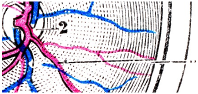
Diabetic Retinopathy Detection

Sergey Ovcharenko [1]   Rasim Akhunzyanov [2]

[1]Deep Learning Engineer at NTech Lab

[2]Computer Vision Research Engineer at LG Electronics



November 4, 2015

Table of contents

Data

Goal: Identify signs of diabetic retinopathy in eye images
Given: 35126 images for training, 53576 images in test set
Images are big: 2500x2000 and larger
Compressed data size: 88Gb

Normal	Mild	Moderate	Severe	Proliferative
25810 73.48%	2443 6.96%	5292 15.07%	873 2.48%	708 2.01%

Quality metric I: Cohen's kappa

Cohen's kappa measures the agreement between two raters A and B who each classify N items into C mutually exclusive categories.

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

where

- p_o is the relative observed agreement among raters
- p_e is the hypothetical probability of chance agreement

$$\kappa = \frac{Pr[A = B] - Pr[A = B | A \text{ and } B \text{ independent}]}{1 - Pr[A = B | A \text{ and } B \text{ independent}]}$$

If the raters are in complete agreement then $\kappa = 1$.

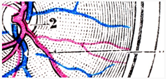
Quality metric I: Cohen's kappa (simple example)

		A			
		1	2	3	Total
B	1	55	5	5	65
	2	15	45	15	75
	3	5	10	45	60
	Total	75	60	65	200

$$p_o = P(1) + P(2) + P(3) = 0.725$$

$$p_e = P(1|A)P(1|B) + P(2|A)P(2|B) + P(3|A)P(3|B) = 0.313$$

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 0.588$$



Quality metric II: quadratic weighted kappa

Images have five possible ratings, 0,1,2,3,4. Image is characterized by a tuple (e_a, e_b), which corresponds to scores by *RaterA* (human) and *RaterB* (predicted).

Quadratic weighted kappa is calculated as:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}},$$

where:

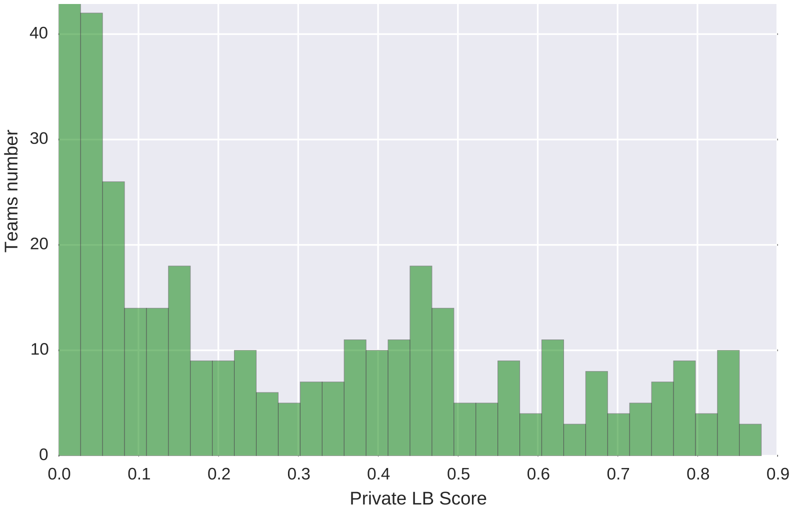
- O is $N \times N$ histogram matrix, such that $O_{i,j}$ corresponds to the number of images that received a rating i by A and a rating j by B.
- E, is $N \times N$ histogram matrix of expected ratings. E calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between each rater's histogram vector of ratings, normalized such that E and O have the same sum.
- w is $N \times N$ matrix of weights, which calculated based on the difference between raters' scores:

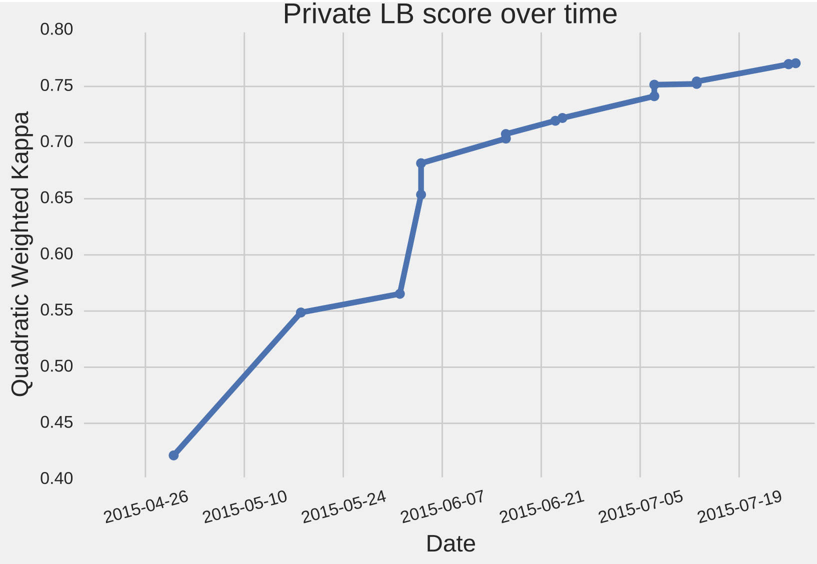
$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2}$$

Private leaderboard

#	Δrank	Team Name ‡ model uploaded * in the money	Score	Entries	Last Submission UTC (Best – Last Submission)
1	—	Min-Pooling ‡ *	0.84958	37	Mon, 27 Jul 2015 22:38:33 (-0h)
2	—	o_O ‡ *	0.84479	61	Mon, 27 Jul 2015 18:31:43 (-10.3h)
3	—	Reformed Gamblers ‡ *	0.83937	58	Mon, 27 Jul 2015 04:40:09 (-17d)
4	—	Julian de Wit & Daniel Hammack	0.83626	49	Mon, 27 Jul 2015 23:57:17 (-5.7h)
5	—	Jeffrey De Fauw	0.82899	133	Mon, 27 Jul 2015 18:31:46 (-0.1h)
6	—	DeepSense.io	0.82854	88	Mon, 27 Jul 2015 23:58:39 (-0.3h)
		...			
23	↑2	gstieger	0.75170	18	Mon, 27 Jul 2015 23:28:36 (-0.2h)
24	**↓3**	**Sergey, Rasim, Alexander**	**0.75060**	**59**	**Mon, 27 Jul 2015 23:27:44 (-0.1h)**
25	↓1	tk	0.75025	25	Mon, 27 Jul 2015 20:45:32 (-22.8h)
26	↑1	Sungheon Park	0.73599	31	Mon, 27 Jul 2015 14:31:22
		...			
661	↓328	SAKI	-0.00516	1	Wed, 03 Jun 2015 17:14:39

Teams distribution on private LB




Private LB score over time
Quadratic Weighted Kappa
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
2015-04-26
2015-05-10
2015-05-24
2015-06-07
2015-06-21
2015-07-05
2015-07-19
Date

Domain knowledge: diabetic retinopathy symptoms

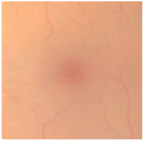
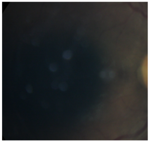
- DR symptoms cheatsheet: https://goo.gl/s5IMt8

Симптомы
(от слабого к сильному)

1. Ветвления/искривления вен
2. Микрокровоизлияния
3. Твёрдые Экссудаты
4. Мягкие Экссудаты
5. Обширные Кровоизвляния (преретинальные)
6. Мембраны (плёночки)
7. Отслойка

Всё зависит от количества. Как правило больше дефектов - хуже зрение.

Макула
Макула выражена, не должно быть отёка, т.е. расплывшегося пятна

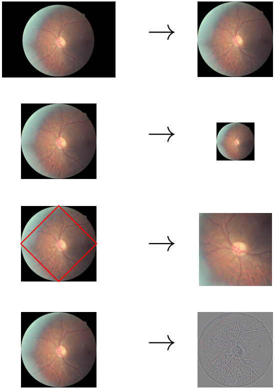
Нормальная макула	Отёкшая
	

Was prepared with assistance of Vera Shevchenko

- International Clinical Diabetic Retinopathy Disease Severity Scale, Detailed Table: http://www.icoph.org/downloads/Diabetic-Retinopathy-Detail.pdf

Preprocessing

- Crop black borders
- Extent to square + resize
- Optionally: crop internal square
- Optionally: Local Contrast Normalization (LCN)
  $\hat{I}_{x,y} = \frac{I_{x,y} - \mu_{x,y}}{\sigma_{x,y}}$

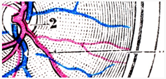


Augmentation

- Stuff that worked
  - Vertical/horizontal Mirror
  - Random shifts
  - Random color noise
- Stuff that not quite worked
  - Rotations
  - Krizhevsky-style[1]color augmentation
  - Scaling
  - Sheering
  - Many more...

We suppose that few augmentations worked because of insufficient depth of our network, but experiments with deeper nets led to more overfitting.

[1] Krizhevsky, Sutskever, and Hinton 2012.

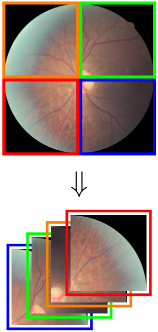


Network configuration

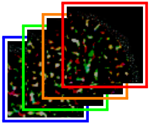
	Layer type	Size		Output Shape	Outputs
1	InputLayer			(64, 3, 256, 256)	196 608
2	**SliceRotateLayer**			(256, 3, 128, 128)	49 152
3	Conv2DDNNLayer	3x3	LReLU	(256, 64, 126, 126)	1 016 064
4	MaxPool2DDNNLayer	3x3	stride 2x2	(256, 64, 62, 62)	246 016
5	DropoutLayer		P=0.1	(256, 64, 62, 62)	246 016
6	Conv2DDNNLayer	3x3		(256, 96, 60, 60)	345 600
7	MaxPool2DDNNLayer	3x3	stride 2x2	(256, 96, 29, 29)	80 736
8	DropoutLayer		P=0.2	(256, 96, 29, 29)	80 736
9	Conv2DDNNLayer	3x3	LReLU	(256, 128, 27, 27)	93 312
10	DropoutLayer		P=0.3	(256, 128, 27, 27)	93 312
11	Conv2DDNNLayer	3x3	LReLU	(256, 96, 25, 25)	60 000
12	MaxPool2DDNNLayer	3x3	stride 2x2	(256, 96, 12, 12)	13 824
13	DropoutLayer		P=0.4	(256, 96, 12, 12)	13 824
14	Conv2DDNNLayer	3x3	LReLU	(256, 128, 10, 10)	12 800
15	MaxPool2DDNNLayer	2x2	stride 2x2	(256, 128, 5, 5)	3 200
16	**RotateMergeLayer**			(64, 12800)	12 800
17	DropoutLayer		P=0.5	(64, 12800)	12 800
18	DenseLayer	512		(64, 512)	512
19	FeaturePoolLayer		FeaturePool	(64, 256)	256
20	DropoutLayer		P=0.5	(64, 256)	256
21	DenseLayer	512		(64, 512)	512
22	FeaturePoolLayer		FeaturePool	(64, 256)	256
23	DropoutLayer		P=0.5	(64, 256)	256
24	DenseLayer			(64, 4)	4

Special layers

SliceRotateLayer



RotateMergeLayer



⇓

Dense layer

Ordinal regression

- Ordinal regression[2] is like an ordered classification.
- Target coding:

$0 \rightarrow 0\ 0\ 0\ 0$
$2 \rightarrow 1\ 1\ 0\ 0$
$4 \rightarrow 1\ 1\ 1\ 1$

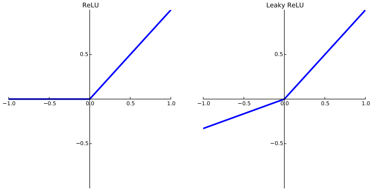
- Do not normalize the sigmoids in the last fully-connected layer:

$$\frac{e^{-z_i}}{\sum_{i=1}^{K} e^{-z_i}} \rightarrow \frac{1}{1+e^{-z_i}}$$

- Use MSE loss function.

[2] Jianlin Cheng, Zheng Wang, and G. Pollastri (2008). "A neural network approach to ordinal regression". In: *Neural Networks, 2008. IJCNN 2008*.

Activations



We used Leaky ReLUs for convolutional layer, this activation function acts like a regularizer.
Used Maxout[3] activations for fully-connected layers.

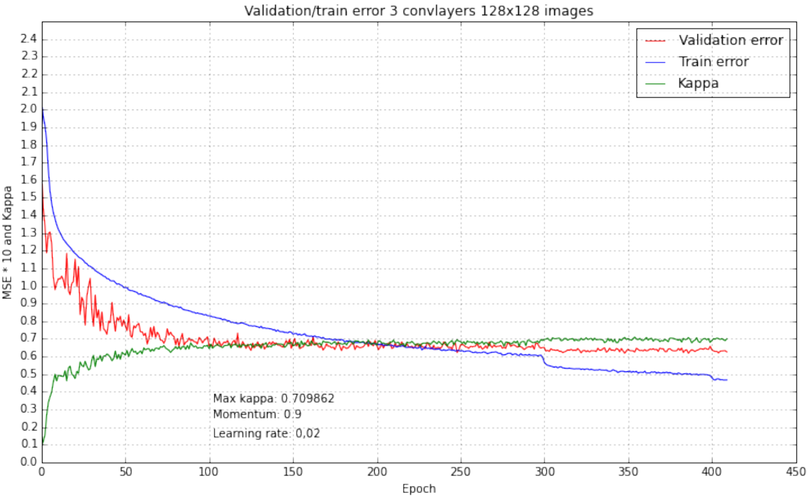
[3] "Ian J. Goodfellow et al. (2013). "Maxout Networks". In: arXiv: abs/1302.4389v4.

Optimization I

- SGD with Nesterov[4] momentum of 0.9
- About 450000 gradient steps
- 3 step learning rate decay: 0.02, 0.01, 0.001
- Minibatch size of 32

[4] Y.Nesterov (1983). "A method for unconstrained convex minimization problem with the rate of convergence $o(\frac{1}{k^2})$". In: *Doklady AN SSSR*.

Optimization II



Overfitting

- During the whole competition we were badly overfitting.
- Not all of the augmentation approaches worked for us, which made things worse.
- The best thing we could come up with was an extensive dropout usage.

Decision making

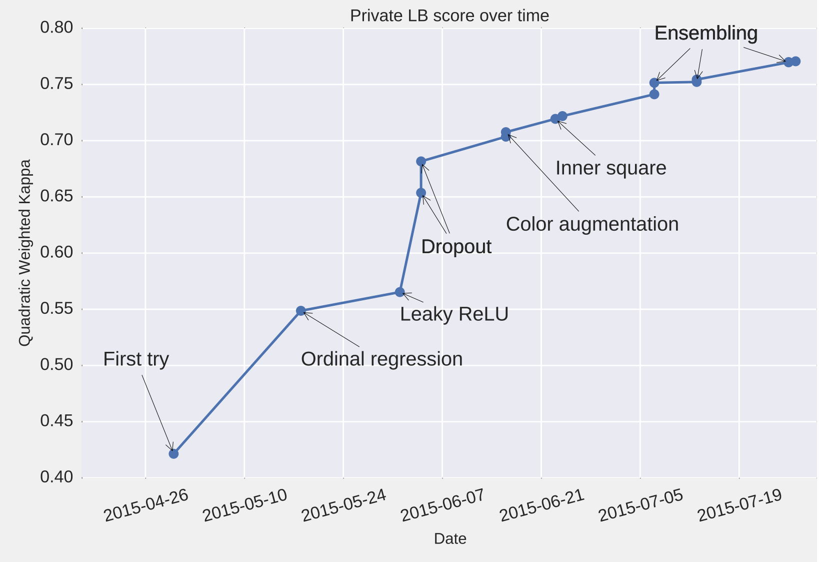
- Each eye was processed independently
- A maximum score among the eyes was assigned to both of them
- This improved the score by 0.05 points

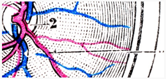
Ensembling

- Our final submission was en ensemble of 5 neural networks.
- The predictions we weighted according to the confusion matrix on a held-out validation set.
- This improved the score by 0.05 points (7 % improvement).

- The solution was built in Python on top of Lasagne[5] and Theano[6]
- Numpy, Pandas and scikit-image were used for loading and manipulating data
- Self-written C++/OpenCV utilities for preprocessing and microaneurism detection
- Theano was built with cuDNN support
- Code available at https://github.com/dudevil/DRD

[5] https://github.com/Lasagne/Lasagne
[6] http://deeplearning.net/software/theano/


Private LB score over time
Quadratic Weighted Kappa
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
Ensembling
Inner square
Color augmentation
Dropout
Leaky ReLU
Ordinal regression
First try
2015-04-26
2015-05-10
2015-05-24
2015-06-07
2015-06-21
2015-07-05
2015-07-19
Date

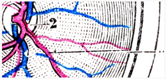


Motivation I

Microaneurysms the earliest clinical sign of diabetic retinopathy; they appear as small, red dots in the superficial retinal layers

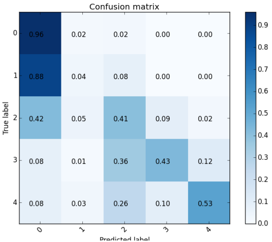
Disease level	Findings observable upon dilated ophthalmoscopy
None	No abnormalities
Mild	Microaneurysms only
Moderate	More than just MA but less than severe NPDR
Severe	>20 intraretinal hemorrhages in each quad or Definite venous beading in 2+ quads or Intraretinal microvascular anomalies in 1+ quad
Proliferative	Neovascularization or/and Vitreous/preretinal hemorrhage

International Clinical Diabetic Retinopathy Disease Severity Scale, Detailed Table: http://www.icoph.org/downloads/Diabetic-Retinopathy-Detail.pdf



Motivation II

We had problems with detection of early symptoms



Confusion matrix on 256x256 pixels input

MA have round shape with 2-5 pixels in radius on 1024x1024 image

MA became invisible after downsampling to 256x256

$\Rightarrow$ Classes 0,1,2 almost indistinguishable due to low resolution

We have not enough resources&data to learn on highres images

$\Rightarrow$ Let's try plain old image processing

Microaneurysm candidates using the determinant of the Hessian I

We want to know how much pixel location is similar to blob shape. Let's calculate Hessian matrix at that point:

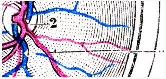
$$H(\mathbf{x}) = \begin{bmatrix} L_{xx}(\mathbf{x}) & L_{xy}(\mathbf{x}) \\ L_{xy}(\mathbf{x}) & L_{yy}(\mathbf{x}) \end{bmatrix}$$

- $L_{aa}(\mathbf{x})$ is second partial derivative in the a direction
- $L_{ab}(\mathbf{x})$ is the mixed partial second derivative in the a and b directions.

Derivatives are computed in some scale σ_I – smoothed by a Gaussian kernel

$$L(\mathbf{x}) = g(\sigma_I) \otimes I(\mathbf{x}).$$

Derivatives must be scaled by factor related to the Gaussian kernel: σ_I^2.



Microaneurysm candidates using the determinant of the Hessian II

At each scale, **blobs points** are those points that are local extrema of determinant the Hessian matrix.

$$\det H(x;\sigma) = \sigma_I^2 (L_{xx} L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x}))$$

Sign of the trace of Hessian matrix help distinguish dark from light points:

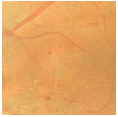
$$\operatorname{trace} H(x;\sigma) = \sigma_I (L_{xx} + L_{yy}).$$

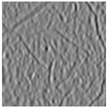
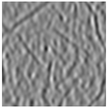
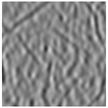
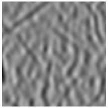
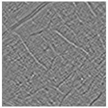
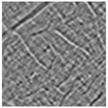
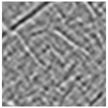
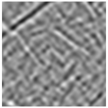
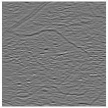
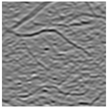
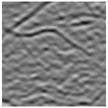
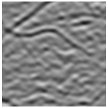
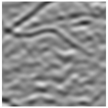
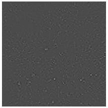
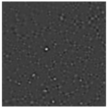
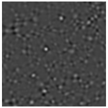
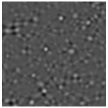
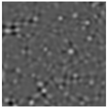
Straightforward differential blob detector with automatic scale selection:

$$(\hat{x}, \hat{\sigma}) = \operatorname{argmaxlocal}_{(x;t)}(\det H(x;\sigma))$$

Microaneurysm candidates using the determinant of the Hessian III

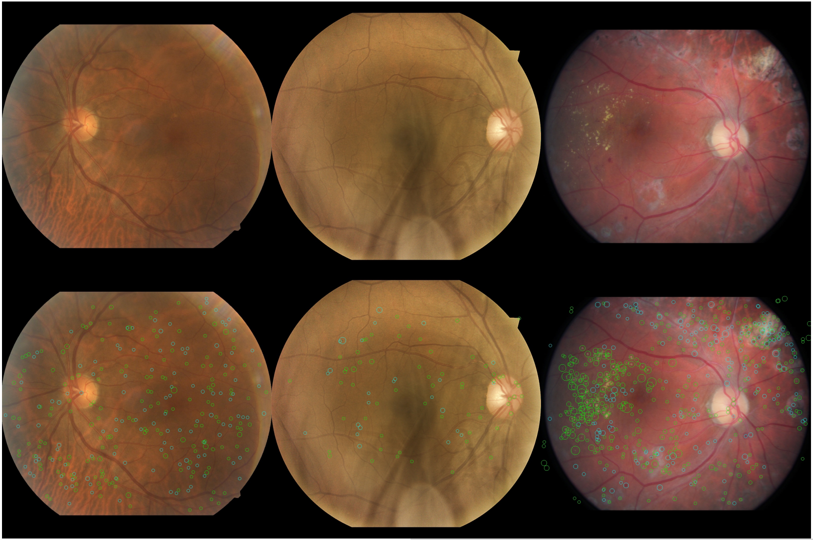
687_right



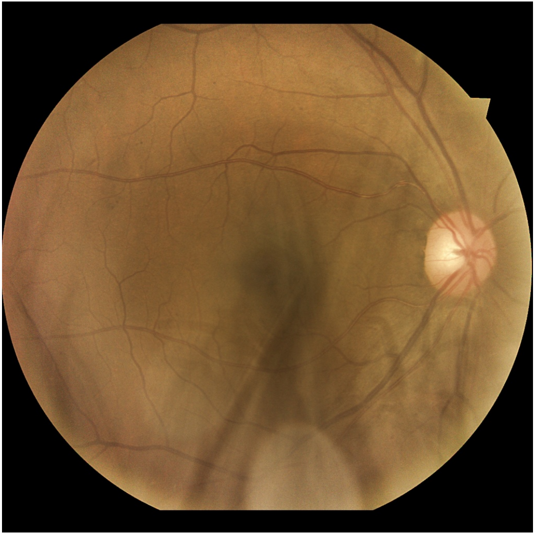
	$\sigma = 1.7$	$\sigma = 3.4$	$\sigma = 5.1$	$\sigma = 6.8$	$\sigma = 8.0$
L_{xx}					
L_{xy}					
L_{yy}					
$\det H(x;\sigma)$					

Microaneurysm candidates

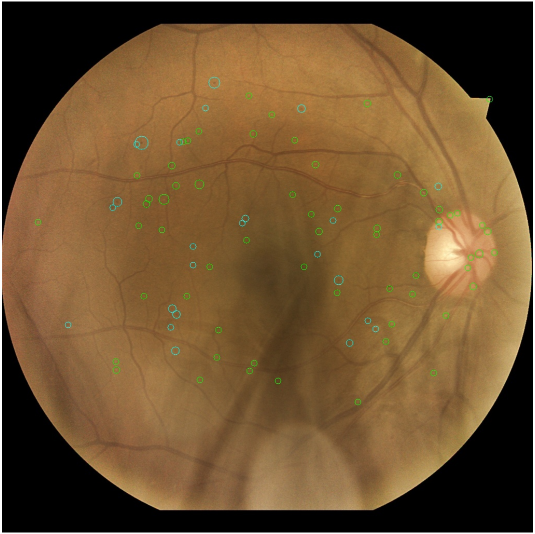
Normal	Moderate	Proliferative



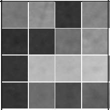
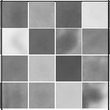
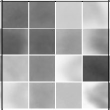
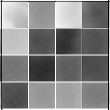
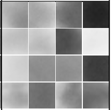
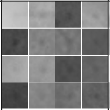
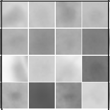
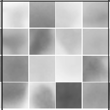
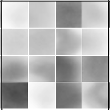
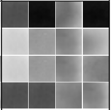
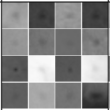
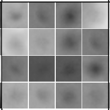
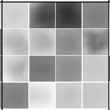
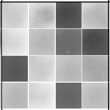
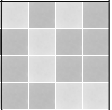
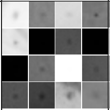
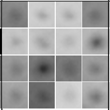
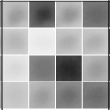
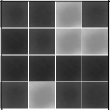
Microaneurysm candidates



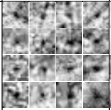
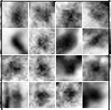
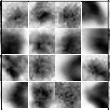
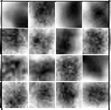
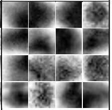
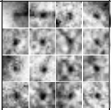
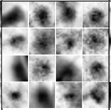
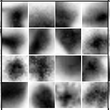
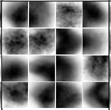
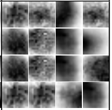
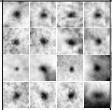
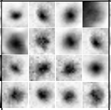
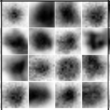
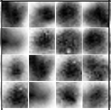
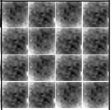
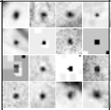
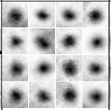
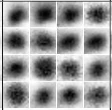
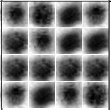
Microaneurysm candidates



How blobs looks

strength	$\sigma = 1.7$	$\sigma = 3.4$	$\sigma = 5.1$	$\sigma = 6.8$	$\sigma = 8.0$
$[300, 450)$					
$[450, 600)$					
$[600, 750)$					
$[750, \infty)$					

How blobs looks

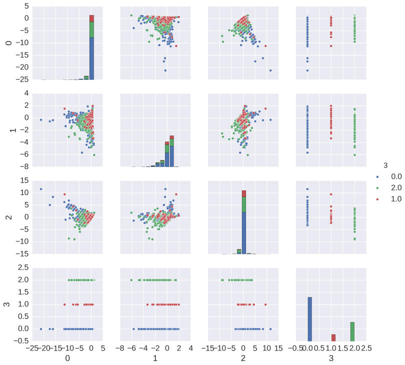
strength	$\sigma = 1.7$	$\sigma = 3.4$	$\sigma = 5.1$	$\sigma = 6.8$	$\sigma = 8.0$
$[300, 450)$					
$[450, 600)$					
$[600, 750)$					
$[750, \infty)$					

BoVW preparation

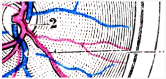
- Extract local descriptors from blob patch: HOG, LBP
- Create code book using K-means vector quantization
- Use histograms of visual words as feature vectors

BoVW preparation

Unfortunately I got stuck on this point two weeks before challenge deadline. :-(



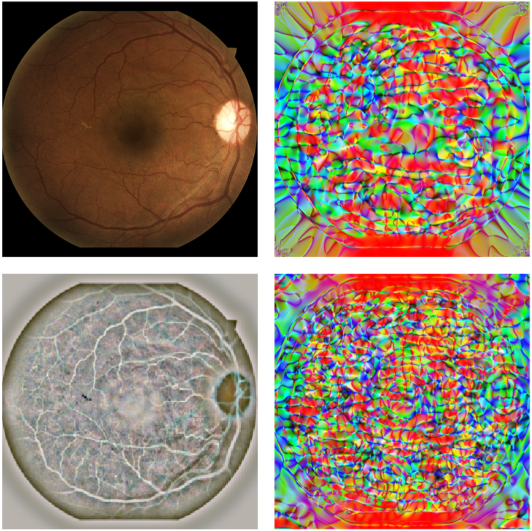
Typical picture of BoW features after applying PCA.



Conclusions

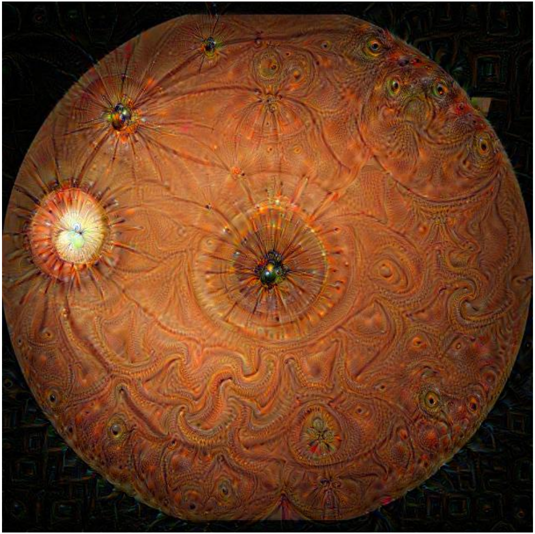
- Teamwork
  - Find good tools for effective collaboration
  - Divide and conquer
  - Learn together
  - Maintain model diversity
- Competition
  - Setup a reliable experiment-evaluate loop
  - Be careful when keeping track of experiments
  - Plan ahead when experiments are long (24+ hours in out case)
  - Understand evaluation metric
  - Try different things

Bonus images



Visualization of buggy blobs detector output

Bonus images



Network dreams on retina image

Bonus images

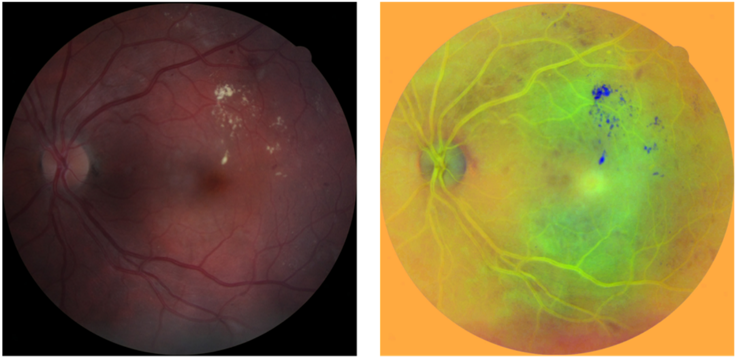
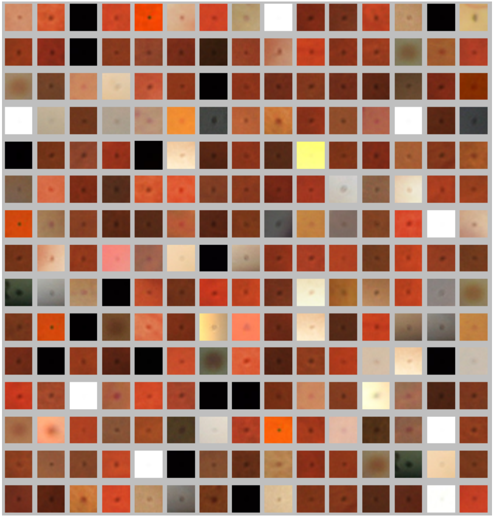


Image recoloring by 'PCA colors'

Bonus images



Strongest blobs found on retina images