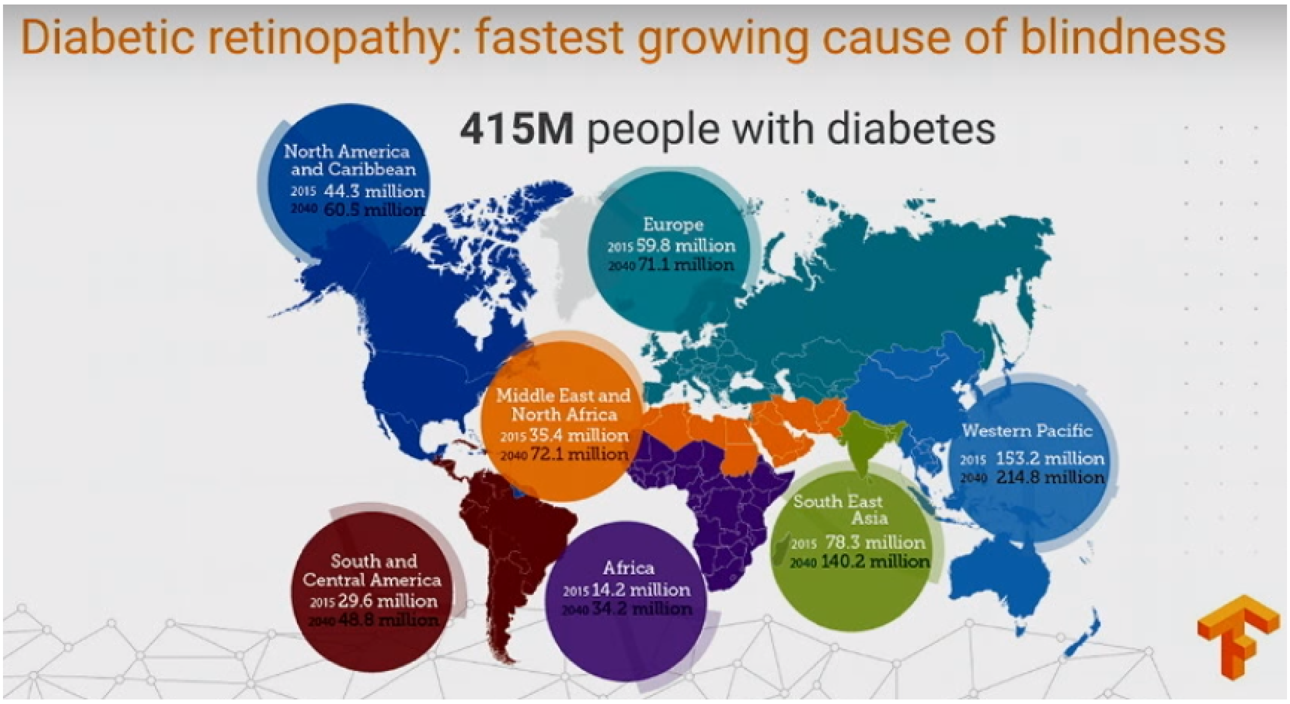
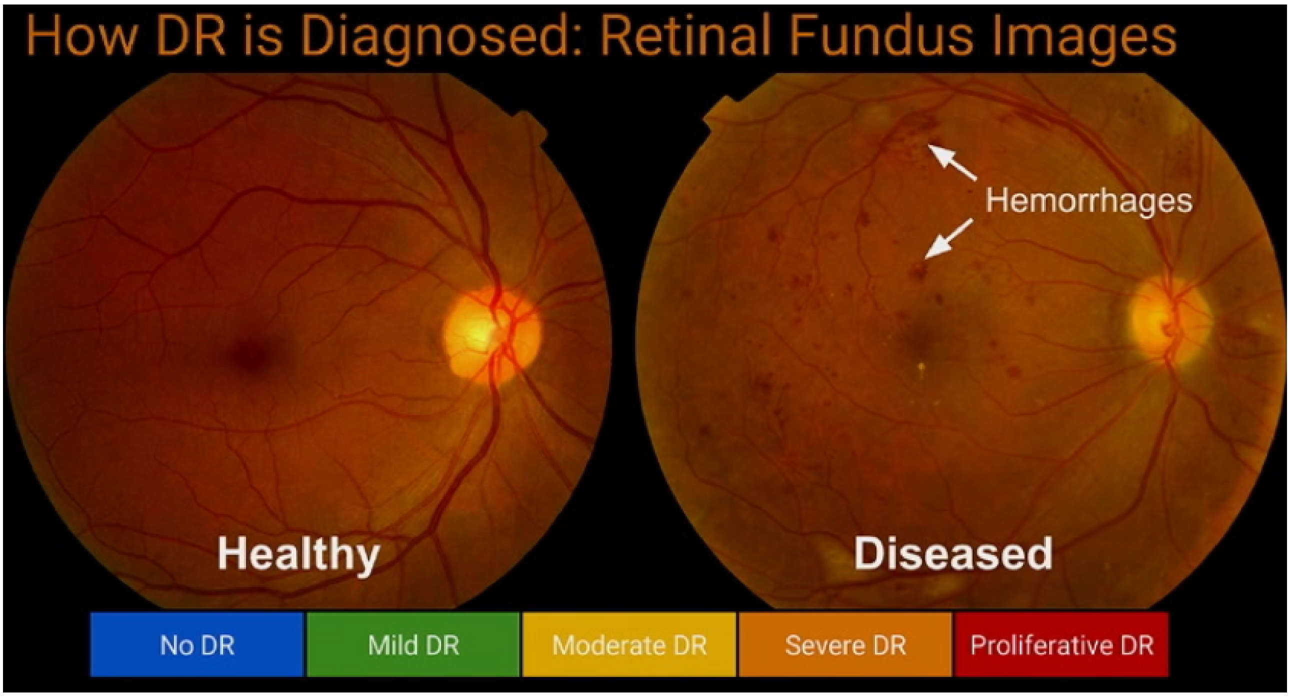
DiabeticRetinopathy Review and Attempt

Goole Approch

Intro to DiabeticRetinopathy

TensorFlow 在医疗领域的应用 - 视网膜图像（2017年TensorFlow开发者大会）




How DR is Diagnosed: Retinal Fundus Images
Hemorrhages
Healthy
Diseased
No DR
Mild DR
Moderate DR
Severe DR
Proliferative DR


Regular screening is key to preventing blindness
=

Pathological symptoms

- 糖尿病损害视网膜主要是由于血糖增高，小血管管壁增厚，渗透性增大，使小血管更易变形和渗漏。糖尿病视网膜病变的严重性和视力下降的程度与血糖水平控制情况以及患糖尿病时间的长短有关。患病时间长短尤为重要，一般患糖尿病至少10年后才出现糖尿病性视网膜病变
- 在非增殖性（单纯型）视网膜病变，视网膜小毛细血管发生破裂和渗漏。在每一膨大的毛细血管破裂之处，形成一有血蛋白沉淀的小囊。医生根据眼底检查可发现这些改变
- 血管渗漏和闭塞。黄斑部水肿或局部缺血是导致视力减退的主要原因。增殖性是由单纯性进一步发展的结果。
- 由于血管变化，毛细血管内皮细胞开始增殖，缺氧的网膜组织释放出血管增殖物质，促使形成新生血管，进而导致出血、机化，而发生增殖性病变，并造成极其严重的不良后果。

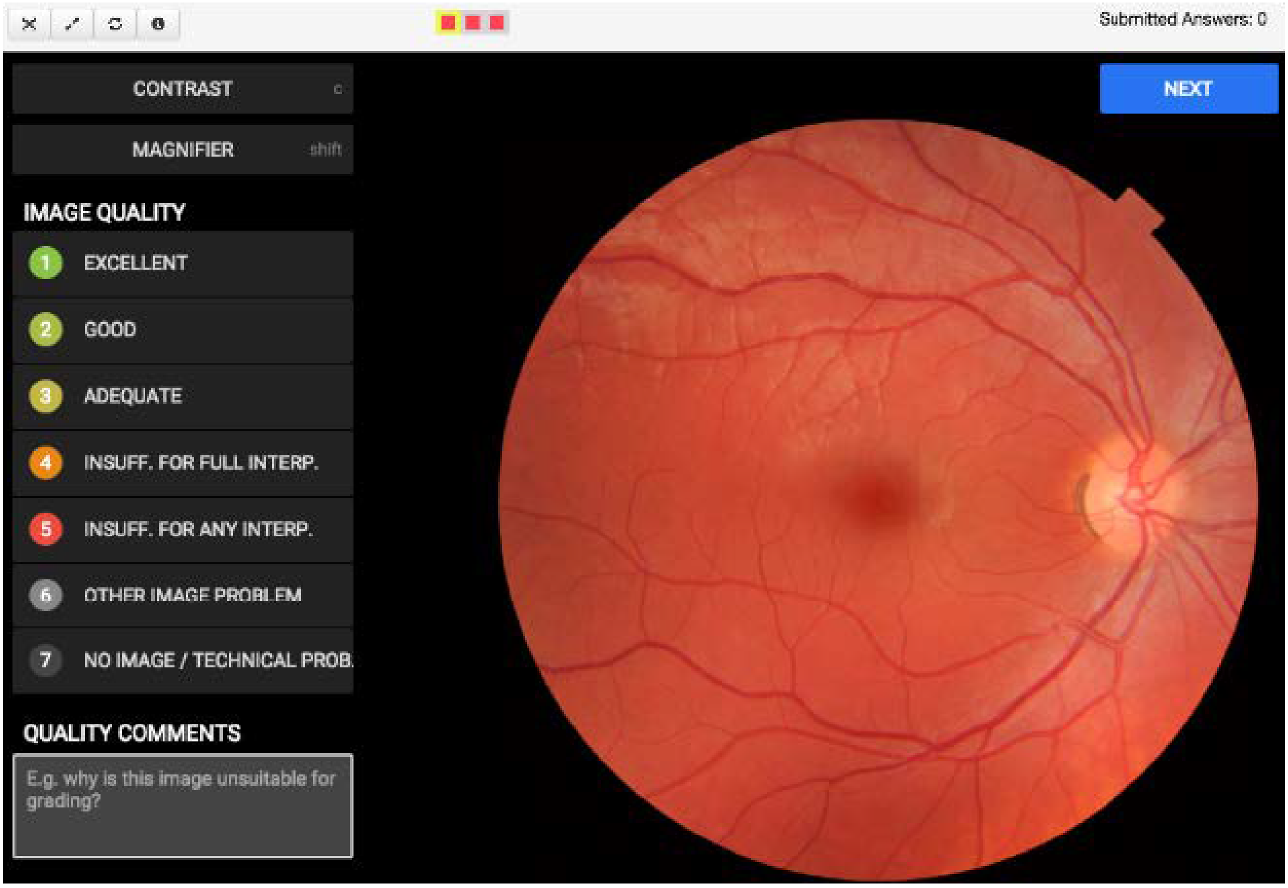
Material

- Macula-centered retinal fundus images were retrospectively obtained from EyePACS in the
United States and 3 eye hospitals in India (Aravind Eye Hospital, Sankara Nethralaya, and Narayana Nethralaya) among
patients presenting for diabetic retinopathy screening. All images were deidentified according to Health Insurance
Portability and Accountability Act Safe Harbor prior to transfer to study investigators. Ethics review and institutional

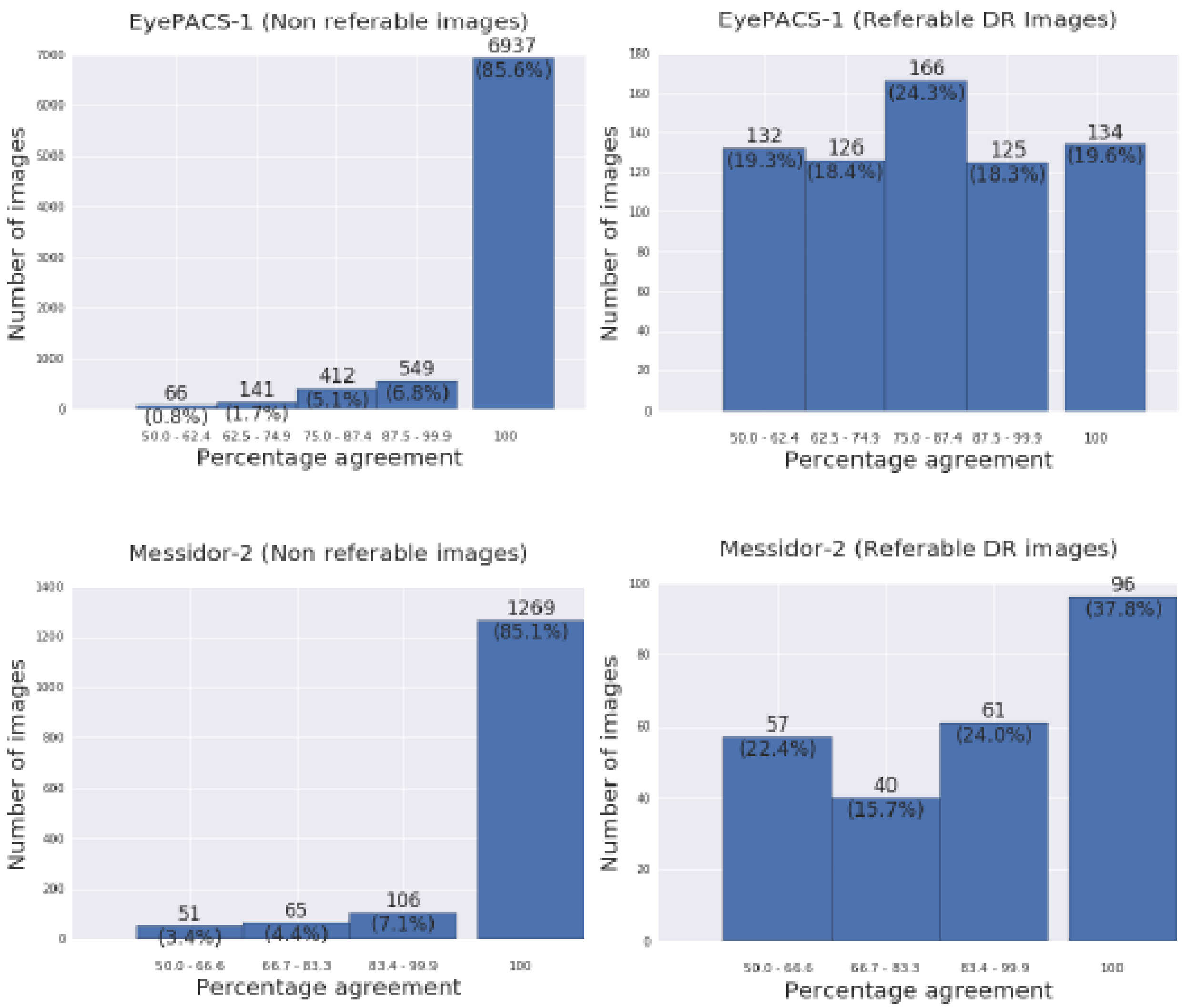
review board exemption was obtained using Quorum Review IRB.

- large data sets of images (n = 128 175) to first “train” an algorithm, then using 2 separate data sets (n = 9963 images and n = 1748 images) to “test” this algorithm

Grade quality control



eFigure 3. Distribution of Agreement Amongst Ophthalmologists on EyePACS-1 (8 Ophthalmologists) and Messidor-2 (7 Ophthalmologists)

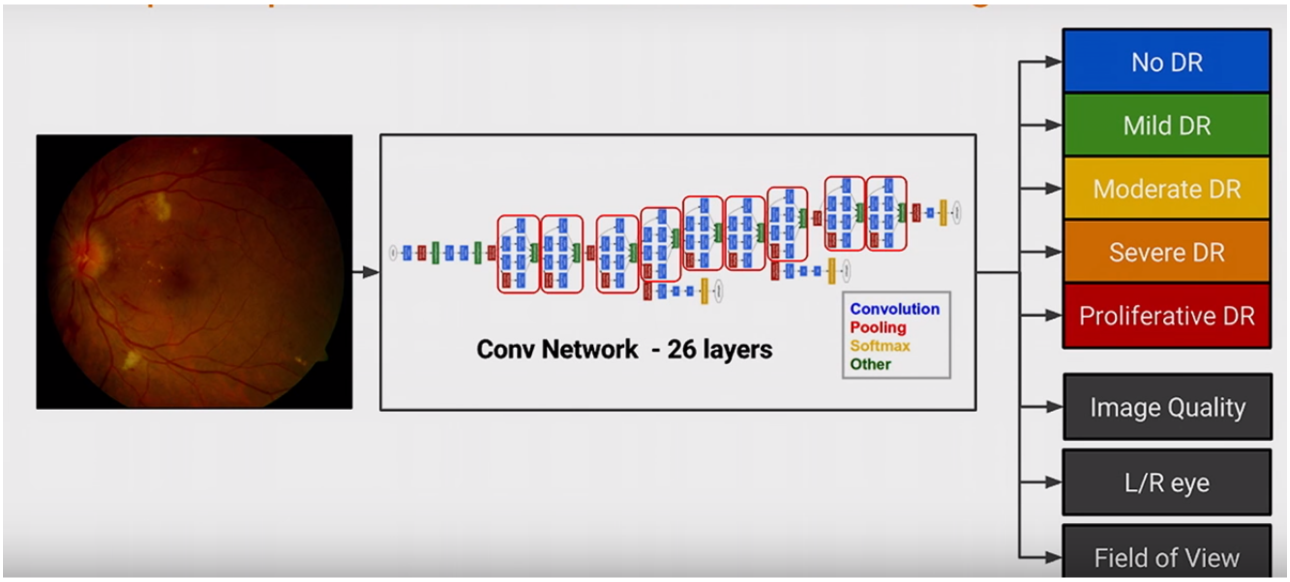


Considered fully gradable by the majority

pre-processing

For algorithm training[1], input images were **scale normalized** by detecting the circular mask of the fundus image and resizing the diameter of the fundus to be 299 pixels wide. Images for which **the circular mask** could not be detected were **excluded** from the development and the clinical validation sets

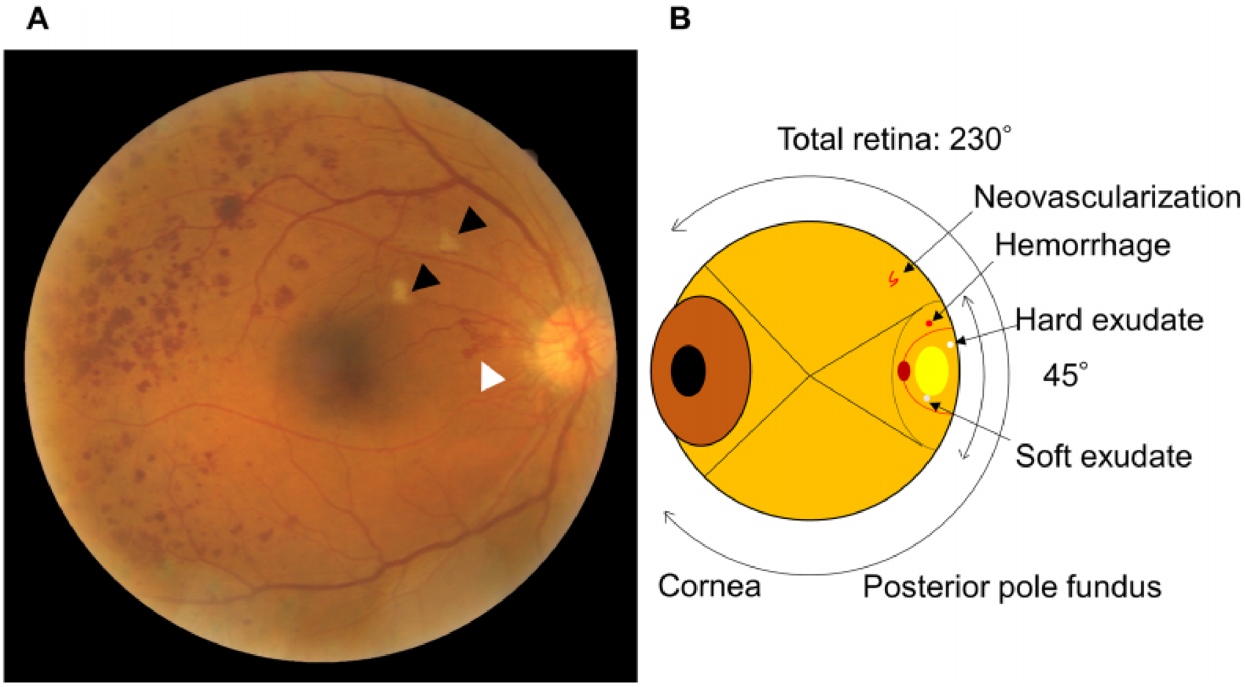
Algorithm



- The optimization algorithm used to train the networkweights was a **distributed stochastic gradient descent** implementation by Dean et al.16 To speed up the training, **batch normalization**[2] as well as **preinitialization** using weights from the same network trained to classify objects
  in the ImageNet data set17 were used. Preinitialization also improved performance
- training: 80% of the data was used to optimize the network weights and (2) tuning: 20% of the data was used to optimize hyperparameters (such as early stopping for training,
  image preprocessing options). An **ensemble** of 10 networks trained on the same data was used, and the final prediction was computed by a linear average over the predictions
  of the ensemble.

Department Ophthalmology, Jichi Medical University

> published: June 22,2017



A fundus photograph and schema of an eye ball

Material

- 9,939 posterior pole photographs from 2,740 patients with diabetes.at Jichi Medical University between May 2011 and June 2015 [3]
- The original photographs were 2,720 × 2,720 pixels. The outlying 88 pixels of the margin were deleted and the photographs shrunken by 50% to 1,272 × 1,272 pixels

Grading

- Modified Davis grading

Grading	
SDR	Microaneurysm, retinal hemorrhage, hard exudate, retinal edema, and more than 3 small soft exudates
PPDR	Soft exudate, varicose veins, intraretinal microvascular abnormality, and non-perfusion area over one disc area
PDR	Neovascularization, pre-retinal hemorrhage, vitreous hemorrhage, fibrovascular proliferative membrane, and tractional retinal detachment

SDR, simple diabetic retinopathy; PPDR, pre-proliferative diabetic retinopathy; PDR, proliferative diabetic retinopathy.

- Grading by actual prognosis.

Grading	Needed treatments	When	Prognosis	No. of 4709	NDR	SDR	PPDR	PDR
0	None	Next visit	All	4445	2479	1289	333	344
1	DME treatments			12	0	5	1	6
2	PRP		Improve	20	1	2	14	3
3			Stable	6	1	1	0	4
4			Worsen	0	0	0	0	0
5	Vitrectomy		Improve	2	0	0	0	2
6			Stable	4	0	0	0	4
7			Worsen	2	0	0	0	2
8	DME treatments	Current visit	All	16	0	0	7	9
9	PRP		Improve	108	0	0	40	68
10			Stable	31	0	1	5	25
11			Worsen	29	0	0	9	20
12	Vitrectomy		Improve	10	3	0	0	7
13			Stable	16	0	0	1	15
14			Worsen	8	0	0	0	8

Model

- Modified GoogLeNet ResNet Used
- 4 GPU Used Simulteneously si(12 GB, GeForce GTX TITAN X; NVIDIA)
- Open framework for deep learning (Caffe, Berkeley Vision and Learning Center, Berkeley, CA, USA).

ICLR2018 Interpretable Computer Aided Diagnosis of Diabetic Retinopathy

> published:10/20/2017

Material

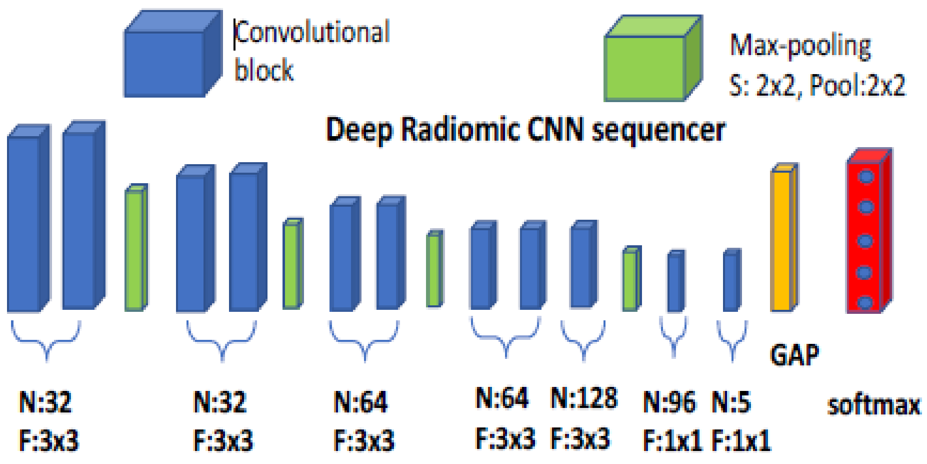
Kaggle diabetic retinopathy dataset five grades of diabetic retinopathy are as follows: 0: Negative, 1: Mild, 2: Moderate, 3: Severe, and 4: Proliferative. [4]

Approach

- Select retinal fundus images for one eye (right) only and performed an automatic selective cropping to remove the background information. The use of a single
  eye led to 53,354 images in total.
- Divide the dataset into 90% and 10% of the dataset for training and testing respectively.
- Performe horizontal and vertical flipping along with channel-wise normalization for the whole dataset as data augmentation

Model

- Deep Radiomic CNN sequencer



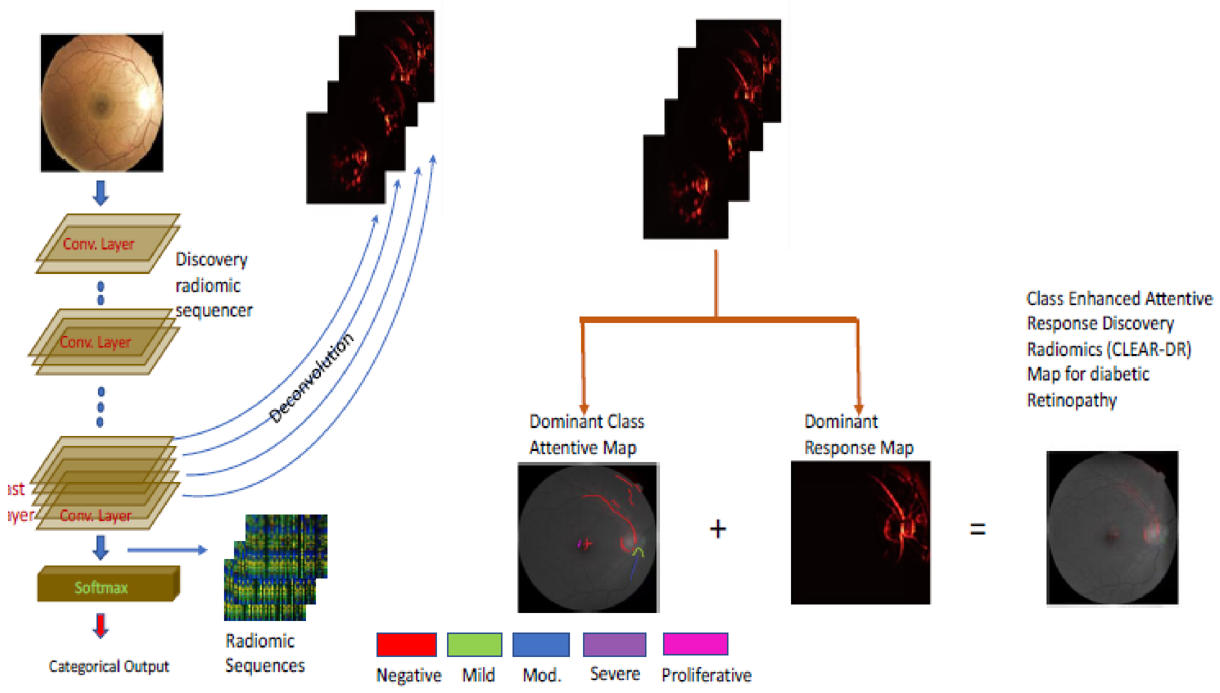
Accuracy

73.2% overall

HighLight

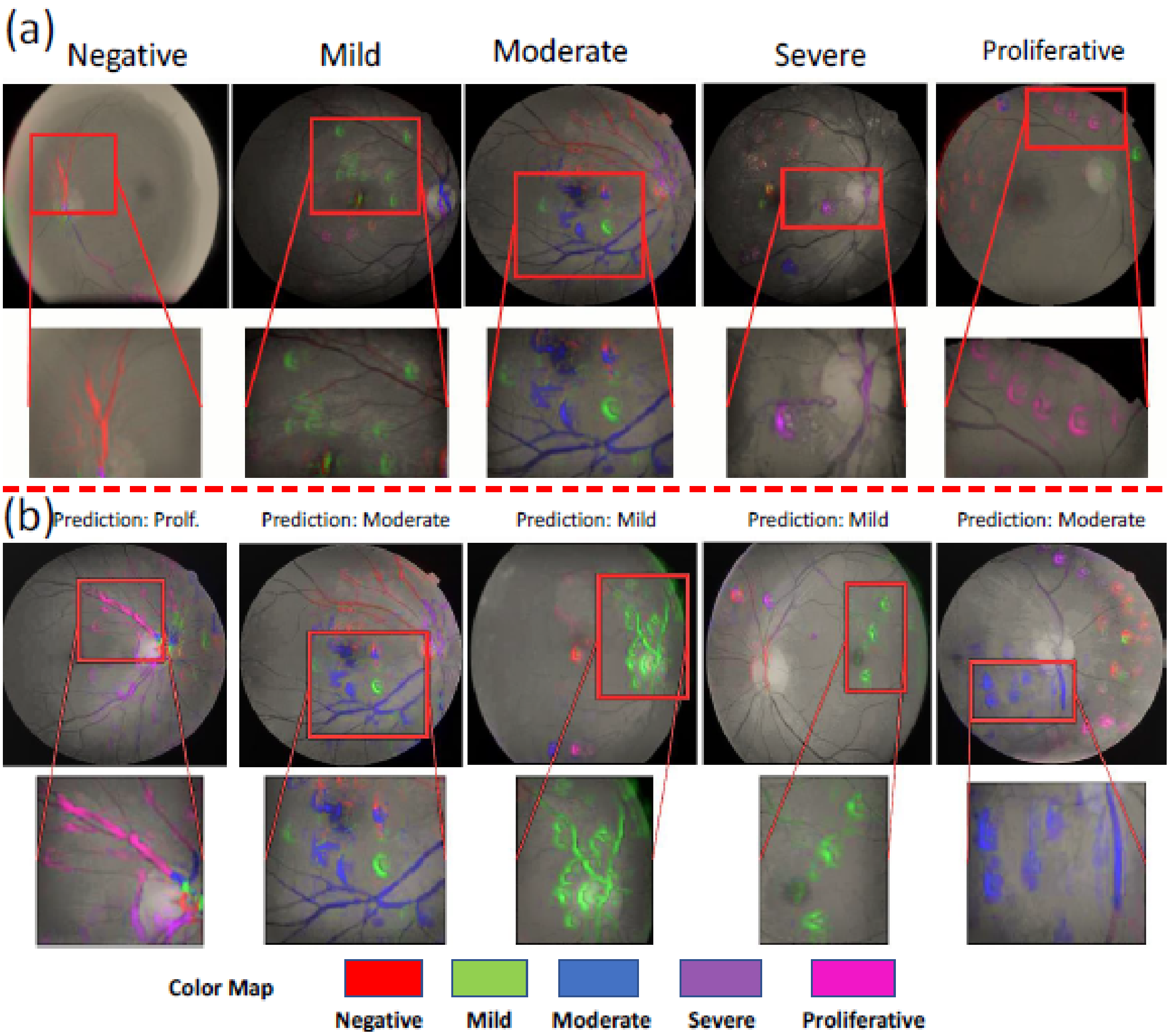
Not only generates discriminating radiomic sequences for diabetic retinopathy grading **but also** provides a mechanism to **visually interpret its decision making process**
.* individual attentive response maps are computed for each kernel associated with a grade by back-projecting activations from the output layer

- Compute two different types of maps
  a. A dominant attentive response map, which shows the dominant attentive level for each location in the image;
  b. A domainant grade involed in the desicion-making progress at each location
  c. the dominant attentive response map and the dominant attentive grade map are combined visually by using color and intensity to produce the final CLEAR-DR map



Experiment

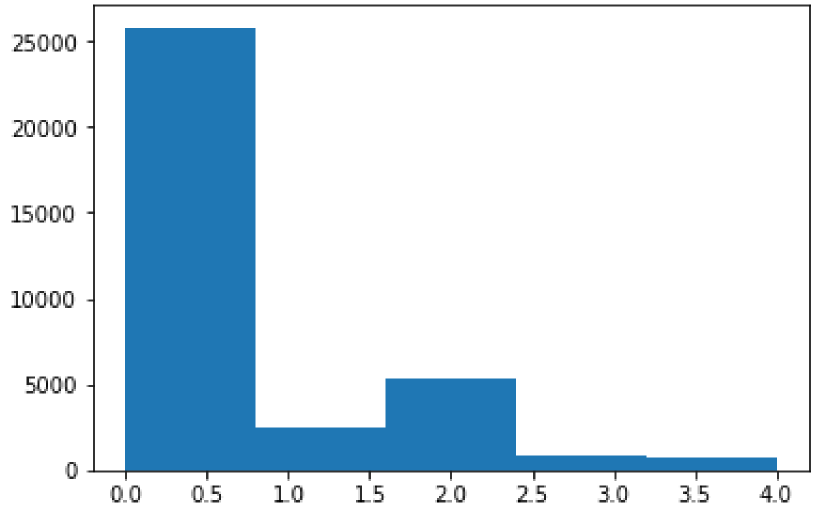
Correctly (a) and **Mis-classified** (b) examples for all diabetic retinopathy grades. Each color represents a single grade, as identified by the color map at the bottom of the figure. As well, the red box indicates the most attentive region used for grade prediction. It can be observed that the attentive regions used by the deep radiomic sequencer for **making correct decisions corresponds to medically relevant landmarks**, thus providing additional evidence for the proposed prediction. Best viewed in color and zoomed in.



My Approach For DiabeticRetinopathy Detection Problem

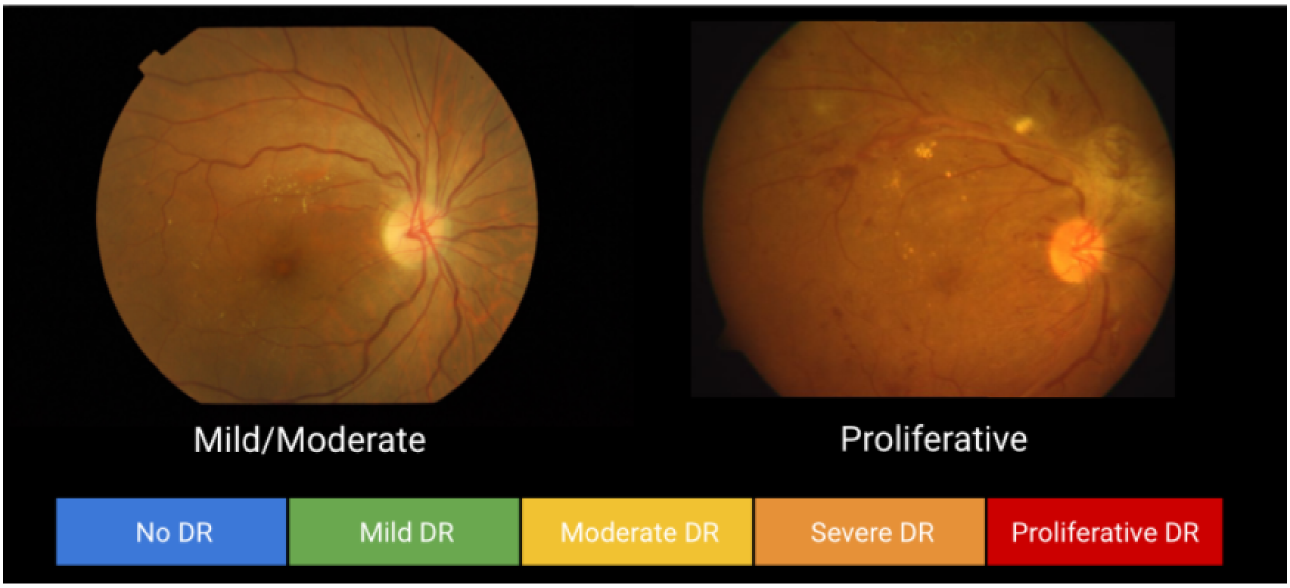
Material

- The data set comes from **Kaggle** which is sponsored by the **California Healthcare Foundation** and Retinal images were provided **by EyePACS**, a free platform for retinopathy screening.
- it consists of high-resolution retinal fundus images with varying degrees of illumination conditions captured using different types of cameras. The retinal fundus images in the dataset were clinically annotated with five different grades related to the presence of diabetic retinopathy.
- fairly unbalanced classed



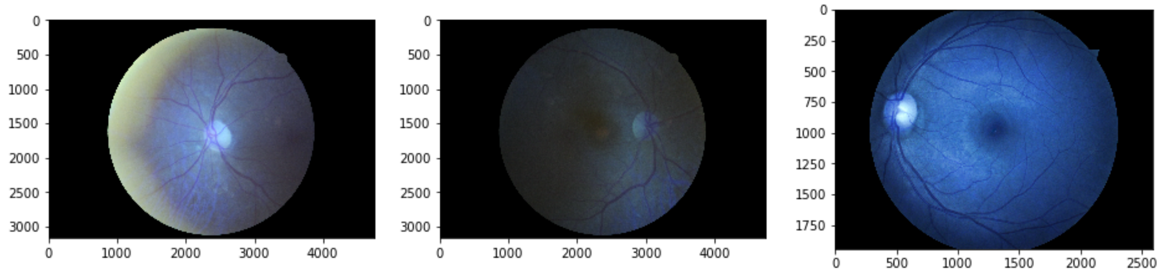
Class	Name	Number of images	Percentage
0	Normal	25810	73.48%
1	Mild NPDR	2443	6.96%
2	Moderate NPDR	5292	15.07%
3	Severe NPDR	873	2.48%
4	PDR	708	2.01%

Grading

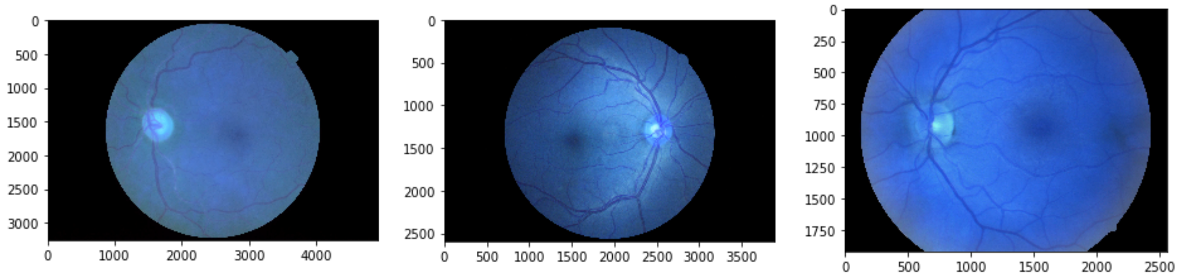


Exploration

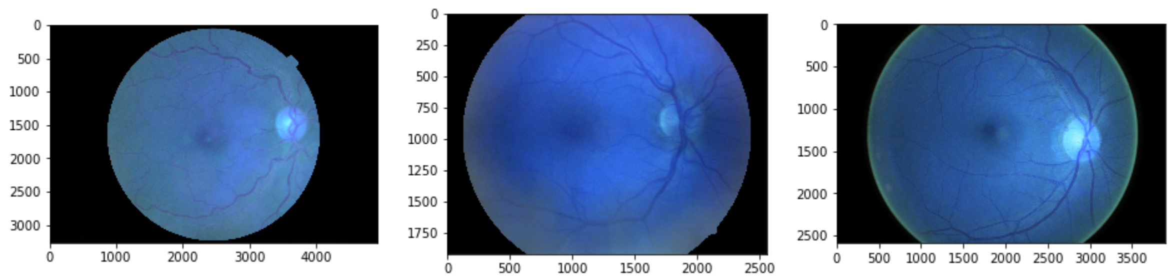
- No DR



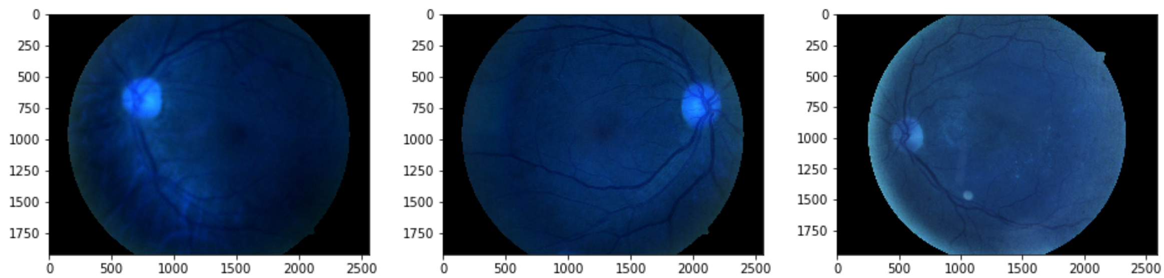
- Mild DR



- Moderate DR



- Severe DR



- Proliferative DR

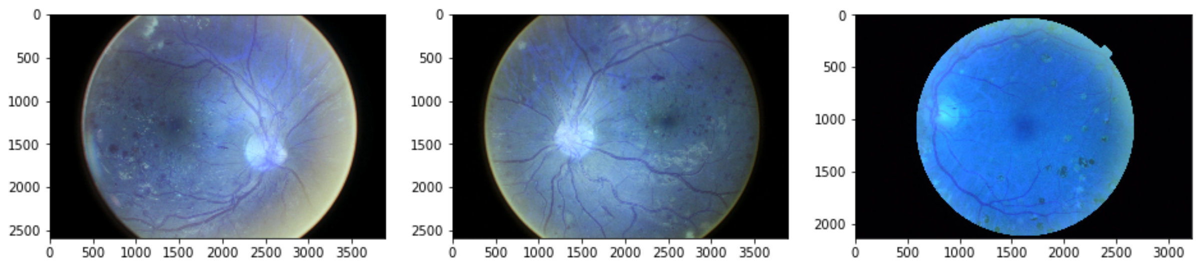
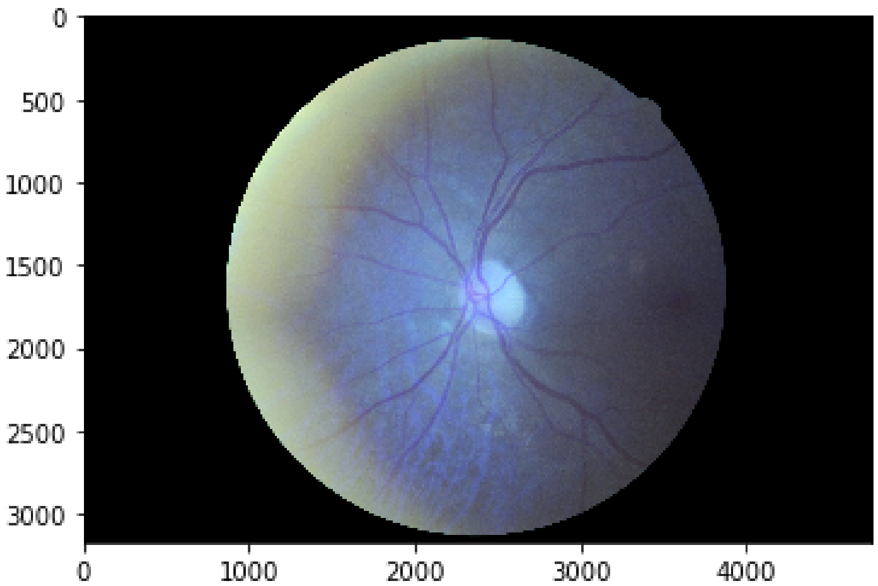


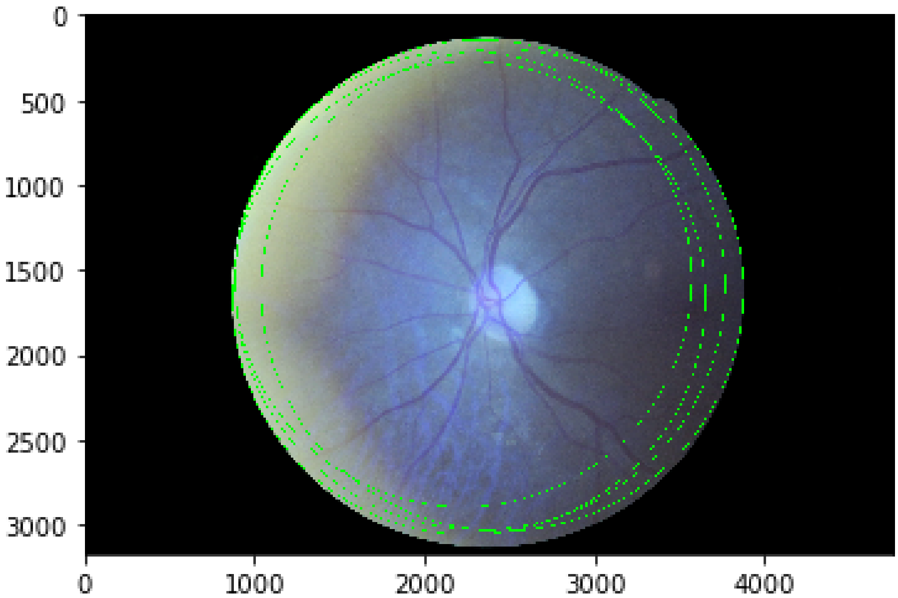
Image Preprocessing

- original image

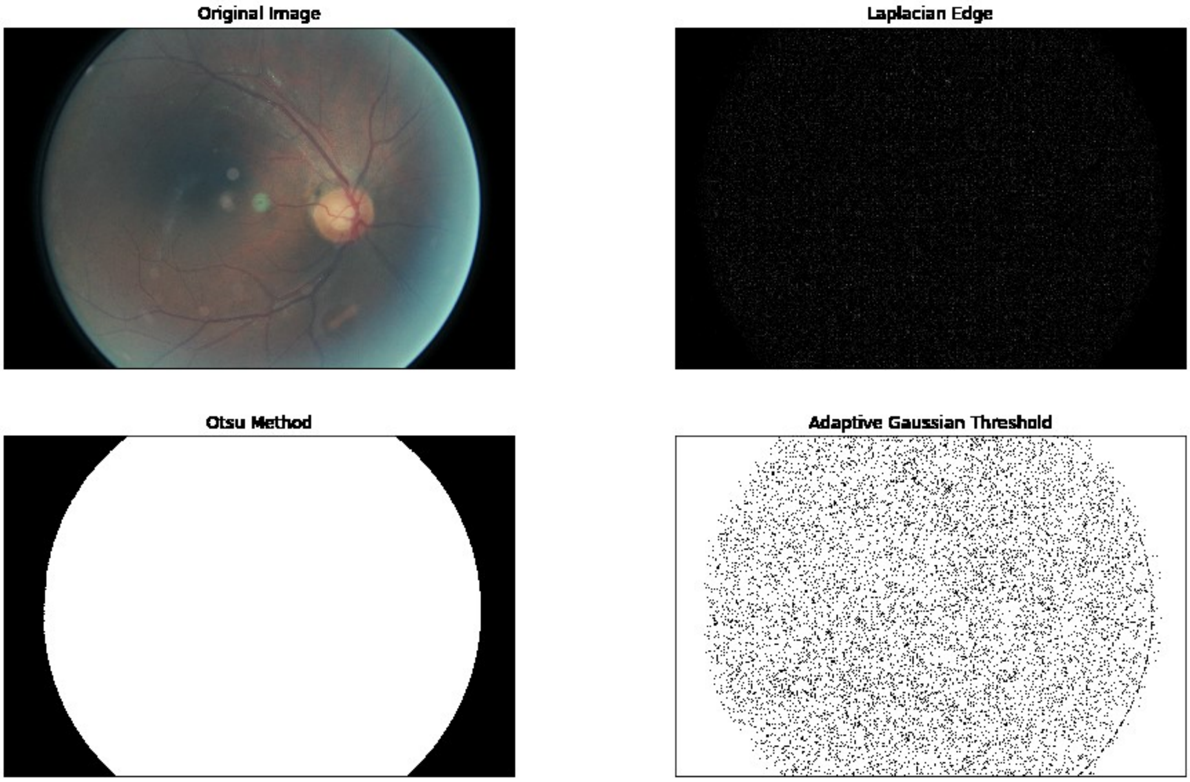


crop the extra black edge

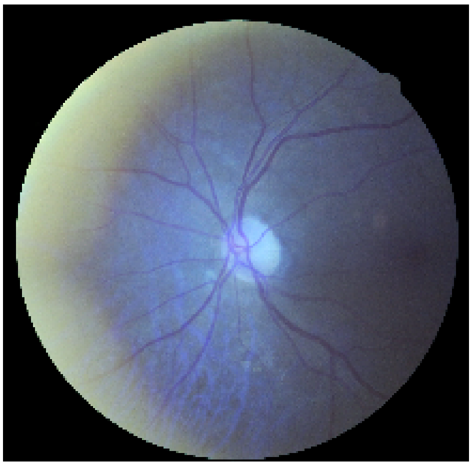
1. detect the circle and crop



2. Use filter to detect the edge (implement opencv library)

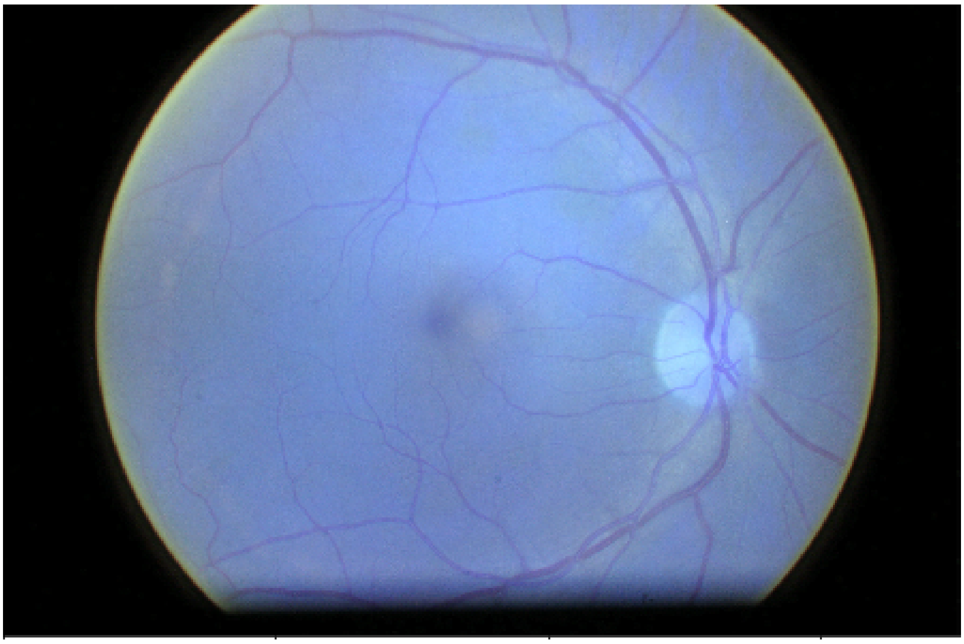


3. Finaly It looks like this below

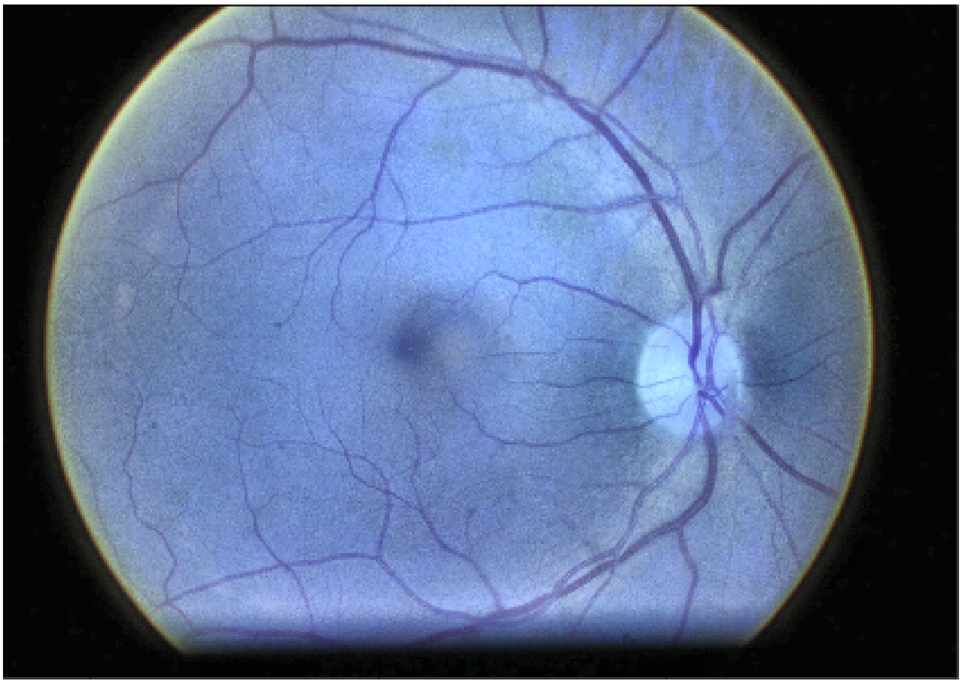


Enhance the image

- before



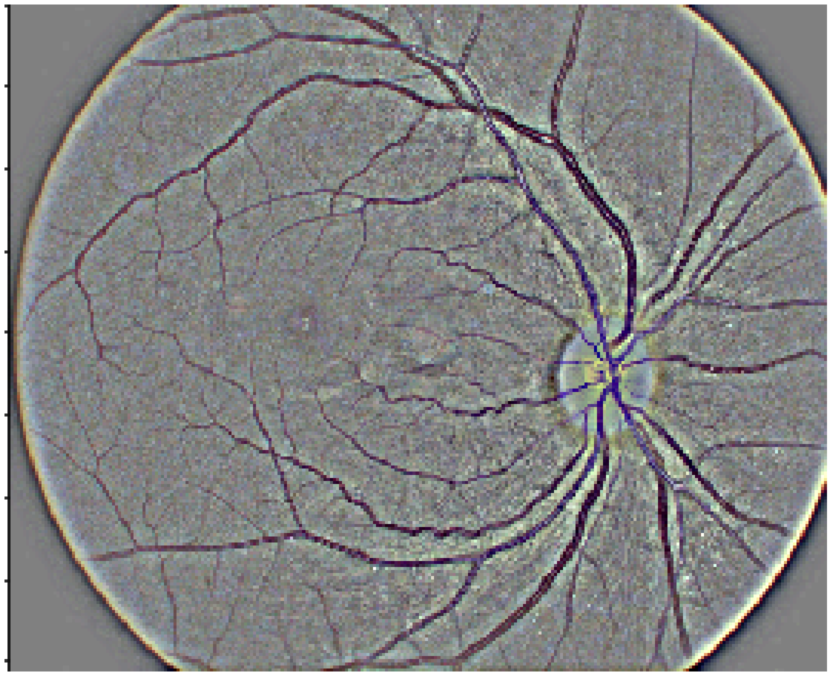
- After Enhance



> opencv createCLAH method implemented

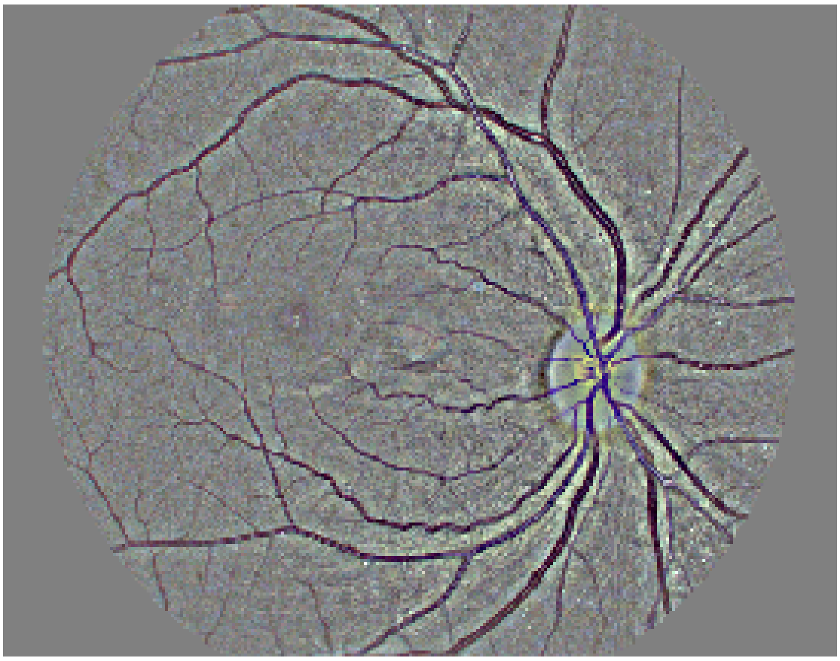
Resize to (299,299) (512,512)

Subtract local mean color

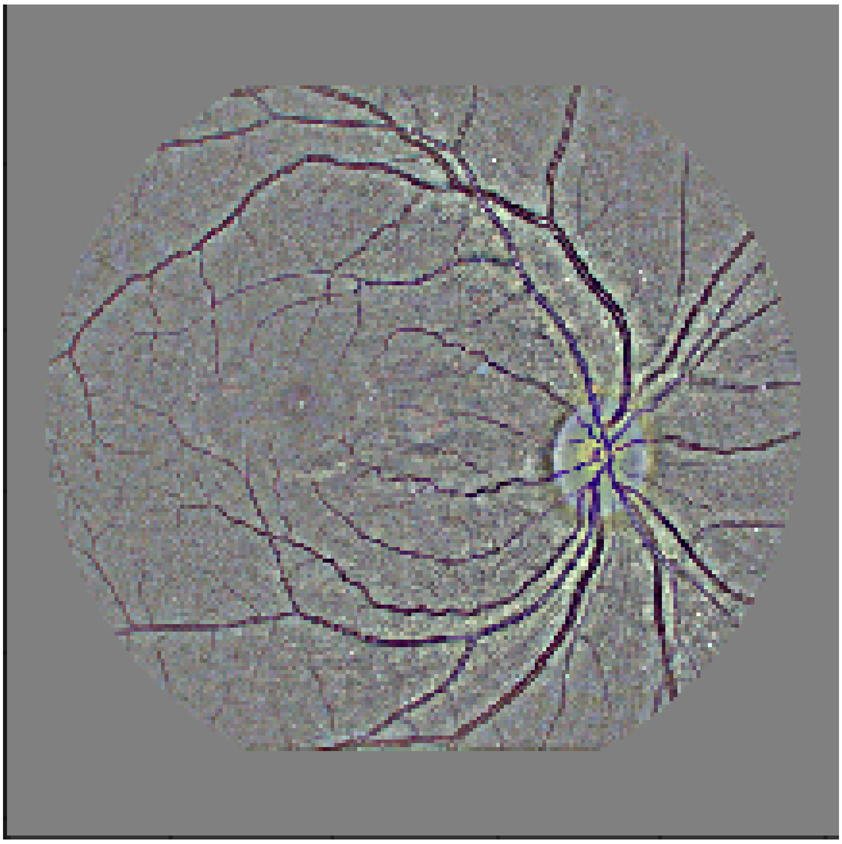


```
def subtract_gaussian_blur(img):
    # http://docs.opencv.org/trunk/d0/d86/tutorial_py_image_arithmetics.html
    # http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html
    gb_img = cv2.GaussianBlur(img, (0, 0), 5)

    return cv2.addWeighted(img, 4, gb_img, -4, 128)

img_gbs = subtract_gaussian_blur(img_crop.copy())
plt.imshow(img_gbs)
```

Remove 10% of the outer circle



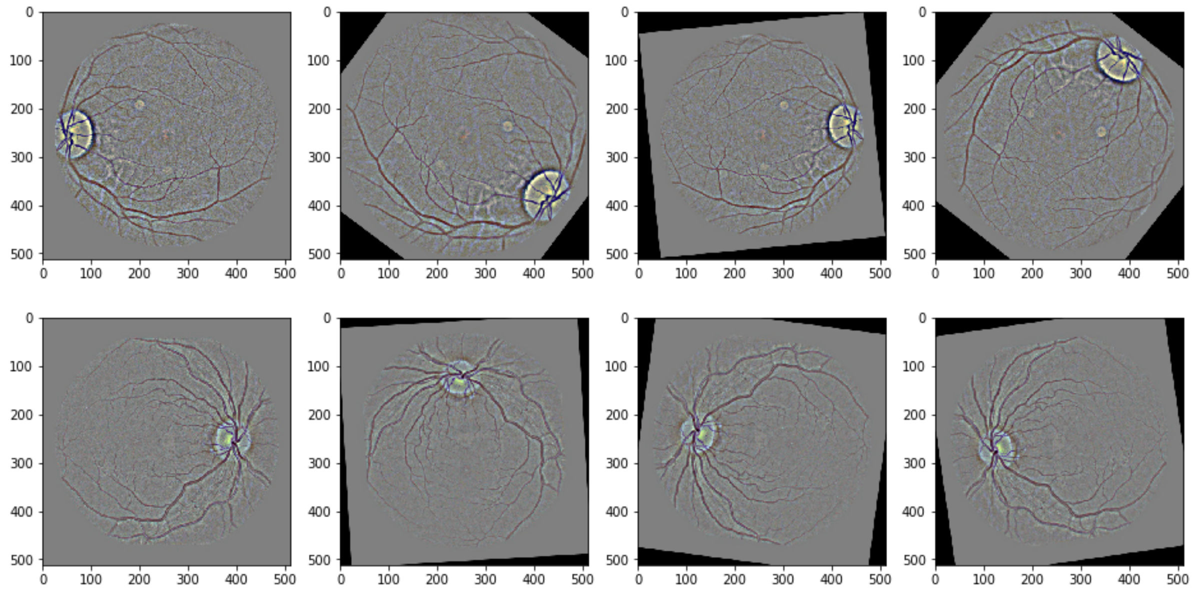
Place the processed image in the center of image

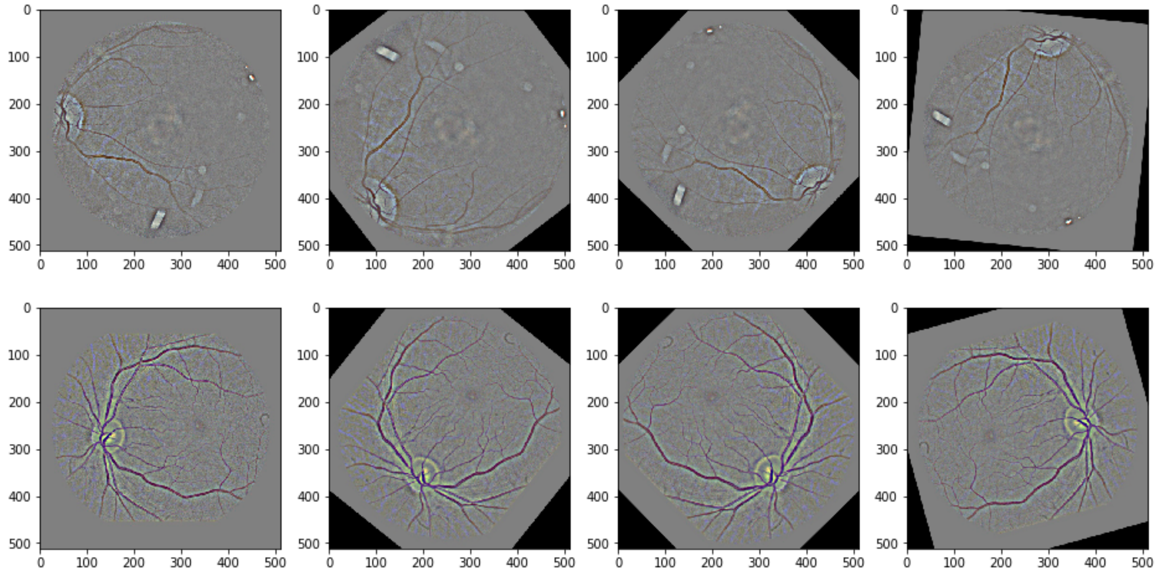


Abandon the Image with std lower than 16(too dark or too bright)

DataAgumentation

- Flip
- Sclae
- Rotate



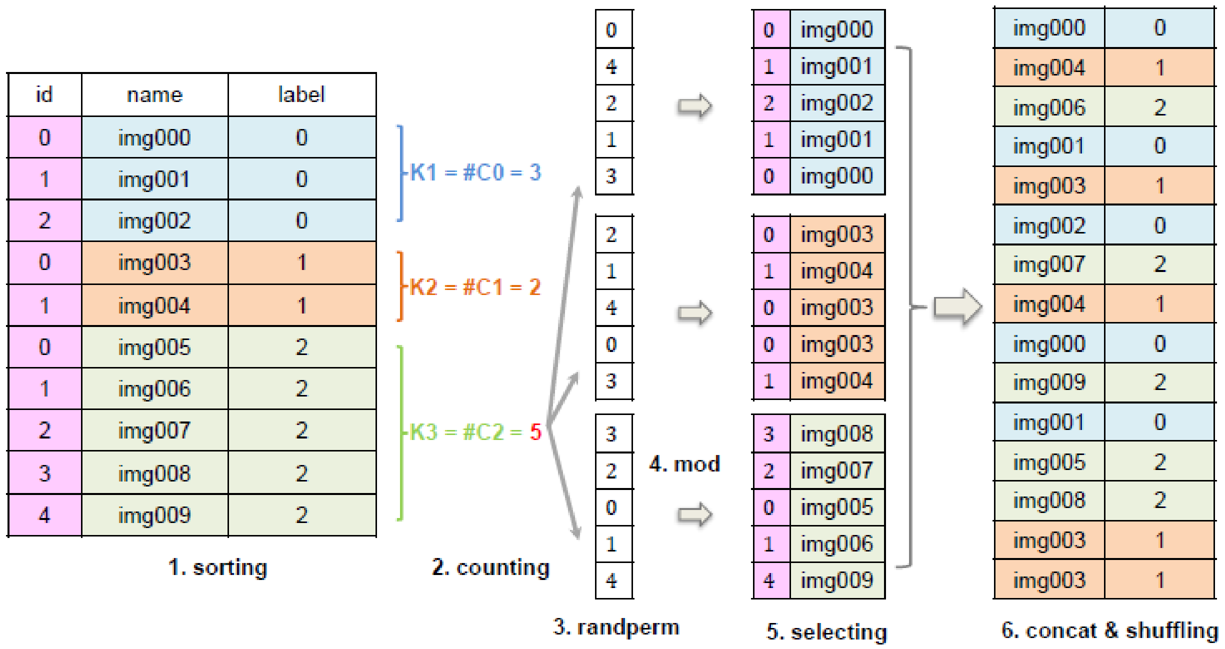


BatchNormalization

Preprocessing Parralelismly Using MultiProcessor

```
top - 14:38:40 up 3 days, 16:54,  9 users,  load average: 6.57, 2.72, 1.62
Tasks: 825 total,  38 running, 787 sleeping,   0 stopped,   0 zombie
%Cpu(s): 87.3 us,  3.1 sy,  2.4 ni,  7.0 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 13175270+total, 43812444 free, 26394320 used, 61545944 buff/cache
KiB Swap:  4194300 total,  4194300 free,        0 used. 10319016+avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 85452 zhangst   20   0 3419220 111588   4176 R 101.0  0.1   0:04.71 python
 85422 zhangst   20   0 3400788  73096   4192 R 100.7  0.1   0:04.72 python
 85435 zhangst   20   0 3415372  87868   4200 R 100.7  0.1   0:04.71 python
 85443 zhangst   20   0 3417920 110736   4196 R 100.7  0.1   0:04.69 python
 85419 zhangst   20   0 3400788  68756   4176 R 100.3  0.1   0:04.70 python
 85421 zhangst   20   0 3419224 111764   4180 R 100.3  0.1   0:04.71 python
 85418 zhangst   20   0 3419224 111456   4196 R 100.0  0.1   0:04.70 python
 85432 zhangst   20   0 3419220 111752   4184 R 100.0  0.1   0:04.68 python
 85433 zhangst   20   0 3400788  77748   4192 R 100.0  0.1   0:04.68 python
 85438 zhangst   20   0 3406508  98752   4196 R 100.0  0.1   0:04.70 python
 85440 zhangst   20   0 3400788  63696   4196 R 100.0  0.0   0:04.70 python
 85450 zhangst   20   0 3371260  66284   4180 R 100.0  0.1   0:04.68 python
 85420 zhangst   20   0 3400788  92684   4196 R  99.7  0.1   0:04.68 python
 85424 zhangst   20   0 3395244  85808   4192 R  99.7  0.1   0:04.66 python
 85425 zhangst   20   0 3400788  92532   4196 R  99.7  0.1   0:04.69 python
 85446 zhangst   20   0 3379696  72124   4196 R  99.7  0.1   0:04.69 python
 85447 zhangst   20   0 3402284  94708   4196 R  99.7  0.1   0:04.65 python
 85448 zhangst   20   0 3419224 111004   4176 R  99.7  0.1   0:04.66 python
 85449 zhangst   20   0 3419224 112400   4196 R  99.7  0.1   0:04.67 python
 85423 zhangst   20   0 3467920 128176   4196 R  99.3  0.1   0:04.68 python
 85428 zhangst   20   0 3444172 132948   4184 R  99.3  0.1   0:04.67 python
 85429 zhangst   20   0 3419224 111748   4188 R  99.3  0.1   0:04.67 python
 85437 zhangst   20   0 3400788  71380   4196 R  99.3  0.1   0:04.68 python
 85444 zhangst   20   0 3419224 110964   4196 R  99.3  0.1   0:04.64 python
 85431 zhangst   20   0 3400788  63976   4196 R  99.0  0.0   0:04.65 python
```

Balanced Sampling via Label Shuffing



Label Smooth

Model Architeche

Modified InceptionV3

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 299, 299, 3)	0
batch_normalization_95 (Batc	(None, 299, 299, 3)	12
inception_v3 (Model)	(None, 8, 8, 2048)	21802784
global_average_pooling2d_1 (	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 5)	10245

Total params: 21,813,041
Trainable params: 21,778,603
Non-trainable params: 34,438

Modified Xception

Layer (type)	Output Shape	Param #
input_8 (InputLayer)	(None, 299, 299, 3)	0
batch_normalization_104 (Bat	(None, 299, 299, 3)	12
xception (Model)	(None, 10, 10, 2048)	20861480
global_average_pooling2d_2 (	(None, 2048)	0
dropout_2 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 5)	10245

Total params: 20,871,737
Trainable params: 20,817,203
Non-trainable params: 54,534

Basic CNN model

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 510, 510, 32)	896
conv2d_2 (Conv2D)	(None, 508, 508, 32)	9248
max_pooling2d_1 (MaxPooling2	(None, 254, 254, 32)	0
conv2d_3 (Conv2D)	(None, 252, 252, 64)	18496
conv2d_4 (Conv2D)	(None, 250, 250, 64)	36928
max_pooling2d_2 (MaxPooling2	(None, 125, 125, 64)	0
conv2d_5 (Conv2D)	(None, 123, 123, 96)	55392
conv2d_6 (Conv2D)	(None, 121, 121, 96)	83040
max_pooling2d_3 (MaxPooling2	(None, 60, 60, 96)	0
conv2d_7 (Conv2D)	(None, 58, 58, 128)	110720
conv2d_8 (Conv2D)	(None, 56, 56, 128)	147584
max_pooling2d_4 (MaxPooling2	(None, 28, 28, 128)	0
conv2d_9 (Conv2D)	(None, 26, 26, 192)	221376
conv2d_10 (Conv2D)	(None, 24, 24, 192)	331968

conv2d_6 (Conv2D)	(None, 121, 121, 96)	83040
max_pooling2d_3 (MaxPooling2	(None, 60, 60, 96)	0
conv2d_7 (Conv2D)	(None, 58, 58, 128)	110720
conv2d_8 (Conv2D)	(None, 56, 56, 128)	147584
max_pooling2d_4 (MaxPooling2	(None, 28, 28, 128)	0
conv2d_9 (Conv2D)	(None, 26, 26, 192)	221376
conv2d_10 (Conv2D)	(None, 24, 24, 192)	331968
max_pooling2d_5 (MaxPooling2	(None, 12, 12, 192)	0
conv2d_11 (Conv2D)	(None, 10, 10, 256)	442624
conv2d_12 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_6 (MaxPooling2	(None, 4, 4, 256)	0
conv2d_13 (Conv2D)	(None, 2, 2, 256)	590080
max_pooling2d_7 (MaxPooling2	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
dense_2 (Dense)	(None, 5)	1285

```
Total params: 2,705,509
Trainable params: 2,705,509
Non-trainable params: 0
```

Ensemble Of Them

Training Environment

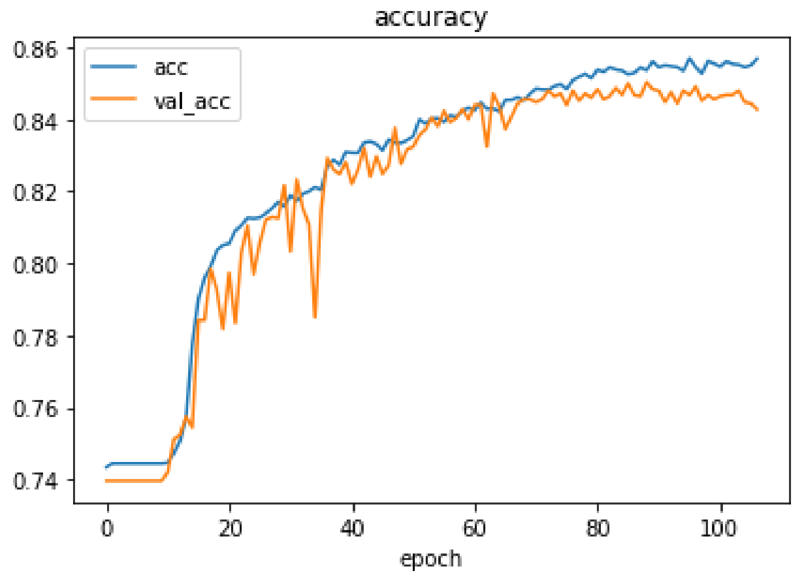
- CUDA8.0 CuDnn5.1
- Open framework for deep learning (Tensorflow-Gpu ,Keras)
- 8 GPUS (12 GB, GeForce GTX TITAN X; NVIDIA) Training Parralelismly

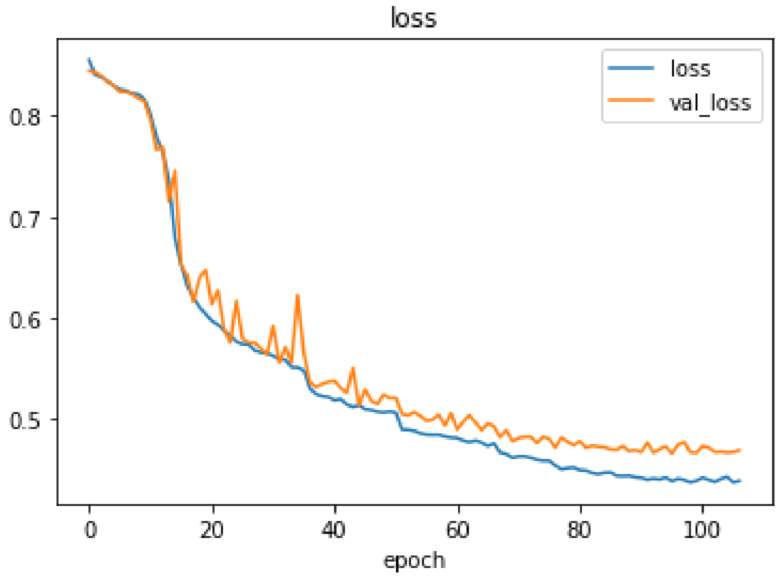
```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 381.22                 Driver Version: 381.22                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  TITAN Xp            Off  | 0000:04:00.0     Off |                  N/A |
| 23%   45C    P2    80W / 250W |  12019MiB / 12189MiB |     46%      Default |
+-------------------------------+----------------------+----------------------+
|   1  TITAN Xp            Off  | 0000:05:00.0     Off |                  N/A |
| 23%   34C    P2    60W / 250W |  11713MiB / 12189MiB |     16%      Default |
+-------------------------------+----------------------+----------------------+
|   2  TITAN Xp            Off  | 0000:08:00.0     Off |                  N/A |
| 23%   40C    P2    60W / 250W |  11713MiB / 12189MiB |     25%      Default |
+-------------------------------+----------------------+----------------------+
|   3  TITAN Xp            Off  | 0000:09:00.0     Off |                  N/A |
| 23%   36C    P2    61W / 250W |  11713MiB / 12189MiB |     18%      Default |
+-------------------------------+----------------------+----------------------+
|   4  TITAN Xp            Off  | 0000:84:00.0     Off |                  N/A |
| 23%   40C    P2    59W / 250W |  11713MiB / 12189MiB |      9%      Default |
+-------------------------------+----------------------+----------------------+
|   5  TITAN Xp            Off  | 0000:85:00.0     Off |                  N/A |
| 23%   34C    P2    60W / 250W |  11713MiB / 12189MiB |      3%      Default |
+-------------------------------+----------------------+----------------------+
|   6  TITAN Xp            Off  | 0000:88:00.0     Off |                  N/A |
| 23%   37C    P2    60W / 250W |  11713MiB / 12189MiB |     21%      Default |
+-------------------------------+----------------------+----------------------+
|   7  TITAN Xp            Off  | 0000:89:00.0     Off |                  N/A |
| 23%   35C    P2    60W / 250W |  11713MiB / 12189MiB |      4%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
```

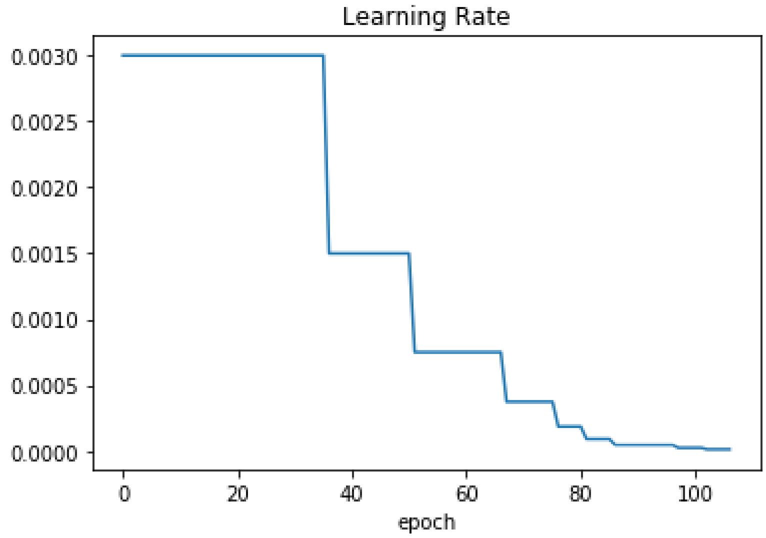
Acuracy

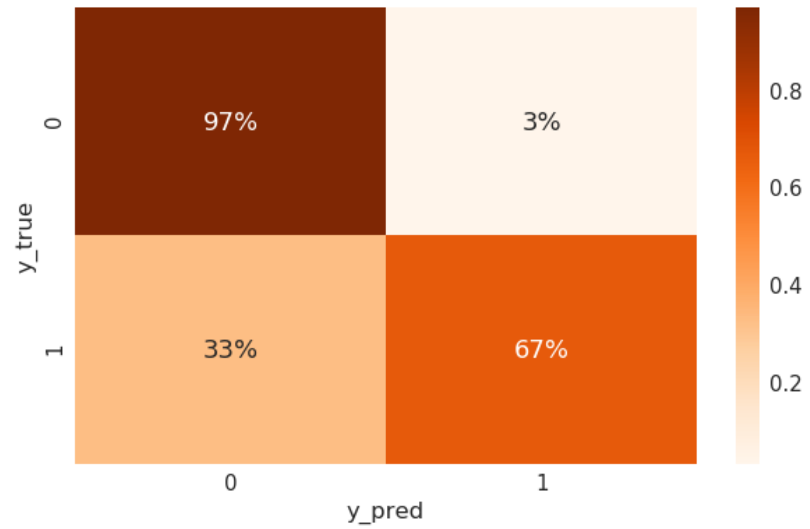
- Divide the dataset into 90% and 10% of the dataset for training and testing respectively.
  85.5% overall for 5 classification (0,1,2,3,4,5)
  90.1% overall for 2 classification(0 vs 1,2,3,4)

PartResult


accuracy
acc
val_acc
0.86
0.84
0.82
0.80
0.78
0.76
0.74
0
20
40
60
80
100
epoch


loss
loss
val_loss
0.8
0.7
0.6
0.5
0
20
40
60
80
100
epoch


Learning Rate
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
0
20
40
60
80
100
epoch



Bad Attempts

Too many to say

Summary

1. The model can get Top 10 result when resubmiting in Kaggle
2. There also leaves somewhere to imporve but the time is limited
3. Many idea comes from papers and imageClassification matches
4. **Visually interpreting** its decision making process and **Deconvolution of the cnn** to help Doctors to Find the new feautres to judge the DiabetcRetinopathy is **Promising** rather than the higher Accuracy.

To be Continued…

---

1. Artificial IntelligenceWith Deep Learning Technology Looks Into Diabetic Retinopathy Screening ↩
2. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift ↩
3. Applying artificial intelligence to disease staging: Deep learning for improved staging of diabetic retinopathy ↩
4. Discovery Radiomics with CLEAR-DR: Interpretable Computer Aided Diagnosis of Diabetic Retinopathy ↩