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

**Total Work Hours: 15 hours [4.7 - 4.20]**

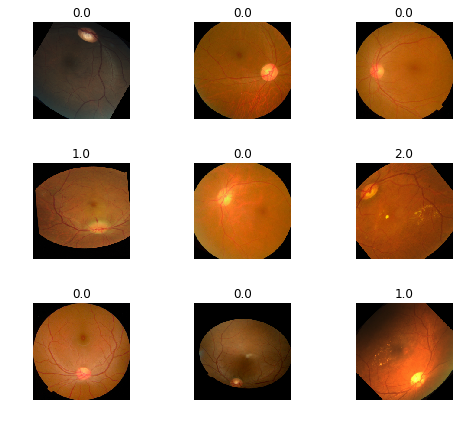
**Work Overview**

In the past two weeks, 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.

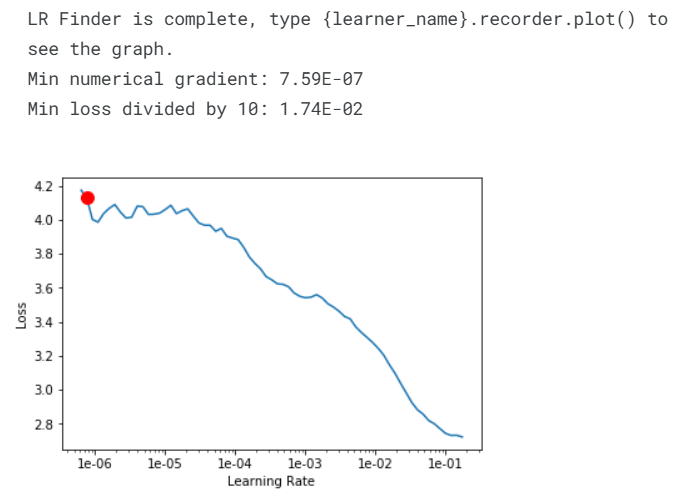
I performed image augmentation using GPU on the *AutoDL* platform and adopted a transfer learning strategy, retaining the model based on the *ResNet-50* architecture. After achieving satisfactory results, I unfroze all layers for model fine-tuning and prediction optimization. Ultimately, I obtained a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

**Task Progress**

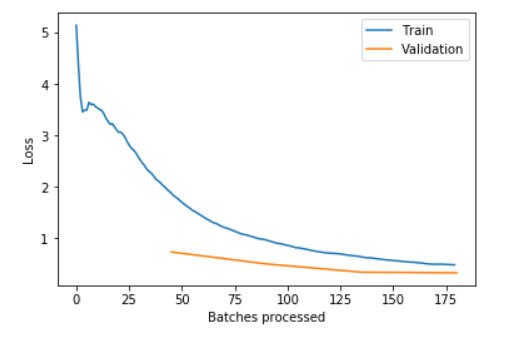
1. **Image Augmentation**
   1. Resize the images to a smaller size (batch size = 64, image size = 224).
   2. Use *fastai.vision.get\_transforms()* to define image augmentation. Next, the image data is loaded and randomly split into the training set (80%) and the validation set (20%), followed by image augmentation, batching, and normalization.  
      After completing the image augmentation and batching, one of the batches is shown below:



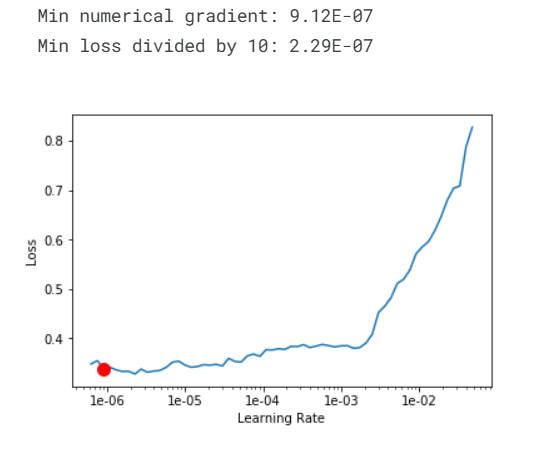
1. **Training (Transfer Learning)**
   1. Use *Cohen’s quadratically weighted kappa*, which is a better metric when dealing with imbalanced datasets like this one, and for measuring inter-rater agreement for categorical classification (the raters being the human-labeled data set and the neural network predictions).
   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, which has been commonly used for pre-training applications in computer vision.
   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.

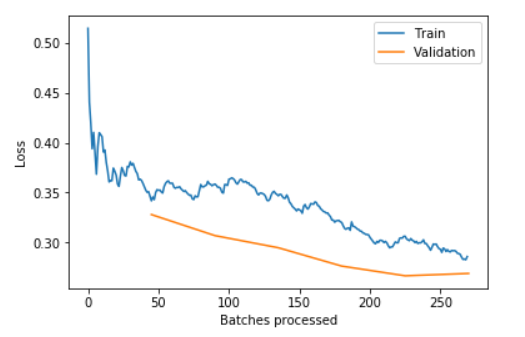


* 1. Also, the loss curves of train and validation during training are shown as follows:



1. **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.





* 1. Evaluate and interpret the model results by plotting the confusion matrix (which allows for a visual understanding of misclassified categories and those with better predictions).
  2. Optimize the threshold so that the predicted values (usually the probability outputs of the model) can be correctly classified into multiple discrete categories, and evaluate the classification performance using Cohen's Kappa.  
     Optimized recognition accuracy:



Using the optimized threshold can improve *Cohen's Kappa score* and optimize classification performance.Optimized threshold coefficient:

