

# Face recognition

# Abstract

- Я запутался
- Все статьи, которые я нашёл, делают что-то своё
- SOTA результаты это "настакаем слоёв"

# О чём будет доклад

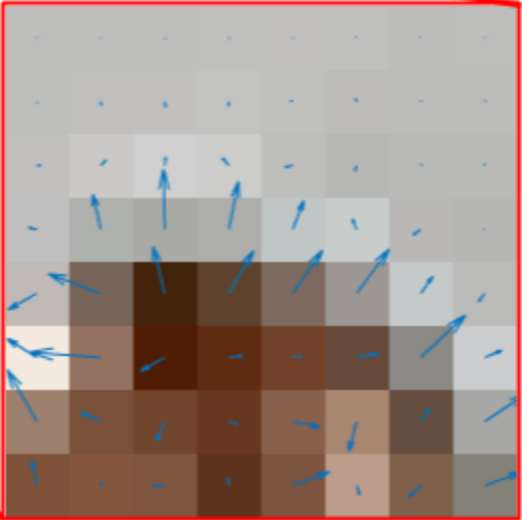
## Face Detection

- Face Detection: Histogram of Oriented Gradients and Bag of Feature Method
- Multi-view Face Detection Using Deep Convolutional Neural Networks
- Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks
- From Facial Parts Responses to Face Detection: A Deep Learning Approach

## Face Recognition

- FaceNet: A Unified Embedding for Face Recognition and Clustering
- DeepFace: Closing the Gap to Human-Level Performance in Face Verification
- Deep Face Recognition

**HOG**

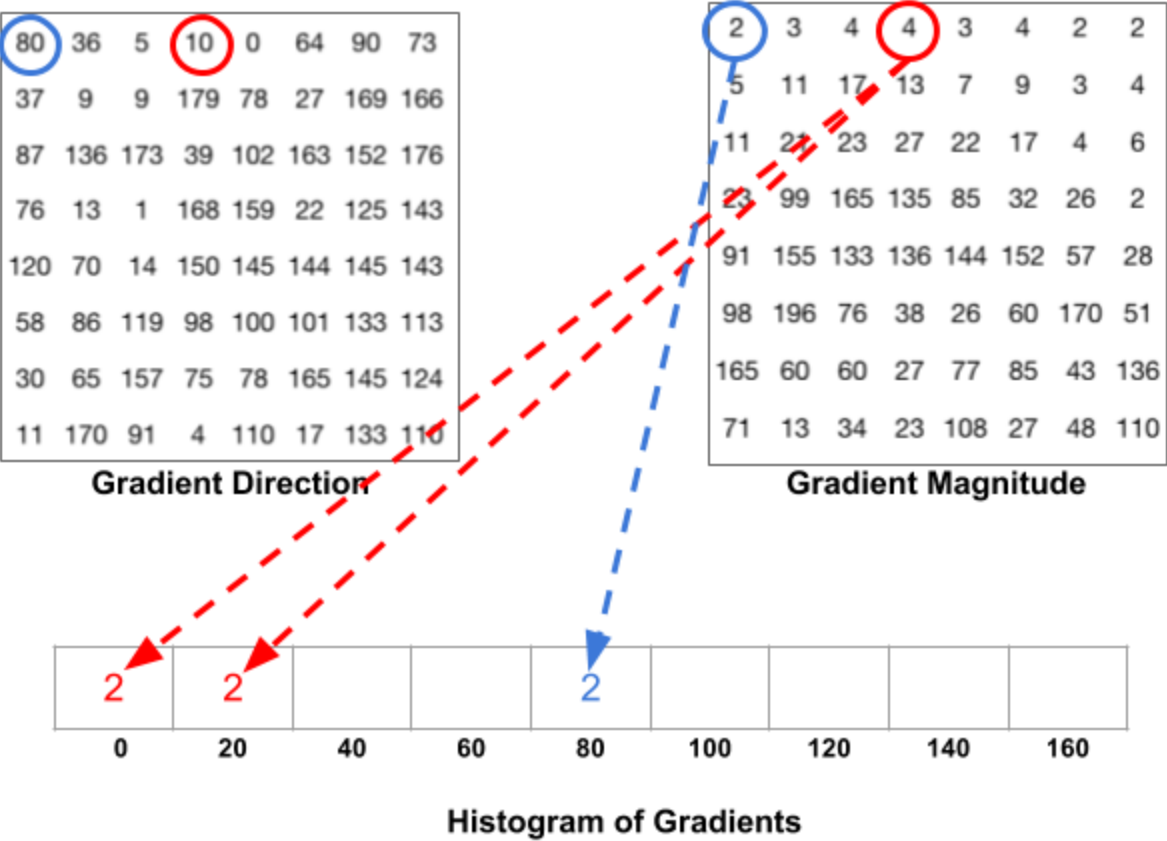


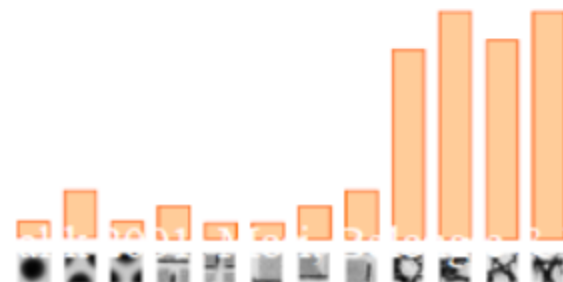
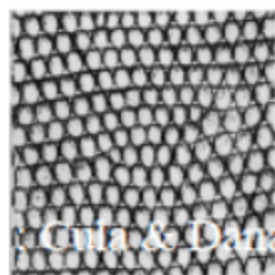
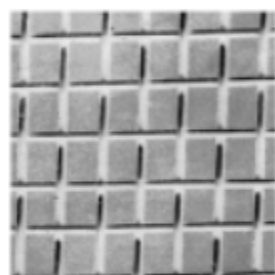
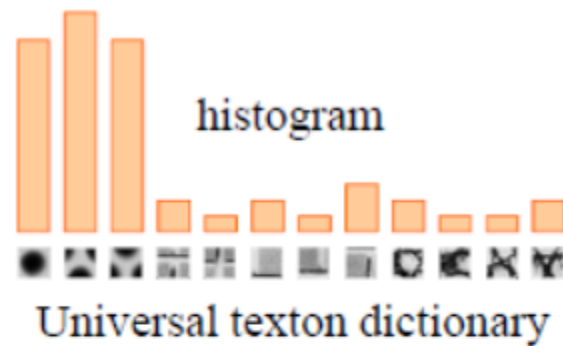
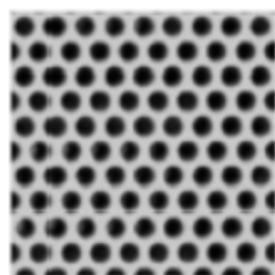
2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

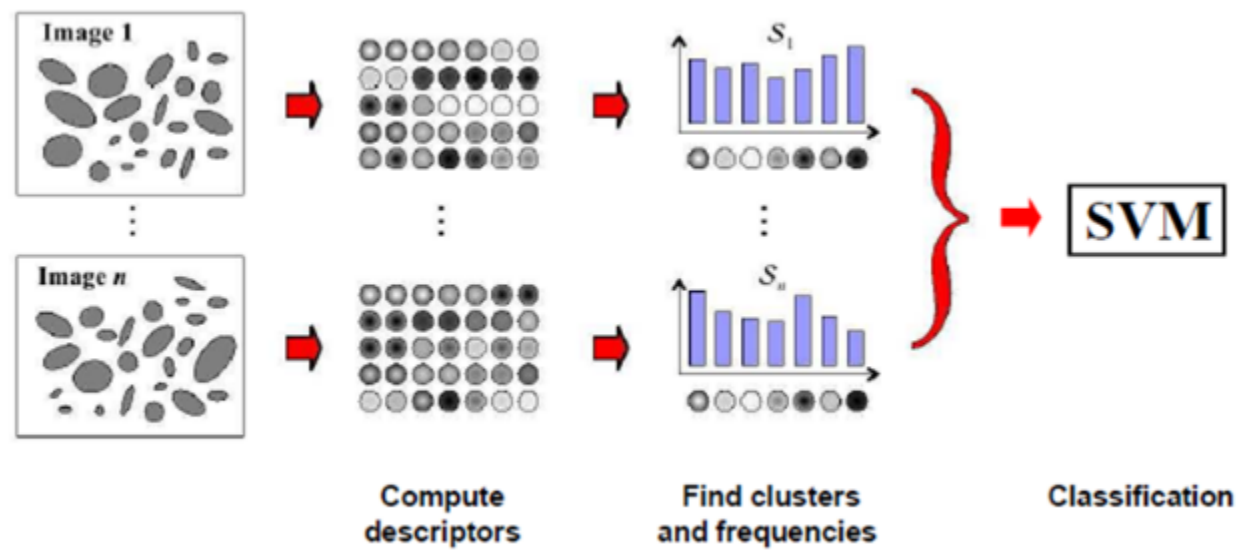
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction





© Cui & Dan

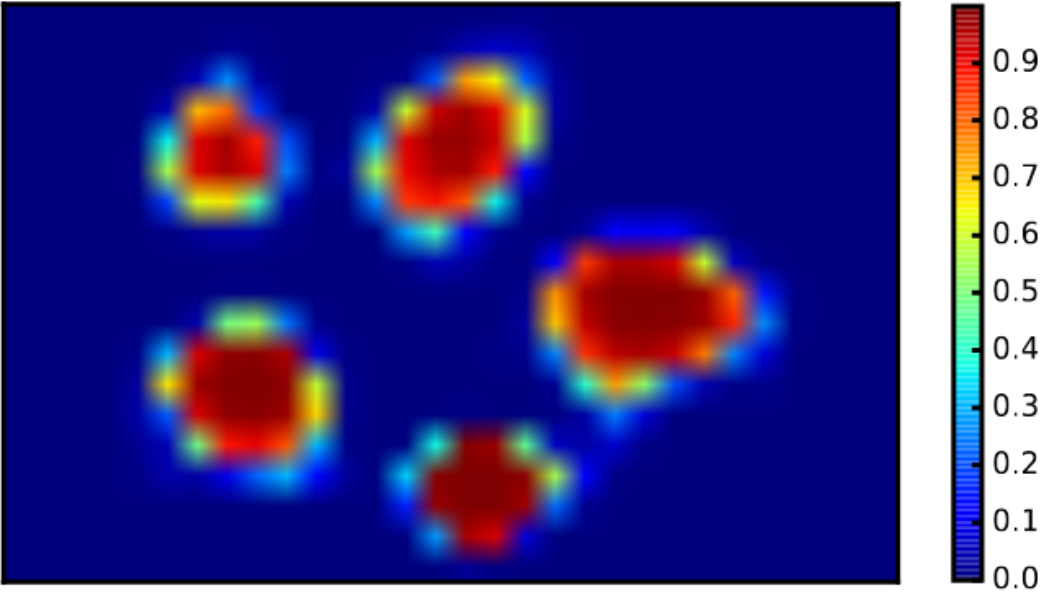
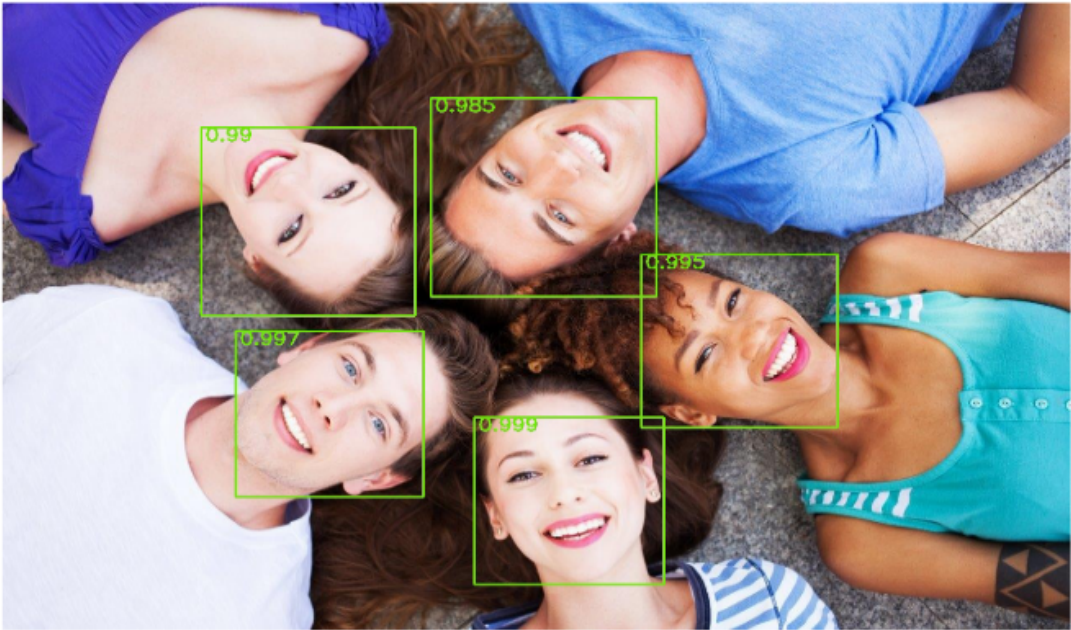




# Multi-view Face Detection Using Deep CNNs

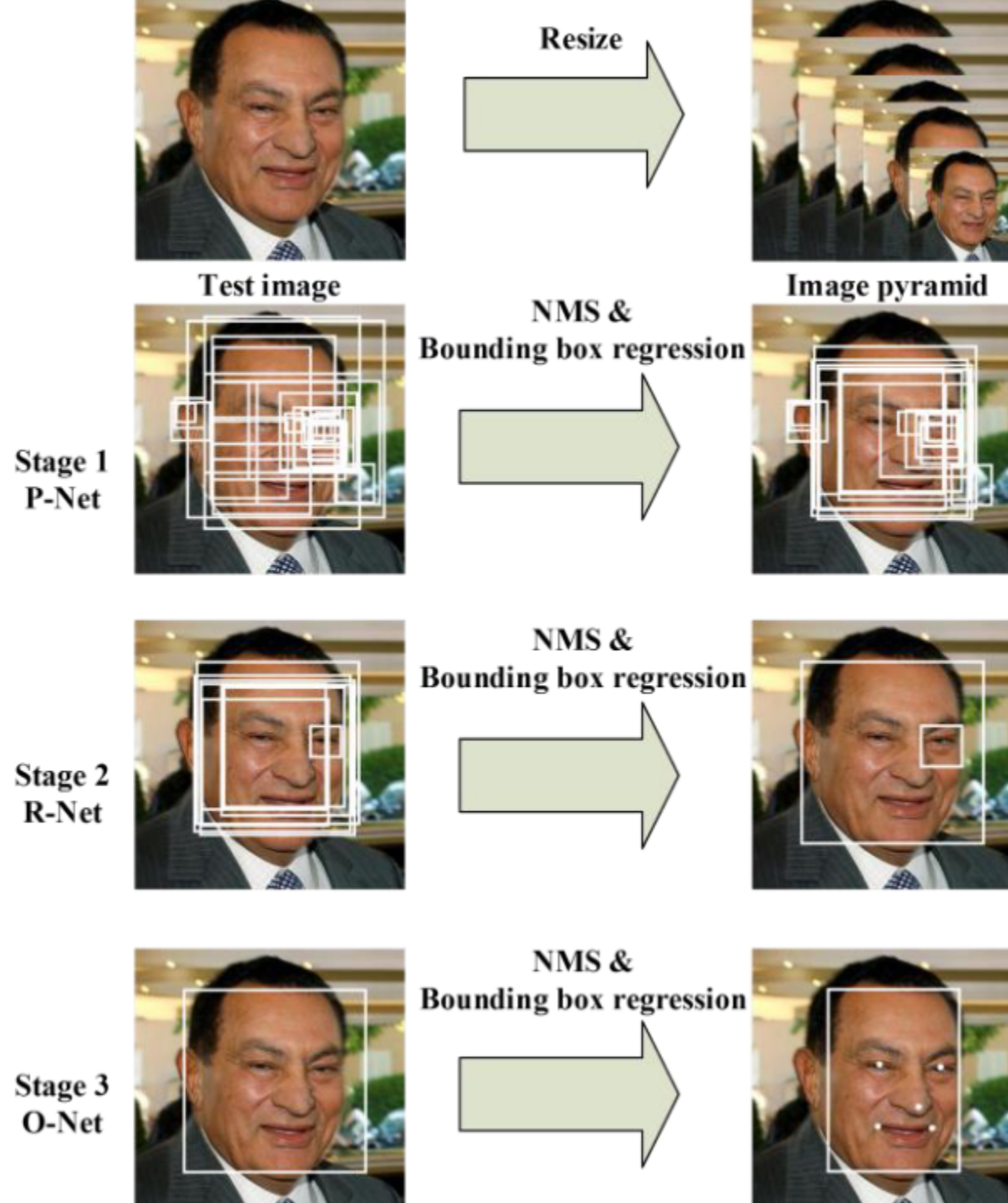
- Файнтюним AlexNet на детекцию лиц (просто пихаем ему картинки, где больше 50% это лицо и не лица)
- Жахаем sliding window на разные скейлы картинки чтобы выделить участки, где есть лицо
- Если заменить полносвязные слои на сверточные, то можно генерировать хитмапы

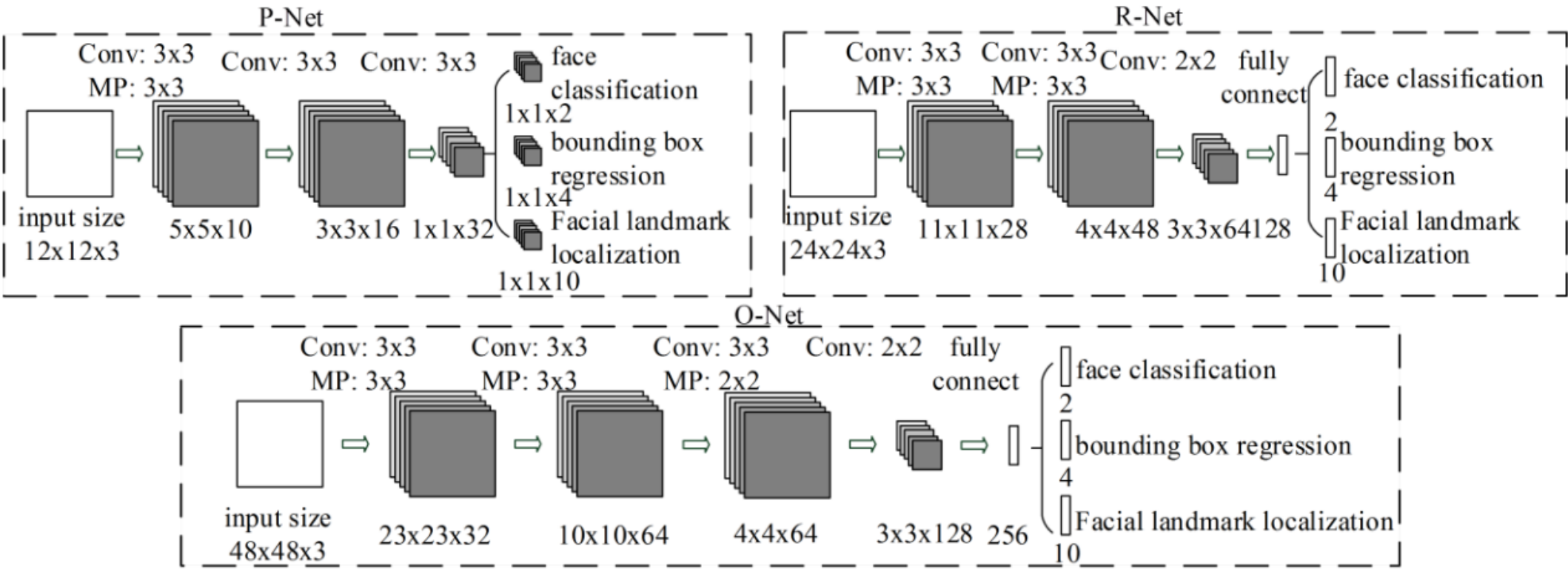
Multi-view Face Detection Using Deep CNNs



# Cascaded CNN

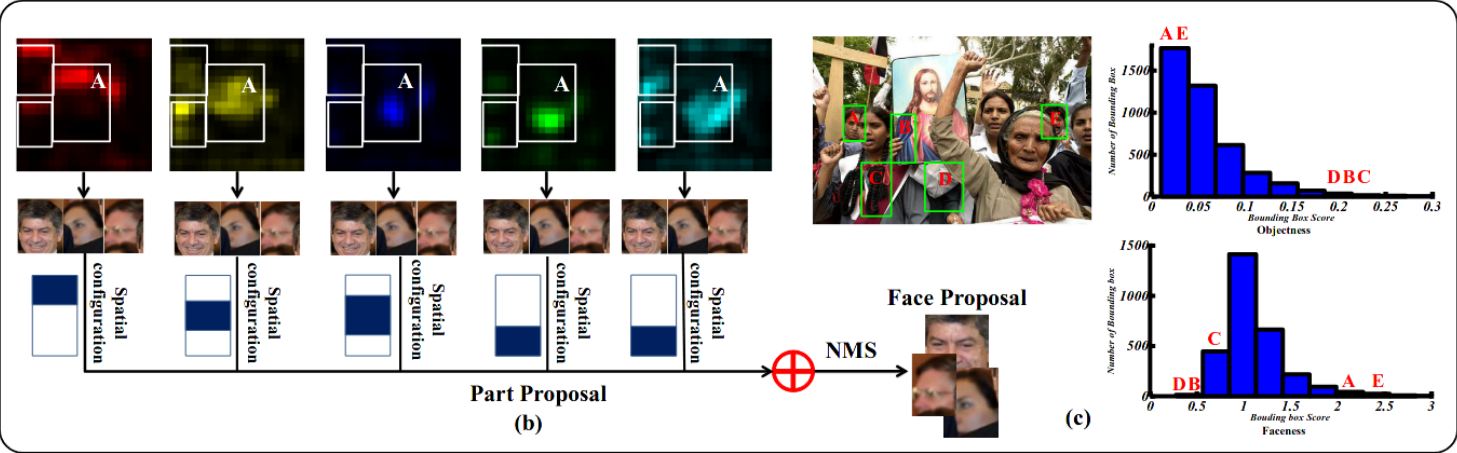
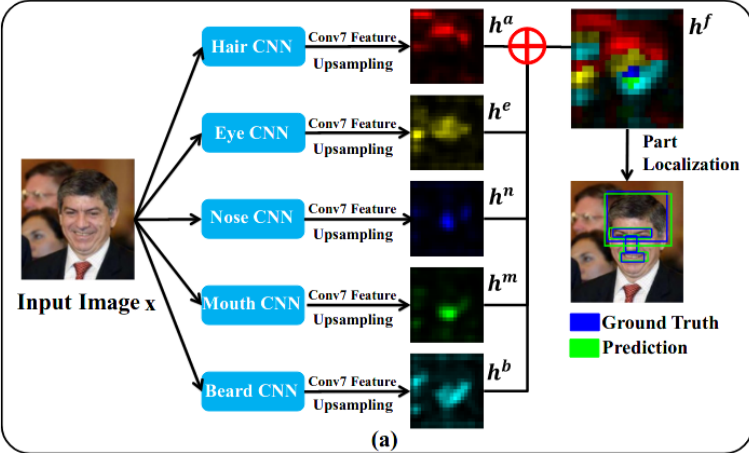
0. Заресайзить картинку несколько раз, делая "пирамиду изображений"
1. Пихаем картинку на вход простой сети для генерации претендентов на лицо, объединяем выходы
2. Пихаем кандидаты 1 стадии на вход новой сети, объединяем выходы
3. Пихаем кандидаты 2 стадии на вход новой сети, она выдает ВВ и лэндмарки





# Deep Learning Approach





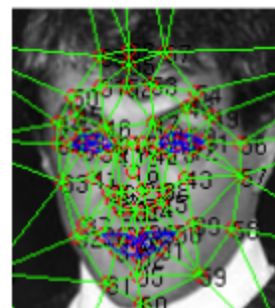
**DeepFace**



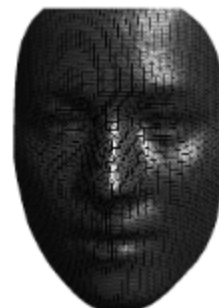
(a)



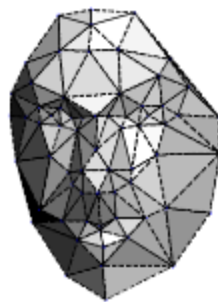
(b)



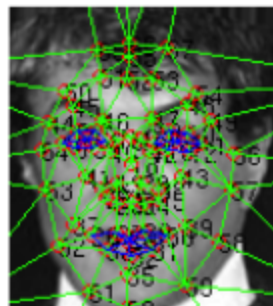
(c)



(d)



(e)



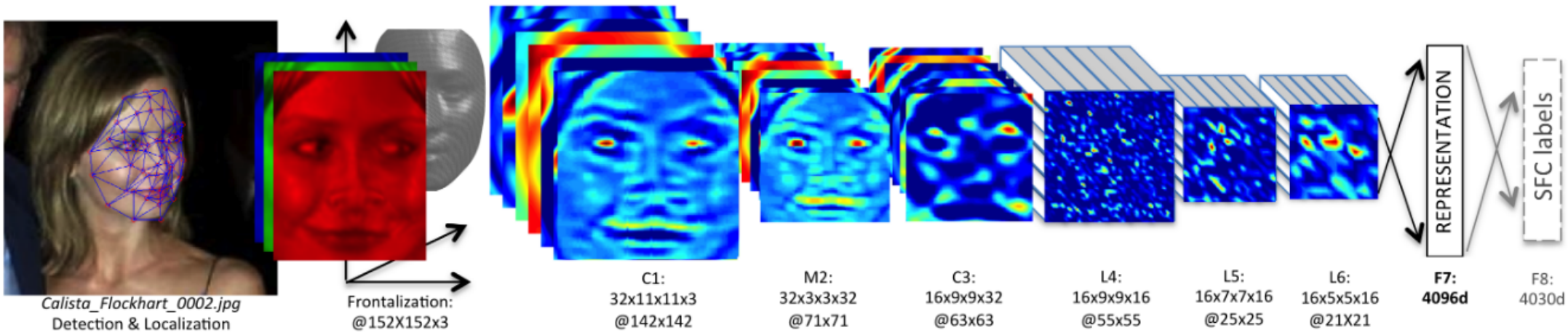
(f)



(g)



(h)



## Как сравнивать выходы?

Используем  $\chi^2$ -similarity:

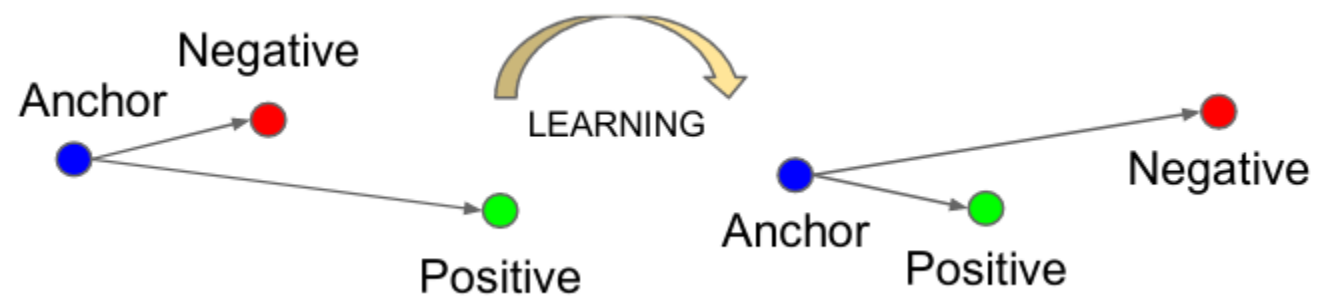
$$\chi^2(f_1, f_2) = \sum_i w_i (f_1[i] - f_2[i])^2 / (f_1[i] + f_2[i])$$

Веса достаём, обучая линейный SVM на векторах  $(f_1[i] - f_2[i])^2 / (f_1[i] + f_2[i])$

# FaceNet



# Triplet loss





Хотим:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2$$

А значит:

$$L = \sum_i^N \|f(x_i^a) - f(x_i^n)\|_2^2 - \|f(x_i^a) - f(x_i^p)\|_2^2 - \alpha$$

## Как выбирать триплеты?

Хотим  $x_i^p = \arg \max_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$

и  $x_i^n = \arg \max_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$

Искать такое дело по всей выборке сложно и вообще говоря плохо, поэтому будем аргмаксить по минибатчам.

# Deep Face Recognition

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmax
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

## Как обучать?

Идея: давайте сначала обучим классификатор на  $N$  классов с обычным логлоссом, а потомотрежем ему голову и дофайнтюним триплет лоссом.