Supplementary Materials for LONGL-Net: Temporal Correlation Structure Guided Deep Learning Model to Predict Longitudinal Age-related Macular Degeneration Severity

Alireza Ganjdanesh1,*, Jipeng Zhang2,*, Emily Y Chew3, Ying Ding2, Heng Huang1,§, Wei Chen2,4,§

1Department of Electrical and Computer Engineering, Swanson School of Engineering, University of Pittsburgh, Pittsburgh, PA.
2Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh, Pittsburgh, PA.
3Division of Epidemiology and Clinical Applications, National Eye Institute, National Institutes of Health, Bethesda, MD.
4Division of Pulmonary Medicine, Department of Pediatrics, UPMC Children’s Hospital of Pittsburgh, University of Pittsburgh, Pittsburgh, PA.

* equal contribution

§ To whom correspondence should be addressed.
Email: heng.huang@pitt.edu; wec47@pitt.edu
S1. Dataset Statistics after Down-sampling

Table S1 summarizes the number of pairs with 2, 3, and 4 years gap between the first time point and the future one used in training/validation/test sets in our experiments for the classifier and GAN model. In each experiment, the data partitions are disjoint in subject level, i.e., the pairs for each participant are only present in one of data partitions.

Table S1: Statistics of the dataset used for training the classifier after down-sampling. The numbers in the parenthesis in columns 4-6 indicate the histogram of the partition in each row. As the maximum follow-up visit length is 13 years, the largest possible value in the visit pairs is 26.

| T (Time Gap in Units of 6 Months), Partition | Pairs of visits used (if available) | Number of pairs available | Number of pairs $y_{ij} = 0$ | Number of pairs $y_{ij} = 1$ | Number of pairs $y_{ij} = 2$ |
|---------------------------------------------|------------------------------------|---------------------------|-------------------------------|-------------------------------|-------------------------------|
| 4, Total {0, 4}, ..., (22, 26}              | 16545                              | 8430 (50.9%)              | 1683 (10.2%)                 | 6432 (38.9%)                  |
| 4, Train {0, 4}, ..., (22, 26}              | 14774                              | 7543 (51%)                | 1503 (10.2%)                 | 5728 (38.8%)                  |
| 4, Validation {0, 4}, ..., (22, 26}        | 887                                | 448 (50.5%)               | 96 (10.8%)                   | 343 (38.7%)                   |
| 4, Test {0, 4}, ..., (22, 26}              | 884                                | 439 (49.7%)               | 84 (9.5%)                    | 361 (40.8%)                   |
| 6, Total {0, 6}, ..., (20, 26}             | 14951                              | 7795 (52.1%)              | 1999 (13.4%)                 | 5157 (34.5%)                  |
| 6, Train {0, 6}, ..., (20, 26}             | 13357                              | 6984 (52.3%)              | 1774 (13.3%)                 | 4599 (34.4%)                  |
| 6, Validation {0, 6}, ..., (20, 26}        | 817                                | 425 (52%)                 | 117 (14.3%)                  | 275 (33.7%)                   |
| 6, Test {0, 6}, ..., (20, 26}              | 777                                | 386 (49.7%)               | 108 (13.9%)                  | 283 (36.4%)                   |
| 8, Total {0, 8}, ..., (18, 26}             | 13654                              | 7304 (53.5%)              | 2180 (16%)                   | 4170 (30.5%)                  |
| 8, Train {0, 8}, ..., (18, 26}             | 12189                              | 6553 (53.8%)              | 1926 (15.8%)                 | 3710 (30.4%)                  |
| 8, Validation {0, 8}, ..., (18, 26}        | 747                                | 389 (52%)                 | 137 (18.4%)                  | 221 (29.6%)                   |
| 8, Test {0, 8}, ..., (18, 26}             | 718                                | 362 (50.4%)               | 117 (16.3%)                  | 239 (33.3%)                   |
S2. Pre-processed Data Examples

The following figures illustrate some original images from batches 2010 and 2014 in the dataset and their corresponding cropped and resized version used in our experiments.

**Batch 2010:**

1. Image Name: ‘1638_04_LE_F2_LS’

   Original:

   ![Original Image](image1)

   Figure S1: A fundus image from AREDS dataset.

   Preprocessed:

   ![Preprocessed Image](image2)

   Figure S2: A preprocessed fundus image from AREDS dataset.
2. Image Name: ‘5653_16_LE_F2_LS’

Original:

![Figure S3: A fundus image from AREDS dataset.](image1)

Preprocessed:

![Figure S4: A preprocessed fundus image from AREDS dataset.](image2)
Batch 2014:

1. Image Name: ‘52705_QUA_F2_LE_LS’

   Original:

   ![Figure S5: A fundus image from AREDS dataset.](image)

   Preprocessed:

   ![Figure S6: A preprocessed fundus image from AREDS dataset.](image)
2. Image Name: ‘52634_QUA_F2_LE_LS’

Original:

Figure S7: A fundus image from AREDS dataset.

Preprocessed:

Figure S8: A preprocessed fundus image from AREDS dataset.
S3. GAN Model Details

Generator

U-Net [1] architecture is used for the generator because of its skip connections that enable it to pass low level structural information of its input to output easier than encoder-decoder architectures in which all the information should get passed through the bottleneck layer.

The number of channels in the generator’s architecture (U-Net 256) [1, 2] is as follows:

\[
3 \text{ (input } RGB \text{ image)} \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 512 \rightarrow 512 \rightarrow 512 \rightarrow 512 \\
\rightarrow 1024 \rightarrow 1024 \rightarrow 1024 \rightarrow 1024 \rightarrow 512 \rightarrow 256 \rightarrow 128 \\
\rightarrow 3 \text{ (output } RGB \text{ image)}
\]

Discriminator

It has been observed [2, 3] that using patch discriminator which penalizes its input’s structure in the patch level rather than full image can still result in sharp realistic-looking (preserving high frequency information) generated images and also benefit from more computational efficiency because the size of patches can be smaller than the image. As mentioned in the paper, we used 70 * 70 ‘Patch’ discriminator network. The number of channels in the discriminator’s architecture is as follows:

\[
6 \text{ (3 for first time point image + 3 for predicted image of the generator)} \rightarrow 64 \\
\rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 1 \text{ (discriminator scores for each patch)}
\]

Network Initialization

We followed Isola et al. [2] for network initialization, i.e., the generator and discriminator are initialized from a Gaussian distribution with zero mean and standard deviation equal to 0.02.

S4. Best Hyperparameter Setting for Classification Models

Table S2 summarizes hyperparameter settings which have achieved highest validation AUC for the experiments with the pairs having 2, 3, and 4 years gap. We tuned training batch size as well as Adam optimizer’s [4] parameters: Learning Rate, $\beta_1$, $\beta_2$, and Weight Decay (coefficient of L2 regularization on the model’s parameters).
Table S2: Description of the hyperparameter settings which achieved highest validation AUC in each experiment.

| Model   | Batch Size | Learning Rate | $\beta_1$ | $\beta_2$ | Weight Decay |
|---------|------------|----------------|----------|----------|--------------|
| 2 years | 128        | 0.0002         | 0.9      | 0.999    | 0.00001      |
| 3 years | 32         | 0.0003         | 0.9      | 0.99     | 0.            |
| 4 years | 128        | 0.0003         | 0.5      | 0.999    | 0.            |

S5. GAN Model’s Longitudinal Predictions

Here, we provide more examples of our GAN model’s longitudinal prediction for the cases who 1) have not been in advanced AMD condition in the baseline and did not progress to it until the end of study and 2) have been in advanced AMD in the baseline, and as there is not treatment for late AMD so far, have remained in advanced AMD until the end of study. In each figure, the left image in the first row is the baseline image, and the next ones are the predictions for the 4, 6, and 8 time points after baseline. The second rows contain their corresponding ground truth images.

Figure S9: Longitudinal Prediction for the test image ‘52187_QUA_F2_LE_LS’
Figure S10: Longitudinal Prediction for the test image ‘52477_QUA_F2_LE_LS’

Figure S11: Longitudinal Prediction for the test image ‘52712_QUA_F2_LE_LS’
**S6. Images from UK Biobank used in our Experiments**

Here, we list the name of images that we selected from UKBiobank to perform the validation experiments of our model on an independent dataset.

- '2656719_21015_0_0.png'
- '3244608_21015_0_0.png'
- '2638597_21015_0_0.png'
- '1154900_21015_0_0.png'
- '4680067_21015_0_0.png'
- '5672573_21015_0_0.png'
- '5471567_21015_0_0.png'
- '1339583_21015_0_0.png'
- '4440932_21015_0_0.png'
- '4588793_21015_0_0.png'
- '5375555_21015_0_0.png'
- '5471567_21015_0_0.png'
- '4440932_21015_0_0.png'
- '4588793_21015_0_0.png'
S7. Model’s Prediction Results on UK Biobank Dataset

Table S3 provides statistics regarding the performance of our checkpoint models trained on AREDS dataset on the independent test dataset containing 300 images selected from UK Biobank dataset [5].

Table S3: Accuracy and Confusion Matrix of the models trained with the pairs with 2(A), 3(B), and 4(C) years gap between successive time points on the independent validation set chosen from UK Biobank dataset. In the confusion matrices, label ‘0’ is not being in advanced AMD and ‘1’ vice versa.

| Model | Accuracy | Confusion Matrix |
|-------|----------|-----------------|
| A | 0.8667 | [197 3 37 63] |
| B | 0.9233 | [191 9 14 86] |
| C | 0.9267 | [188 12 10 90] |

S8. Structure of Adding Age Information to the Inputs of the Classifier

To explore whether explicitly providing the age information to the model can improve the performance we concatenate age value to the representations of the outputs of the global average pooling layer in ResNet-18 [6] architecture. The procedure is shown in Figure S6.

Figure S14: The classifier architecture when using both fundus image and age information as its inputs.
S9. Classification Model’s Results on Grading the Fundus Images

We compare our model’s performance on the task of grading its input image with the late AMD grading module in DeepSeeNet [7].

Table S4: Our model’s classifier statistics for the task of grading the first time point image (advanced/not advanced AMD) in the pairs. We use the confusion matrix predictions of the model on the test set and combine classes $y \in \{0,1\}$ as both of these classes correspond to the first time point (input) not being in advanced AMD.

| T (Time Gap in Units of 6 Months) | Binarized Confusion Matrix | Accuracy | Sensitivity | Specificity |
|----------------------------------|---------------------------|----------|-------------|-------------|
| 4 (2 years)                      | [503 20]  
|                                 | 66 295]               | 0.902    | 0.817       | 0.962       |
| 6 (3 years)                      | [466 28]  
|                                 | 51 232]               | 0.898    | 0.819       | 0.943       |
| 8 (4 years)                      | [444 35]  
|                                 | 26 213]               | 0.915    | 0.891       | 0.926       |
References

1. Ronneberger, O., P. Fischer, and T. Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Cham: Springer International Publishing.
2. Isola, P., et al. *Image-to-image translation with conditional adversarial networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
3. Li, C. and M. Wand. *Precomputed real-time texture synthesis with markovian generative adversarial networks*. in *European conference on computer vision*. 2016. Springer.
4. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
5. Sudlow, C., et al., *UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age*. Plos med, 2015. 12(3): p. e1001779.
6. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
7. Peng, Y., et al., *DeepSeeNet: A Deep Learning Model for Automated Classification of Patient-based Age-related Macular Degeneration Severity from Color Fundus Photographs*. Ophthalmology, 2019. 126(4): p. 565-575.