13-14 June 2025

Meet the team

Team Prometheus

Guilherme Grancho

Full-Stack Developer

Vasco Pereira

Machine Learning Developer

Agenda

CHALLENGE OVERVIEW

Segment the blood vessels on slit-lamp images of the eye. Use the best computer vision approaches to solve this task.

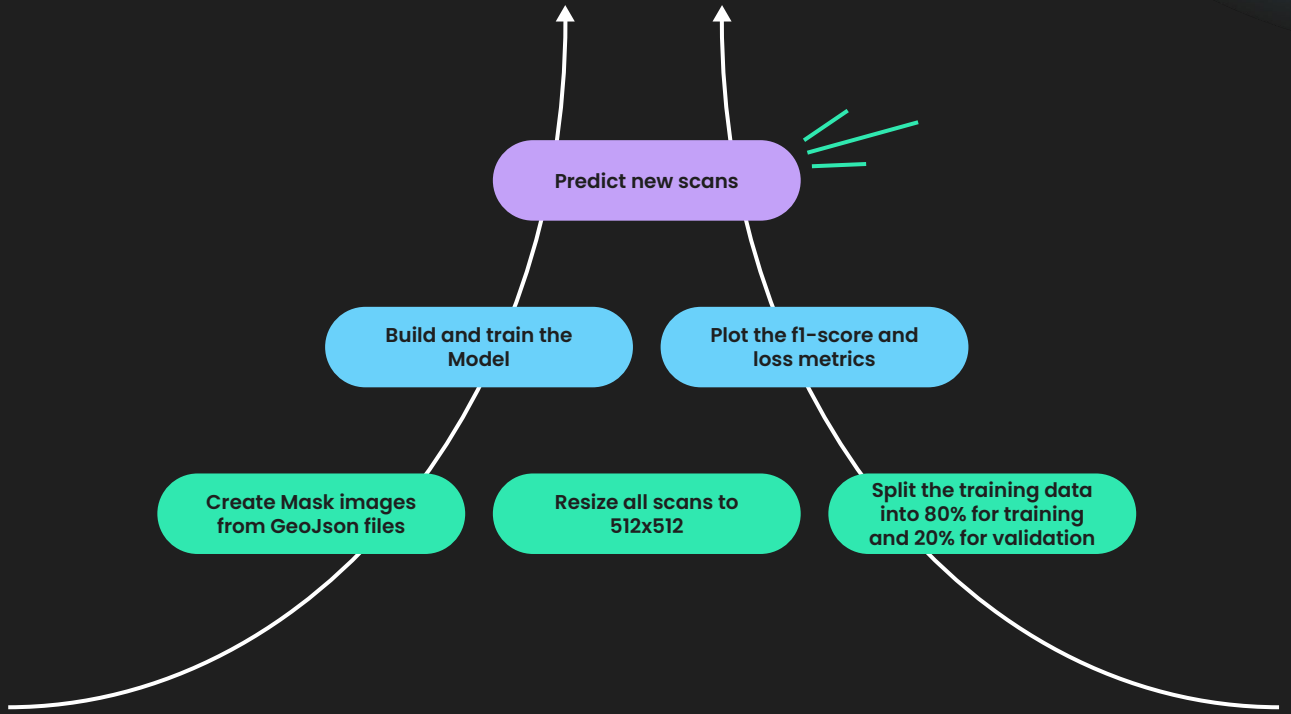
Developer Journey

Step 1	Step 2	Step 3	Step 4	Step 5
Research of machine learning solutions and new approaches in the field.	Developing and testing the machine learning model implemented.	Create an API using the machine learning model created.	Creating a frontend that retrieves the information it needs from the previously created API.	Deploy the UI using cloud services providers and deploy services.

Proposed Solution

We implemented a **U-Net + EfficientNetB7** pipeline, leveraging a pre-trained EfficientNet model as the powerful encoder to extract rich hierarchical features. These features were then fed into a standard U-Net decoder with skip connections, ensuring precise and efficient segmentation.

Solution Steps



Metrics utilized

Dice Loss

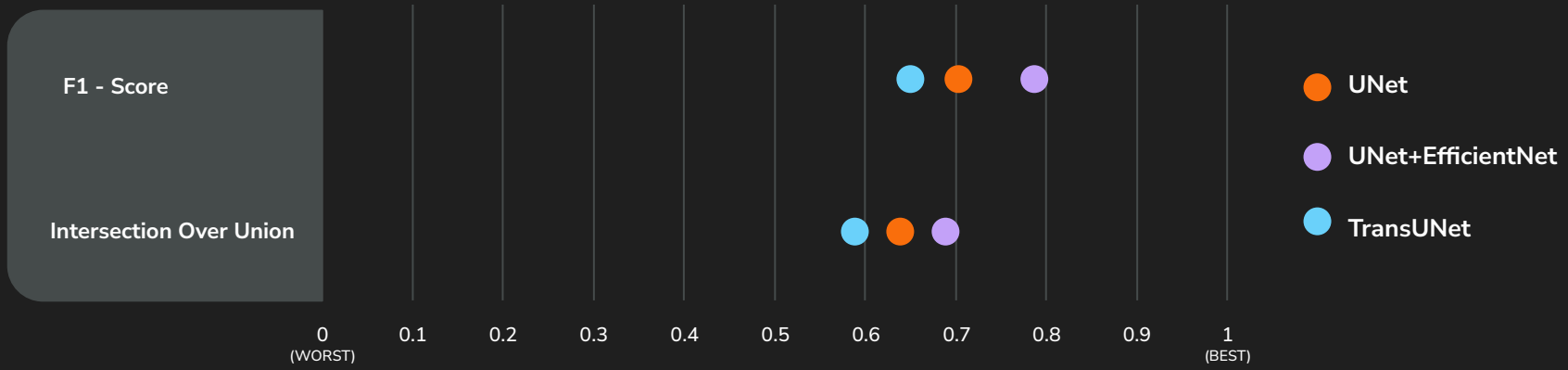
Binary Cross-Entropy Loss

Model CheckPoint Callback

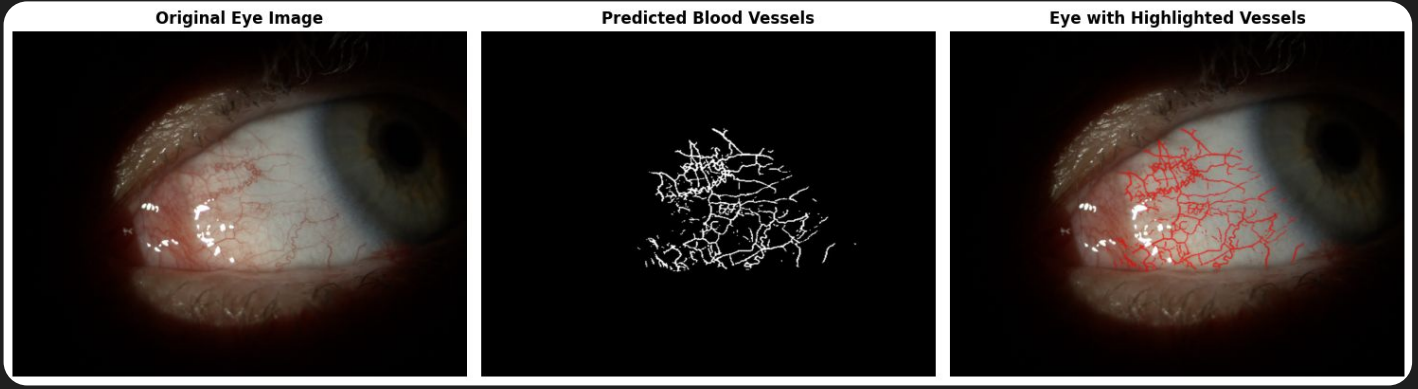
Reduce LR on Plateau Callback

Early Stopping Callback

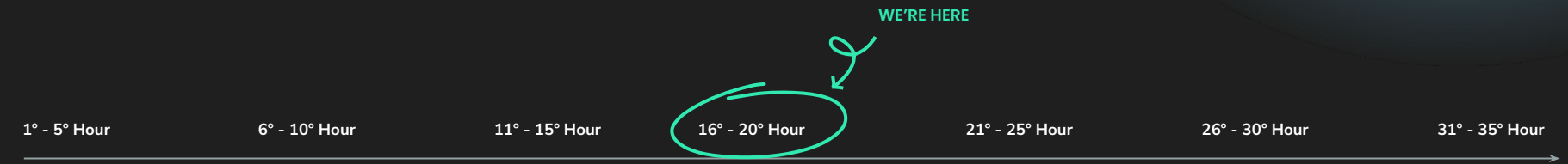
Performance Analysis



Output Analysis



User Interface



Backend

FastAPI & Docker
Spin up the entire vessel-segmentation service with a single docker-compose up.
Send slit-lamp images to FastAPI's /predict endpoint and receive high-fidelity vessel masks instantly.

Frontend

Next.js
The Next.js frontend delivers a responsive SPA with drag-and-drop uploads and real-time vessel-mask overlays. Interactive controls (threshold slider, download buttons) let users fine-tune and export segmentation results directly in the browser.

Backend - Cloud Service

AWS
Host the Dockerized FastAPI backend on AWS ECS/Fargate or EKS for auto-scaling inference.
Use API Gateway, S3, and CloudWatch for secure endpoints, storage, and monitoring.

Frontend - Deploy Service

Vercel
Deploy the Next.js frontend to Vercel with one-click Git integration and instant global CDN.
Get preview URLs on every push plus zero-config SSL, routing, and edge caching out of the box.

Video Demo: Guided Overview

GitHub Repo: Execution and launch

Our project leverages GitHub to streamline development, collaboration, and deployment:

- **Version Control & Collaboration**: Host code repos, manage branches, and review changes via pull requests.
- **Issue Tracking**: Organize tasks, report bugs, and plan milestones through GitHub Issues and Projects.
- **CI/CD Pipeline**s: Automate testing, linting, and Docker image builds with GitHub Actions on every push.
- **Preview Environments**: Generate live frontend previews (Vercel) and backend staging deployments for each PR.

Project Dashboard

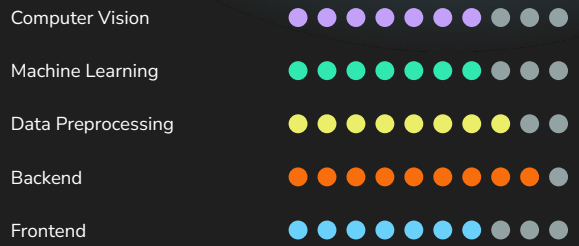
CHALLENGES

- Segment eye blood vessels
- Use computer vision approaches
- Novel approach

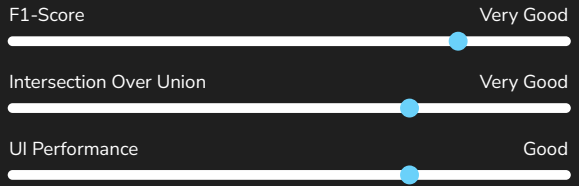
DIFFICULTIES

- Dataset preprocessing
- Machine learning architecture
- UI design and development

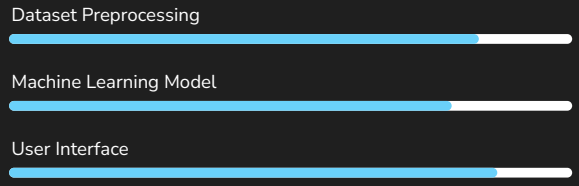
LEARNING CURVE



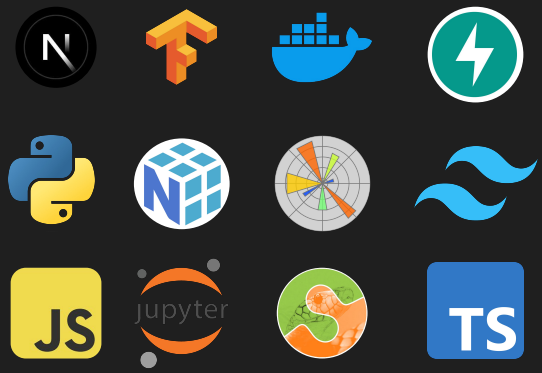
PERFORMANCE METRICS



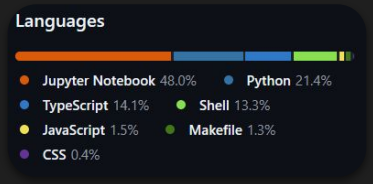
ACHIEVEMENTS



TECH-STACK



REPO STATS



TECH RESOURCES

Next steps

Step 1

Finetune the model using architectures that take more time to train - e.g., FSG-Net, Swin-UNet, e.t.c.

Step 2

Deploy the FastAPI on a cloud provider - e.g., AWS, Azure, Google.

Step 3

Deploy live the Next.js web app using a service like Vercel or GitHub Pages.

Thank you!

Contact the Prometheus Team

guilhermegranchopro@gmail.com

vasco.serpa.pereira@gmail.com