# **Biweekly Report**

Project Title: Diagnosis of Diabetic Retinopathy

Time Period: April 7 – April 20, 2025

Team Leader: Beining Wang

## **Total Working Hours**

Estimated 30 hours per person × 4 = 60 hours

## **Project Progress Overview**

During this biweekly period, our team progressed steadily toward the core implementation phase of the diabetic retinopathy diagnosis system. The main work focused on two technical streams: (1) deep learning model development for classification and segmentation, and (2) user interface design and system architecture planning.

## **Ongoing Work**

• Model Reproduction & Optimization:  
 A classification pipeline based on ResNet50 was optimized using transfer learning strategies. Key improvements include image augmentation, layer unfreezing, and threshold calibration. The model demonstrated enhanced prediction performance with improved Cohen’s Kappa metrics.

**• Semantic Segmentation of Lesions:** A U-Net++ model with EfficientNet encoder was trained on IDRiD dataset for hard exudate segmentation. The process involved advanced augmentation, Dice-BCE-Focal hybrid loss, and visual evaluation using Dice coefficient (~0.62). Early stopping and TensorBoard tracking were also implemented.

**• Existing System Research:** A comparative review of platforms such as iCare RETCAD, RetinaLyze, and Google ARDA provided valuable insights into UI presentation, workflow optimization, and result interpretation.

**• Front-End Interface Design:** The interface was designed to support role-based workflows for doctors and patients. Key features include image upload, diagnosis result visualization, case history browsing, and auto-generated PDF reports. The technical stack was finalized as Vue 3 + Vite + Element Plus + Pinia. A set of mockups and UI layout plans was completed.

## **Next Steps**

• Begin implementation of front-end components and routing  
• Complete backend API definitions for model inference and case records  
• Integrate model output with frontend diagnosis module  
• Explore extended segmentation tasks for comprehensive DR screening

Project Title: Diagnosis of Diabetic Retinopathy System  
Team Member: Peijin Chen  
Time Period: April 7 – April 20, 2025  
Working Hours: 30 hours

**1. Literature Review**

I reviewed two Kaggle notebooks related to DR classification:

* APTOS DR EDA & Starter: Covered data distribution, image preprocessing, and data loading structure using PyTorch.
* CNN for DR Diagnosis (PyTorch): Demonstrated a complete CNN-based workflow with QWK as the evaluation metric.

These readings provided insight into how data preparation and model output formats align with the planned system’s UI integration.

**2. Reference System UI Survey**

Analyzed existing commercial platforms to inform UI design:

* iCare RETCAD: Focused on lesion visualization and diagnostic clarity.
* RetinaLyze: Emphasized upload efficiency and integration with clinical systems.
* Google Health ARDA: Highlighted the value of minimal and mobile-friendly interfaces.

These references helped guide interface layout and functional planning.

**3. Front-End Design Progress**

Finalized initial design goals and technical structure:

* User Roles: Doctor and patient views with separate dashboards.
* Core Modules: Login, image upload, diagnosis result display, history list, report generation.
* Technology Stack: Vue 3 + Vite + Element Plus, with Pinia for state management.
* Design Principles: Responsive layout, clean UI, planned integration of image heatmap and PDF report functions.

**4. Next Steps**

* Build initial Vue components and route structure.
* Implement upload and result preview modules.
* Define back-end interfaces for diagnosis results.
* Integrate model output with front-end logic.

**Biweekly Report: Diagnosis of Diabetic Retinopathy - Chenyu Huang**

**Total Work Hours: 30 hours [3.24 - 4.20]**

**Work Overview**

In the past two weeks (4.7-4.20), I continued the progress made previously, focusing mainly on the reproduction of the paper [APTOS Diabetic Retinopathy (EDA & starter)](https://www.kaggle.com/code/tanlikesmath/intro-aptos-diabetic-retinopathy-eda-starter) in the *Kaggle* competition.

I performed image augmentation using GPU on the *AutoDL* platform and adopted a transfer learning strategy, retaining the model based on the *ResNet-50* architecture. After achieving satisfactory results, I unfroze all layers for model fine-tuning and prediction optimization. Ultimately, I obtained a model with a relatively high recognition success rate.

**Task Progress**

1. **Image Augmentation**
   1. Resize the images to a smaller size (batch size = 64, image size = 224).
   2. Use *fastai.vision.get\_transforms()* to define image augmentation. Next, the image data is loaded and randomly split into the training set (80%) and the validation set (20%), followed by image augmentation, batching, and normalization.
2. **Training (Transfer Learning)**
   1. Use *Cohen’s quadratically weighted kappa*, which is a better metric when dealing with imbalanced datasets like this one.
   2. Use transfer learning, where it retrained the last layers of a pre-trained neural network. I use the ResNet50 architecture trained on the ImageNet data set.
   3. Plot the learning rate curve to identify the learning rate that results in the fastest decrease in loss, and use this learning rate for training to optimize the model training process.
3. **Model Optimization**
   1. Unfreeze all layers of the model for fine-tuning.
   2. Similarly, plot the learning rate curve to find the learning rate that results in the fastest decrease in loss for training (using the one-cycle learning rate scheduling strategy), and then export the trained model.
   3. Evaluate and interpret the model results by plotting the confusion matrix.
   4. Optimize the threshold so that the predicted values can be correctly classified into multiple discrete categories, and evaluate the classification performance using Cohen's Kappa.

 