**Readme**

Parallel discriminator generative adversarial network (P-GAN) is a TensorFlow implementation of an artificial intelligence (AI) method for recovering individual retinal pigment epithelial (RPE) cells from single speckled adaptive optics optical coherence tomography (AO-OCT) volume.

If any portion of this code is used, please cite the following paper in your publication:

Vineeta Das, Furu Zhang, Andrew J Bower, Joanne Li, Tao Liu, Nancy Aguilera, Bruno Alvisio, Zhuolin Liu, Daniel X Hammer and Johnny Tam, "Revealing speckle obscured living human retinal cells with artificial intelligence assisted adaptive optics optical coherence tomography." *Communications Medicine* 4.1 (2024): 68.

To use PGAN, the following system prerequisites and software given below are essential.

**System Requirements**

**Prerequisites**

* Windows 10
* NVIDIA GPU +CUDA (tested on NVIDIA TITAN V, CUDA 11.7)

**Installation**

* Install [Anaconda](https://www.anaconda.com/products/distribution)
* In the anaconda prompt:

conda create -n <newenv> python=3.7.13

conda activate <newenv>

cd P-GAN

pip install -r requirements.txt

cd src

A demo test dataset has been deposited in ./data/test\_data. The folder contains eight speckled images which are input to the model (P-GAN) (./data/test\_data/input).

The ground truth images and the P-GAN recovered images are provided in (./data/test\_data/ground-truth) and (./data/test\_data/result), respectively.

The trained model weights are deposited in ./data/trained\_model.

The python files train\_model.py and python test\_model.py contain the TensorFlow implementation of the training and testing pipelines for P-GAN.

**To test the software**

* Run python test\_model.py to test the trained model whose weights are deposited in ./data/trained\_model.
* The results are stored in ./data/test\_data/result.
* The ground truth images are provided in ./data/test\_data/ground-truth.

**To retrain the model on custom data**

Run python train\_model.py \

--path-gt <path to training ground truth images> \

--path-input <path to training input images> \

--img-width <width of training images, default:150> \

--img-height <height of training images, default:150> \

--bs <batch size, default:8> \

--epoch <number of epochs, default:100>