture / Working

Data layer: DRIVE/STARE images and manual masks. Preprocessing layer: green channel, CLAHE, patch extractor with augmentation. Model layer: U-Net in PyTorch. Training layer: the notebook runs the 60-epoch loop, logging loss and Dice/IoU curves. Inference layer: patch-wise prediction, stitching and thresholding produce binary masks. Viewer layer: the HTML/JS app renders probability maps with a live threshold slider, overlays and ground-truth comparisons.

07. Hardware / Software Requirements

- PC with NVIDIA GPU recommended (16 GB RAM); CPU training possible but slow
- Python 3.10, PyTorch, OpenCV, NumPy
- Matplotlib (curves, visualizations), Jupyter (notebook)
- DRIVE and STARE datasets (public download, documented in setup guide)
- A modern browser for the segmentation viewer

08. Expected Outcomes

A trained U-Net near the ~0.79 Dice design target on DRIVE test, full per-epoch metric logs, STARE transfer measurement, an error analysis by vessel caliber, and the interactive viewer. This project is an academic research prototype and is not a medical device; it must not be used for diagnosis or clinical decisions.

09. Applications

- Teaching semantic segmentation and U-Net
- Reference pipeline for other biomedical segmentation tasks
- Demonstration of handling extreme class imbalance
- Baseline for architecture-variant comparisons (Attention U-Net, UNet++)

10. Future Scope

- Attention U-Net / UNet++ variant comparison
- Uncertainty estimation via Monte-Carlo dropout
- Vessel width / tortuosity quantification on top of masks
- Multi-dataset joint training study