Predicting the progression of ophthalmic disease based on slit-lamp images using a deep temporal sequence network

Jiewei Jiang,Xiyang Liu,Lin Liu,Shuai Wang,Erping Long,Haoqing Yang,Fuqiang Yuan,Deying Yu,Kai Zhang,Liming Wang,Zhenzhen Liu,Dongni Wang,ChangZun Xi,Zhuoling Lin,Xiaohang Wu,Jiangtao Cui,Mingmin Zhu,Haotian Lin

Abstract

Ocular images play an essential role in ophthalmology. Current research mainly focus on computer-aided diagnosis using slit-lamp images, however few studies have been done to predict the progression of ophthalmic disease. Therefore by exploring an effective approach of prediction can help to plan treatment strategies and to provide early warning for the patients. In this study, we present an end-to-end temporal sequence network (TempSeq-Net) to automatically predict the progression of ophthalmic diseases based on consecutive slit-lamp images. First, we comprehensively compare six potential combinations of three convolutional neural networks and long short term memory (or recurrent neural network) in terms of effectiveness and efficiency, to obtain the optimal TempSeq-Net model. Second, we analyze the impacts of sequence lengths on model's performance which help to evaluate their stability and validity and to determine the appropriate range of sequence lengths.

Citation: Jiewei Jiang,Xiyang Liu,Lin Liu,Shuai Wang,Erping Long,Haoqing Yang,Fuqiang Yuan,Deying Yu,Kai Zhang,Liming Wang,Zhenzhen Liu,Dongni Wang,ChangZun Xi,Zhuoling Lin,Xiaohang Wu,Jiangtao Cui,Mingmin Zhu,Haotian Lin Predicting the progression of ophthalmic disease based on slit-lamp images using a deep temporal sequence network. **protocols.io**

dx.doi.org/10.17504/protocols.io.qgzdtx6

Published: 01 Jun 2018

Protocol

Install Prerequisite Libraries

Step 1.

This protocol requires libraries:

- python (https://www.python.org/)
- Caffe - Deep learning framework (http://caffe.berkeleyvision.org/)
- Numpy (http://www.numpy.org/)

Each library can be verified on the property websites.

The main experiment steps are the following: prerequisite libraries, ophthalmic dataset and Temseq-Net code preparation, Temseq-Net models training (using script),
Models evaluation (using python script).

📈 EXPECTED RESULTS

Install prerequisites (Python, Caffe, Numpy)

Ophthalmic dataset and Temseq-Net code preparation

Step 2.

Ophthalmic dataset and Temseq-Net code preparation

The TempSeq-Net prediction system for ophthalmic disease is developed based on the caffe framework. The ophthalmic sequence dataset and Temseq-Net code: https://github.com/Ophthalmology-CAD/TempSeq-Net.

Clone this repository to your folder in the Linux OS (such as Ubuntu 14.04 64bits), let's refer to it as $ROOT

```
git clone https://github.com/Ophthalmology-CAD/TempSeq-Net.git
```

Compile the caffe and pycaffe.

```
cd $ROOT
make all
make test
make runtest
make pycaffe
```

📈 EXPECTED RESULTS

Download the ophthalmic sequence datasets and the TempSeq-Net prediction code from https://github.com/Ophthalmology-CAD/TempSeq-Net. And then compile the TemSeq-Net code.

Download the pre-trained models

Step 3.

Download the pre-trained models

- alexnet model: □https://people.eecs.berkeley.edu/lisa_anne/single_frame_all_layers_hyb_RGB_iter_5000.caffemodel□, put it in $ROOT/examples/LSTM-AlexNet

- googlenet model:□https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet□, put it in $ROOT/examples/LSTM-GoogLeNet

- ResNet-50 model:□https://github.com/KaimingHe/deep-residual-networks#models□, put it in $ROOT/examples/LSTM-ResNet

EXPECTED RESULTS

Download the pre-trained models.

Train the prediction models

Step 4.

Train six potential prediction models for ophthalmic disease

- Run the run_lstm_alexnet_RGB.sh script to train the lstm-alexnet model

```
cd $ROOT/examples/LSTM-AlexNet
sh run_lstm_alexnet_RGB.sh
```

- Run the run_lstm_googlenet_RGB.sh script to train the lstm-googlenet model

```
cd $ROOT/examples/LSTM-GoogLeNet
sh run_lstm_googlenet_RGB.sh
```

- Run the run_lstm_resnet_RGB.sh script to train the lstm-resNet model

```
cd $ROOT/examples/LSTM-ResNet
sh run_lstm_resnet_RGB.sh
```

EXPECTED RESULTS

Train six potential prediction models for ophthalmic disease.

Models evaluation

Step 5.

Evaluate the performance of the six potential prediction models

The test code is in $ROOT/examples/test

- Run 5-6classify_video-alexnet.py to test: in python terminal.

EXPECTED RESULTS

Evaluate the performance of the six potential prediction models.