

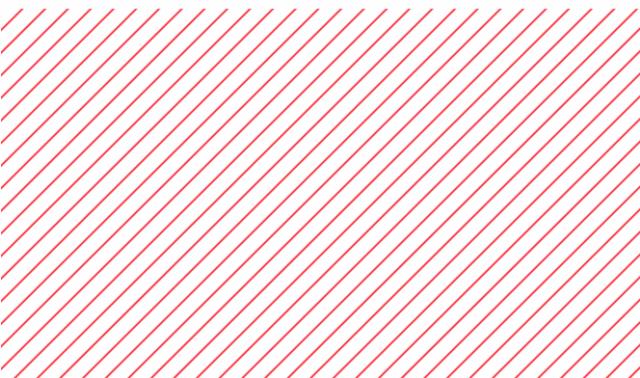
академия
больших
данных



Обработка изображений лиц на мобильных устройствах

Андрей Савченко

Профессор НИУ ВШЭ-Нижний Новгород



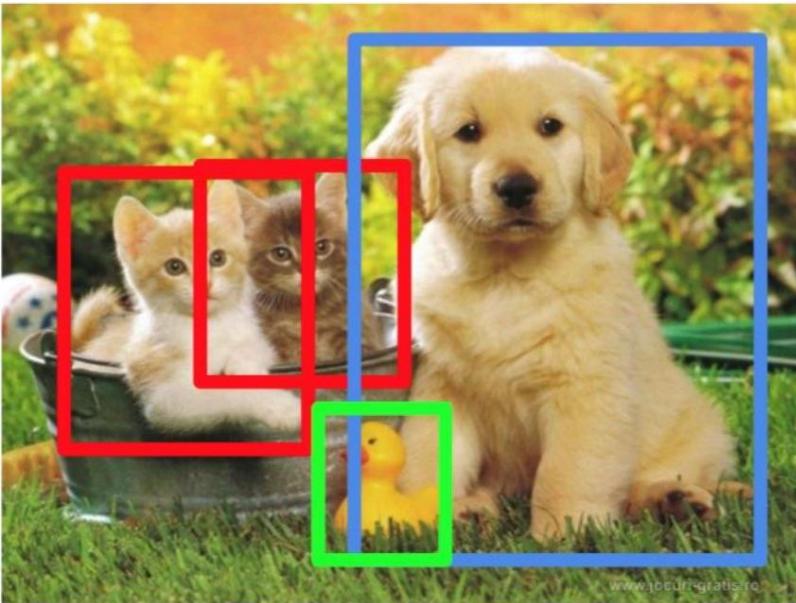


Опрос

<https://forms.gle/3M2noR6cwGVSyuQD6>

Детектирование лиц

Детектирование объектов. Постановка задачи (по материалам В. Писаревского, руководителя проекта OpenCV)



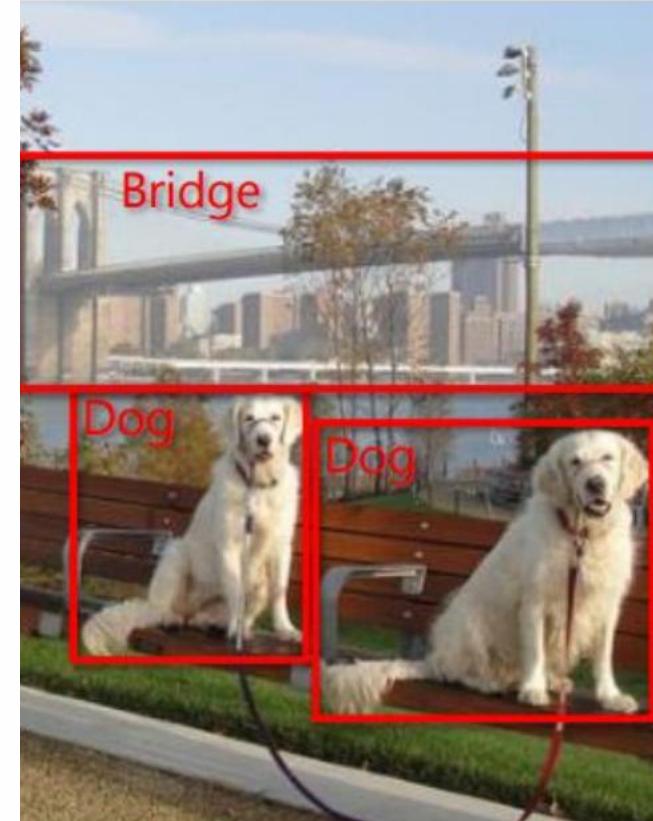
Определить положение (прямоугольник) и метку для каждого объекта (из определенного множества классов) на изображении

Кошка

Собака

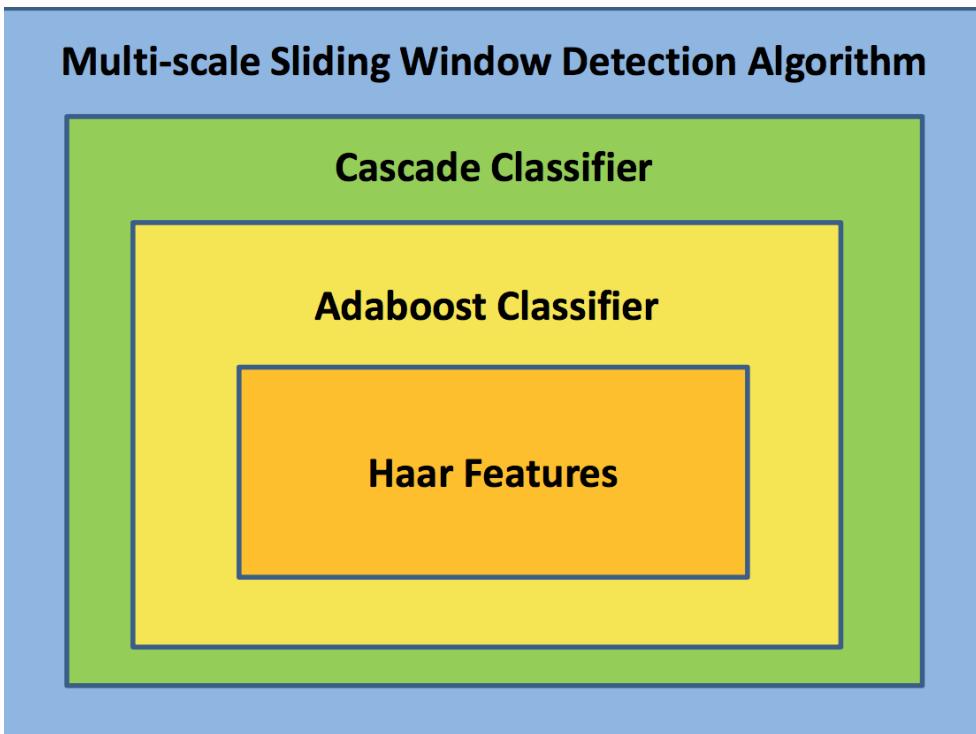
Утка

Распознавание (классификация) и детектирование



Метод Виолы-Джонса

Метод Виолы-Джонса – один из первых детекторов лиц в режиме реального времени – 15 FPS на Pentium III



Sliding window. Детектирование объектов как задача классификации



Классы = [Кошка, Собака, Утка]

Кошка (окно(0,0,w,h))? **Нет**

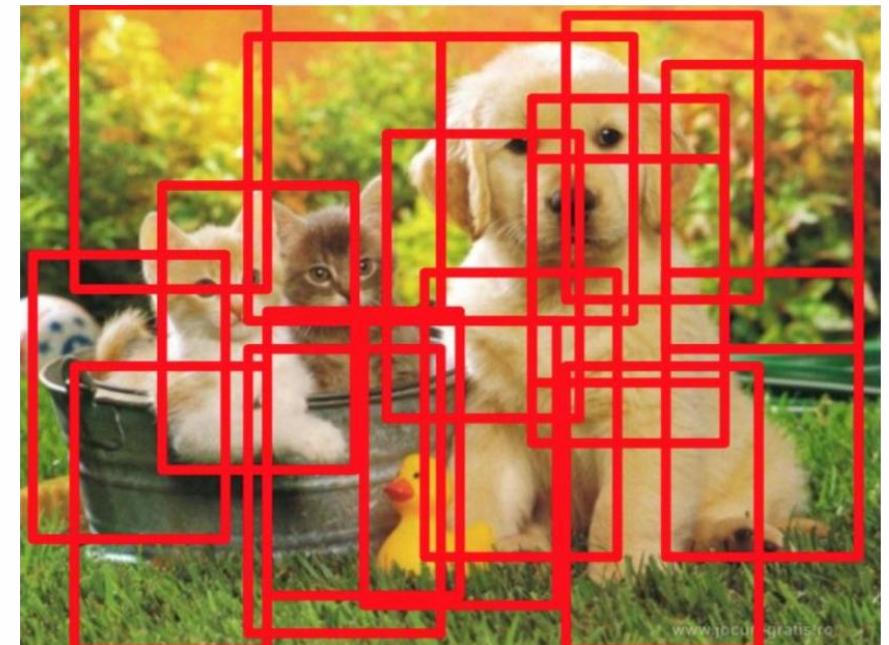
Собака (окно(0,0,w,h))? **Нет**

Утка(окно(0,0,w,h))? **Нет**

Перебираем разные положения и
размеры окон



Классификатор должен быть **очень**
быстрым!



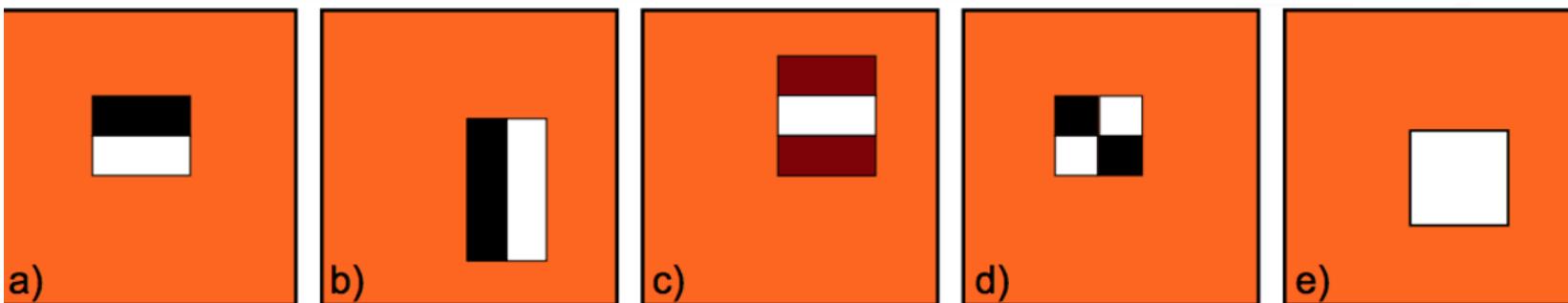
Multi-scale sliding window approach

Пирамида + фиксированный размер окна:

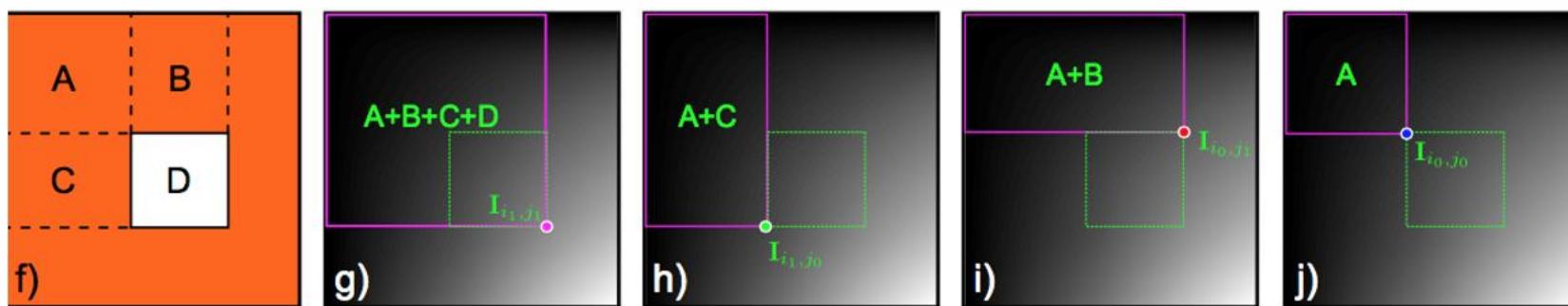
1. Нужно натренировать только один классификатор на каждый класс.
2. На верхних слоях пирамиды меньше позиций окна – экономим вычисления
3. Фиксированное соотношение сторон: ok для лиц, почти ok для пешеходов, не ok – для машин



Фильтры Хаара (1)



Интегральное изображение

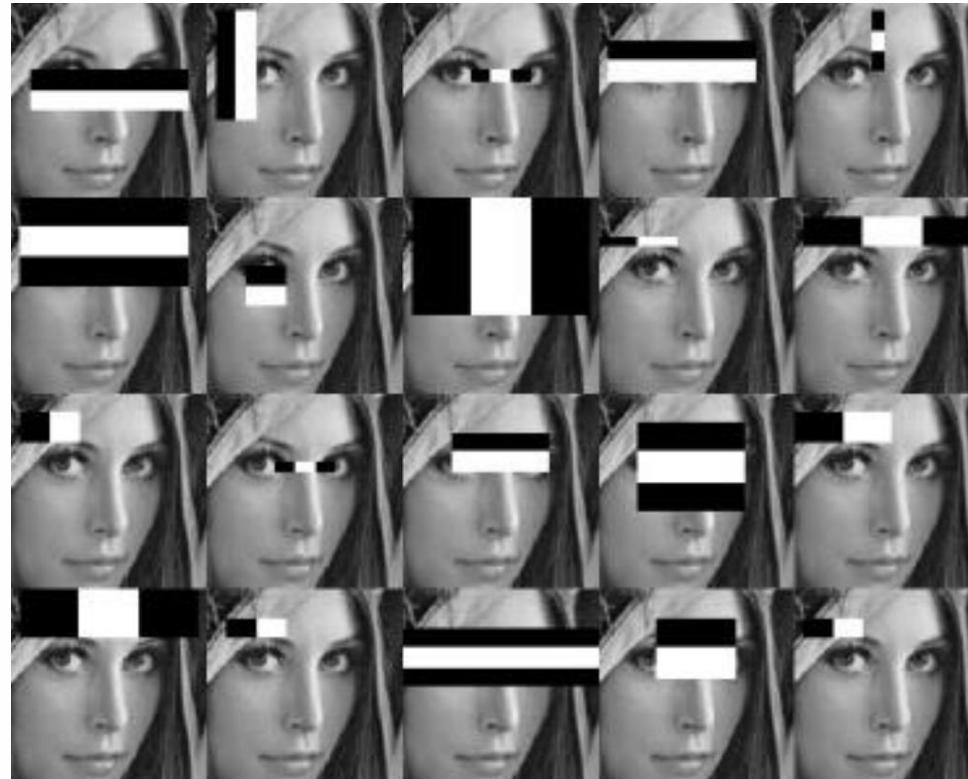
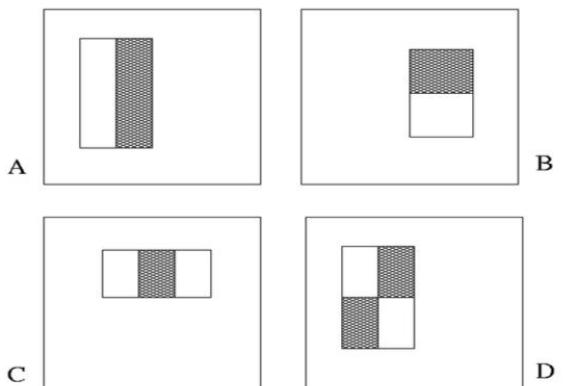
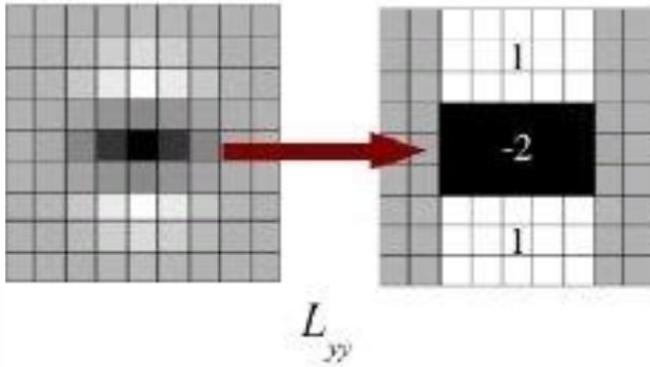


$$I_{i_1, j_1} + I_{i_0, j_0} - I_{i_1, j_0} - I_{i_0, j_1} = (A+B+C+D) + A - (A+C) - (A+B) = D$$

Фильтры Хаара (2)

Тренировочная выборка: 10000 лиц приведенных к размеру 20x20, примерно столько же не-лиц 20x20.

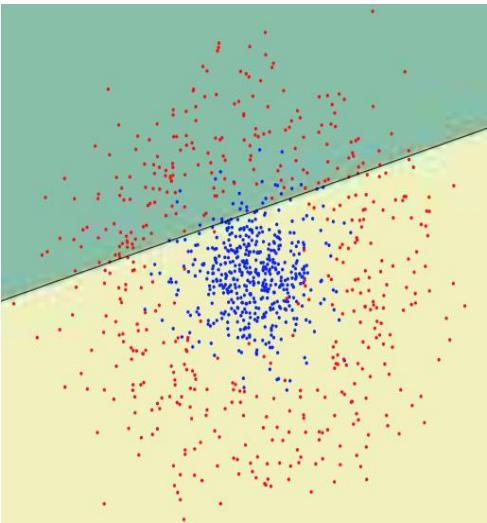
Для окна 20x20 всего порядка 40000 признаков Хаара



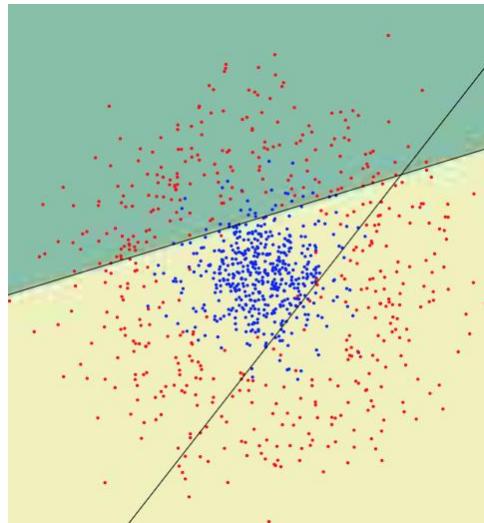
Бустинг

- Комбинирование «слабых» (weak) классификаторов
- Adaptive Boosting (AdaBoost) - каждый следующий («слабый») классификатор строится по объектам, которые плохо классифицируются предыдущими классификаторами.

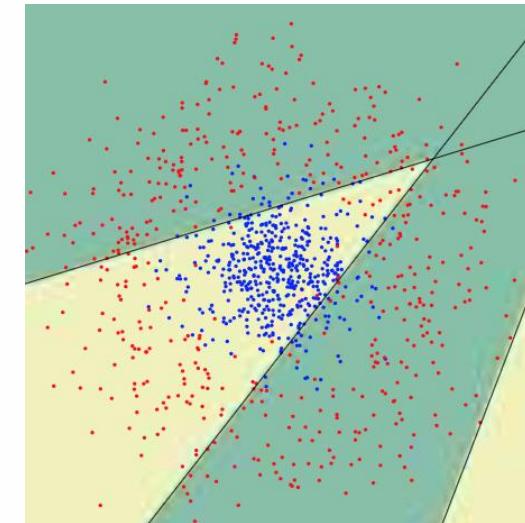
t=1



t=2



t=3



- Обычно в качестве «слабого» классификатора выбирается дерево решений высотой 1 (decision stump)
- Калибровка решений с помощью Platt scaling: Nikolaou et al. (2016). Cost-sensitive boosting algorithms: Do we really need them? Machine Learning

AdaBoost

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

- Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

- For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
- Choose the classifier, h_t , with the lowest error ϵ_t .
- Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

N=(10000 лиц + 10000 не-лиц)



...



...

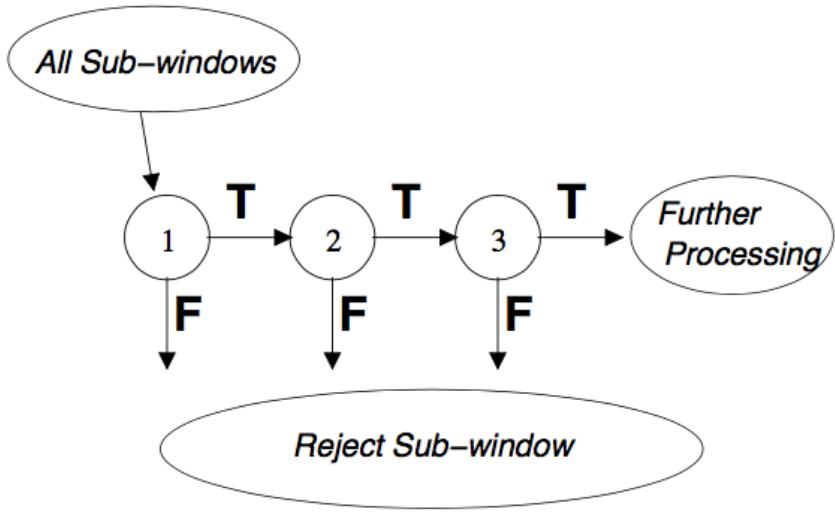


...

K=40000 фич

Шаг 2 выполняется за K*O(N) !!!

Каскад классификаторов

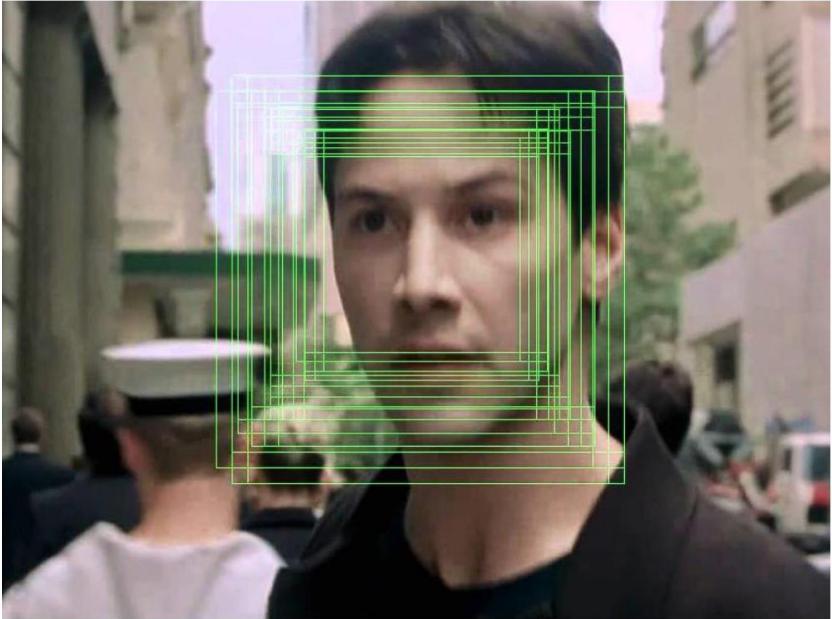


Признаки в первом каскаде



- Каждый классификатор F_i натренирован с ~ 1 (0.999) hit-rate и 0.5 false positive rate.
- Для тренировки каждого F_i отбираем только сэмплы (позитивные и негативные), которые прошли все предыдущие $F_1 \dots F_{i-1}$.
- Результирующий классификатор из 20 стадий очень точный: hit-rate $0.99920 \approx 0.98$ и false alarm rate $0.520 \approx 1E-6$!
- Очень быстрый: откидывает $\sim 90\%$ кандидатов на первых 3 стадиях
- Сложно набрать достаточно негативных примеров для тренировки каждой стадии классификатора

Группировка выходных прямоугольников

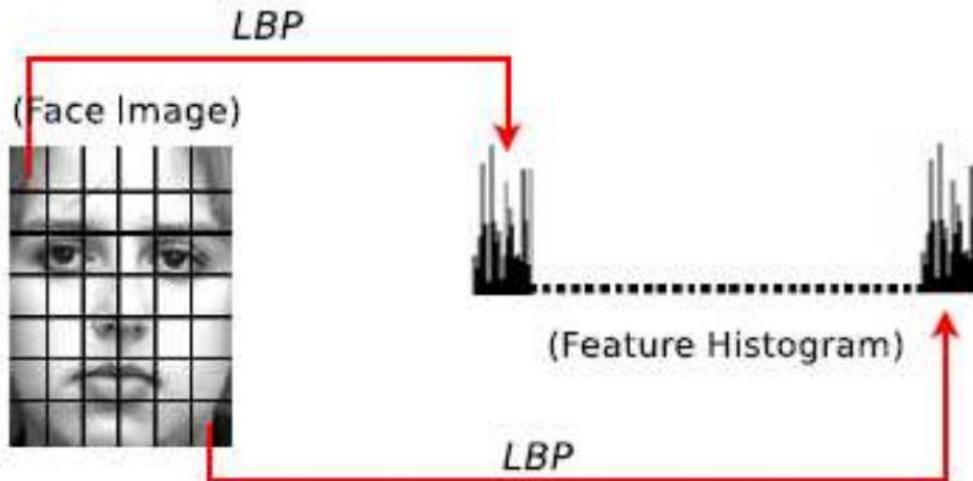


Коэффициент сходства Жаккара (Jaccard)

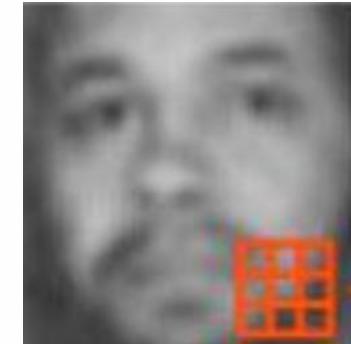
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$
A diagram illustrating the Jaccard similarity coefficient (IoU). It shows two overlapping blue rectangles. The top rectangle is smaller and centered within the larger bottom one. The area where they overlap is shaded dark blue, while the rest of both rectangles is light blue.

- А и В считаются близкими если $\text{JD}(A, B) \leq \text{eps}$ ($\text{eps} \approx 0.6..0.9$)
- Можно построить классы эквивалентности (связные компоненты в графе близких кандидатов). Из каждого класса сформировать один прямоугольник (median, avg, ...)
- Можно использовать non-maxima suppression: рассмотреть все пары близких прямоугольников (A,B), выбрать прямоугольник с большим значением confidence

LBP признаки



«Слабый» классификатор определяется признаком $f_j(x)$ – номером элемента гистограммы, порогом и операцией (меньше/больше)



- Все вычисления целочисленные
- Всего для окна 20x20 имеется ≈ 4000 LBP признаков вместо 40000 признаков Хаара
- Каскад на LBP признаках работает в несколько (до 10) раз быстрее Хаара, занимая на порядок меньше памяти
- Точность LBP по сравнению с признаками Хаара обычно ниже

Пример программного кода в OpenCV

Для повышения точности можно детектировать **область глаз** внутри лица

```
Mat img_gray;
cvtColor( matImage, img_gray, CV_BGR2GRAY );
equalizeHist(img_gray, img_gray);

std::vector<Rect> faces;
face_cascade.detectMultiScale(img_gray, faces,1.3, 2,
    CASCADE_DO_ROUGH_SEARCH|CASCADE_SCALE_IMAGE, Size(30, 30));

for(int i = 0; i < faces.size(); ++i ){
    Rect& r = faces[i];
    std::vector<Rect> eyes;
    cv::Mat faceROI = img_gray (r);
    eye_cascade.detectMultiScale(faceROI, eyes,1.1, 2,
        CASCADE_DO_ROUGH_SEARCH|CASCADE_SCALE_IMAGE, Size(30, 30));
    if(!eyes.empty()){
        //do something with face r
    }
}
```

HOG

- Haar/LBP признаки не подходят для разнотекстурных объектов
- Более точные результаты получаются с помощью HOG



HOG работают и для лиц!

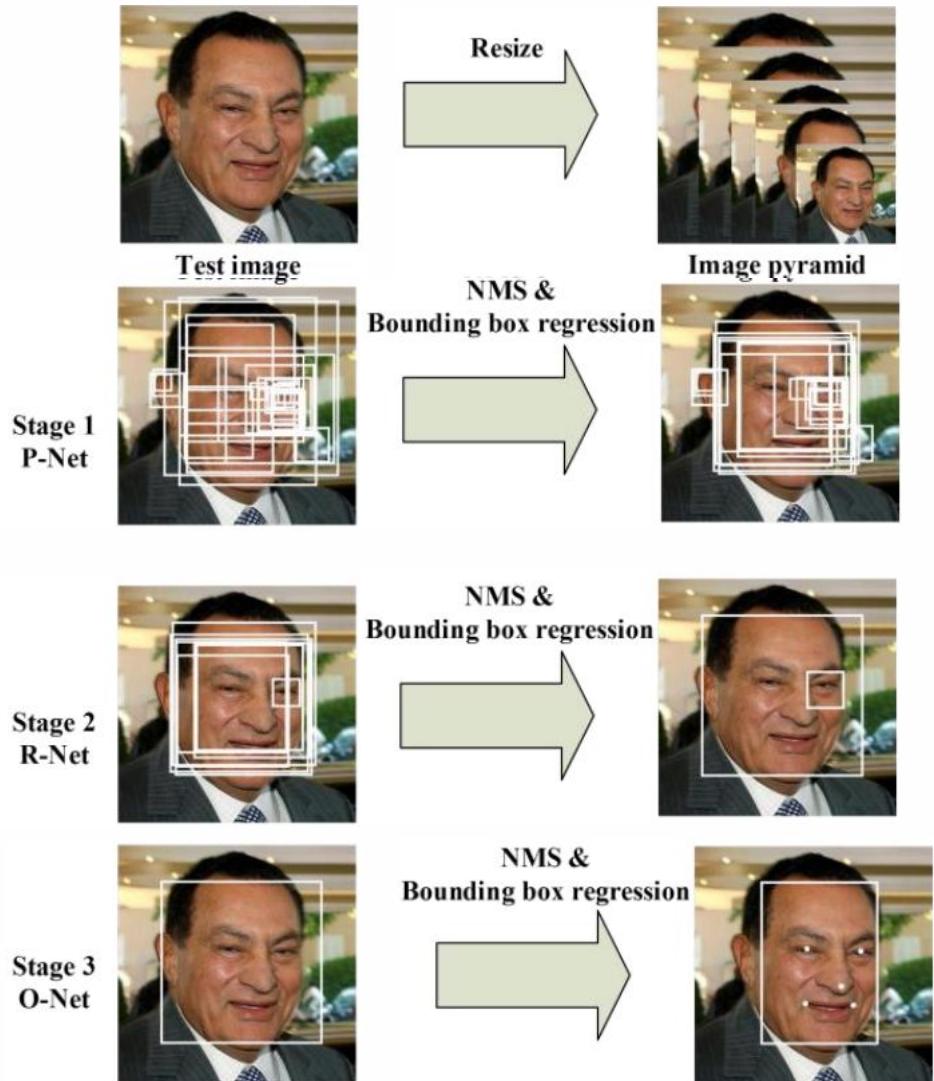


Детектор лиц в Dlib намного точнее
OpenCV, но в несколько раз медленнее

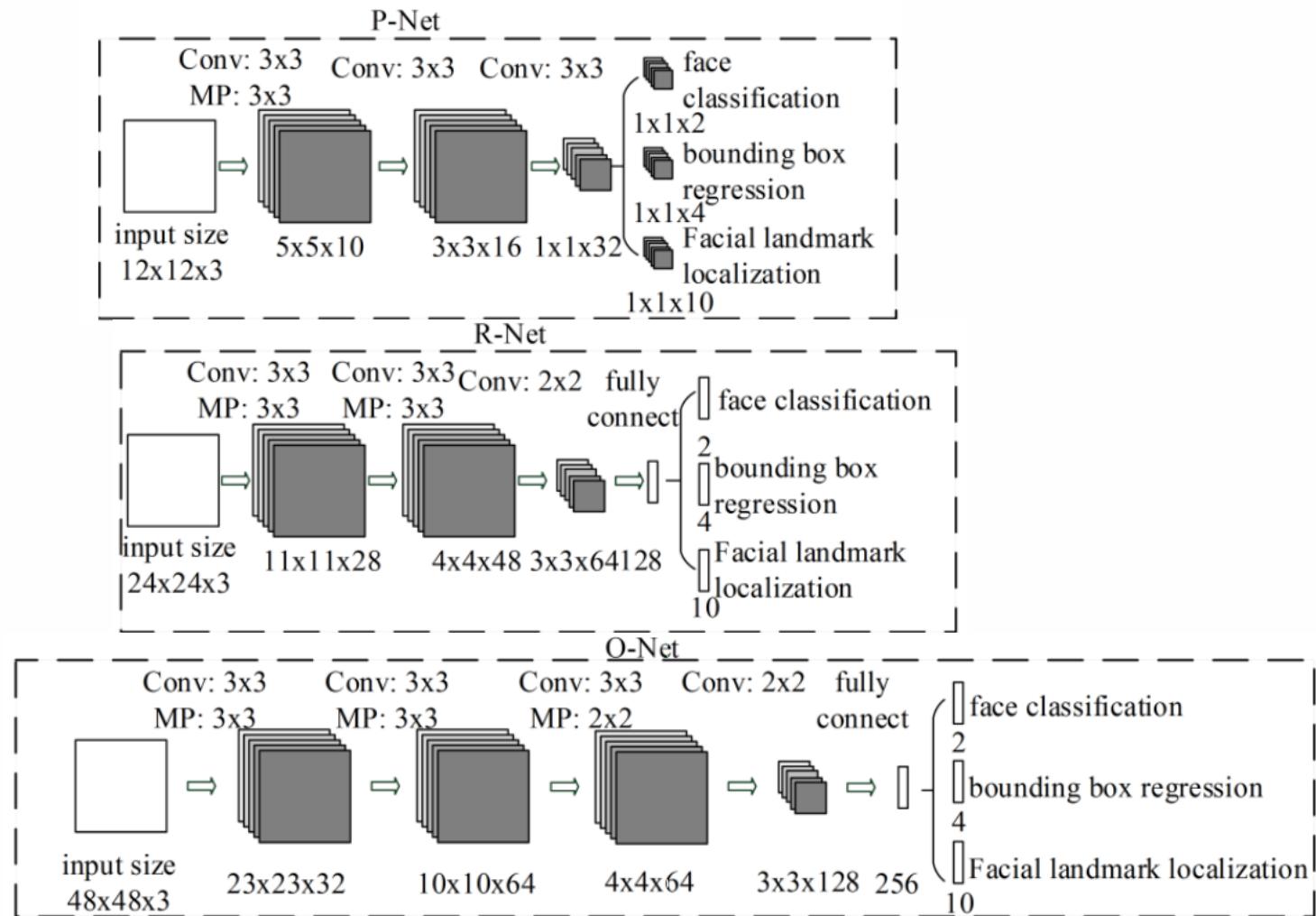


<https://www.youtube.com/watch?v=LsK0hzcEyHI>

MTCNN: multi-task CNN

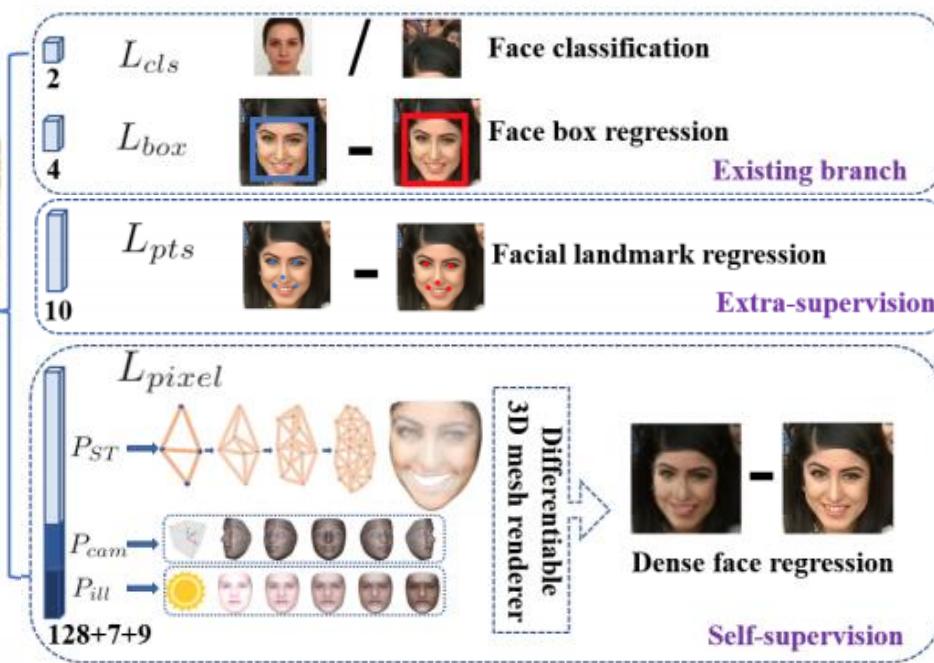


Три стадии (сети)

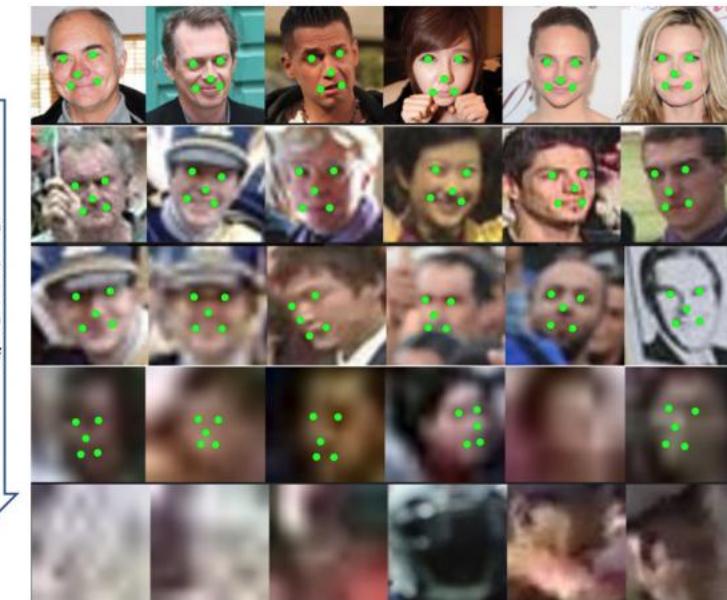


<https://arxiv.org/abs/1604.02878>

RetinaFace



Добавлены аннотации 5 лендмарков



$$\begin{aligned} L = & L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) \\ & + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel}. \end{aligned}$$

Распознавание лиц

Наборы данных для идентификации/верификации лиц

Набор данных LFW (Labeled Faces in the Wild)



CASIA WebFaces



MS-Celeb-1M



<http://arxiv.org/abs/1710.08092>

<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/08/MSCeleb-1M-a.pdf>

G. B. Huang et al, ECCV Workshop on Faces in Real-life Images, 2008

VGGFace2



(c) John Wesley Shipp



(e) Princess Haya Bint Al Hussein

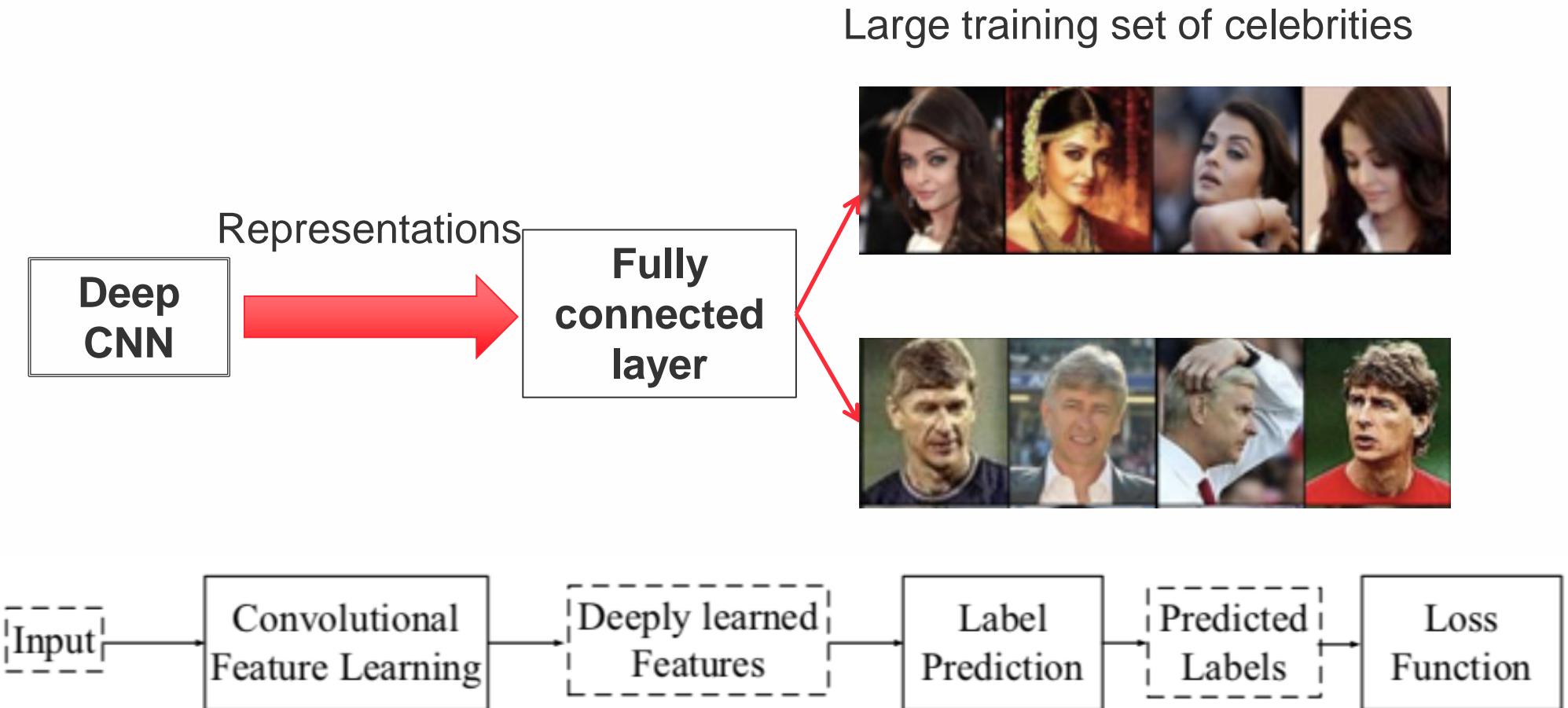


(g) Roy Jones Jr.



(i) Additi Gupta

Deep faces



ФУНКЦИИ ПОТЕРЬ

Softmax activations

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

Non-negative log-likelihood

$$L_1 = -\frac{1}{m} \sum_{i=1}^m \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}}$$

- VGGFace (VGG16) /VGGFace2 (ResNet/SeNet-50), <https://github.com/rcmalli/keras-vggface>
- FaceNet (InceptionResNet v1), <https://github.com/davidsandberg/facenet>
- LightCNN, <https://github.com/AlfredXiangWu/LightCNN>
- MobileNet v1, http://github.com/HSE-asavchenko/HSE_FaceRec_tf
- ...

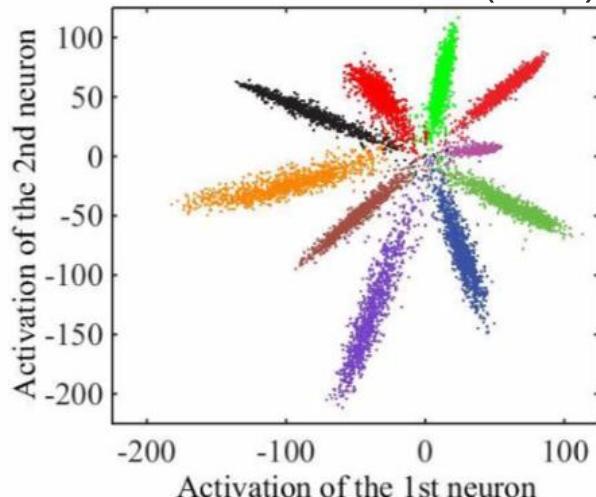
Регуляризация Center loss

Минимизируются вариации внутри класса

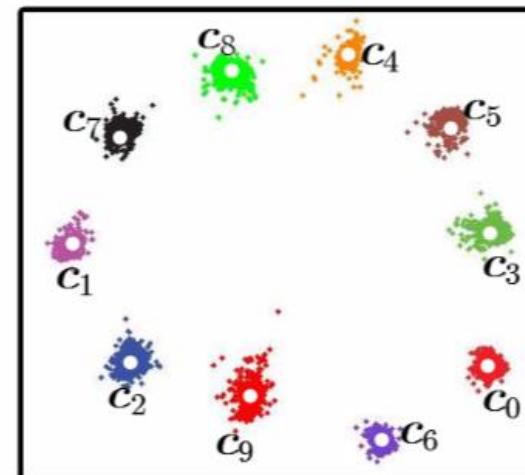
$$-\sum_{i=1}^m \log \frac{e^{W_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T \mathbf{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^m \|\mathbf{x}_i - \mathbf{c}_{y_i}\|_2^2$$

$$\frac{\partial \mathcal{L}_C}{\partial \mathbf{x}_i} = \mathbf{x}_i - \mathbf{c}_{y_i}$$
$$\Delta \mathbf{c}_j = \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (\mathbf{c}_j - \mathbf{x}_i)}{1 + \sum_{i=1}^m \delta(y_i = j)}$$

Обычный softmax ($\lambda=0$)



Center loss ($\lambda=1$)



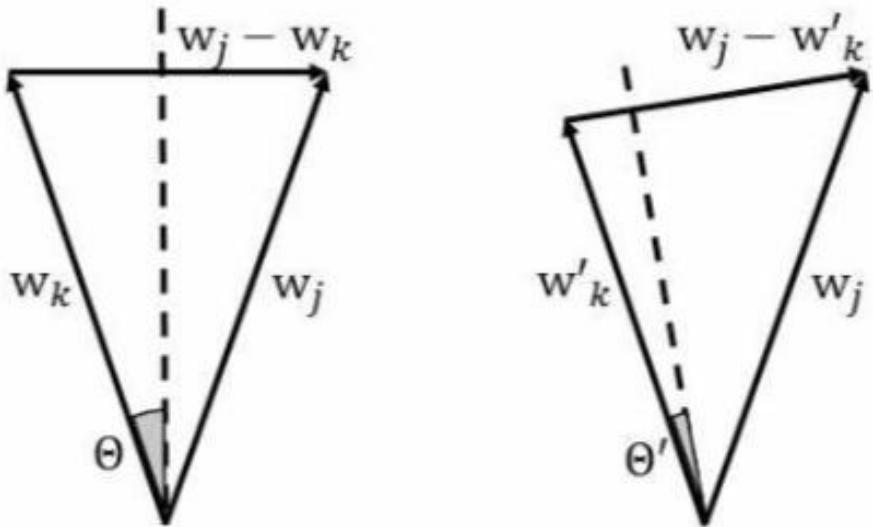
One-shot face recognition

Лица из нового набора данных C_n с малым числом фотографий-образцов и лица из базового набора C_b с большим числом примеров должны обладать примерно схожим разбросом (volume size)

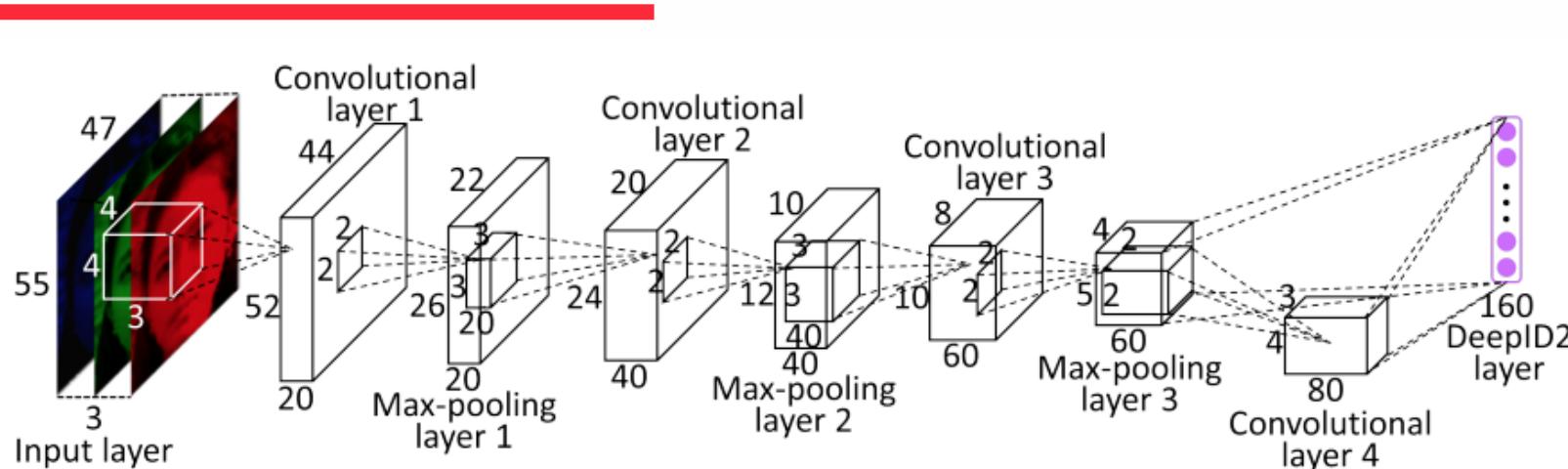
Связь между нормой вектора \mathbf{w}_k и разбросом (volume size) k -го класса

$$\mathcal{L}_{up} = \sum_n -t_{k,n} \log p_k(x_n) + \frac{1}{|C_n|} \sum_{k \in C_n} \| \|\mathbf{w}_k\|_2^2 - \alpha \|_2^2$$

$$\alpha = \frac{1}{|C_b|} \sum_{k \in C_b} \|\mathbf{w}_k\|_2^2$$



DeepID2



1. Фаза идентификации (softmax)

$$\text{Ident}(f, t, \theta_{id}) = - \sum_{i=1}^n -p_i \log \hat{p}_i = - \log \hat{p}_t$$

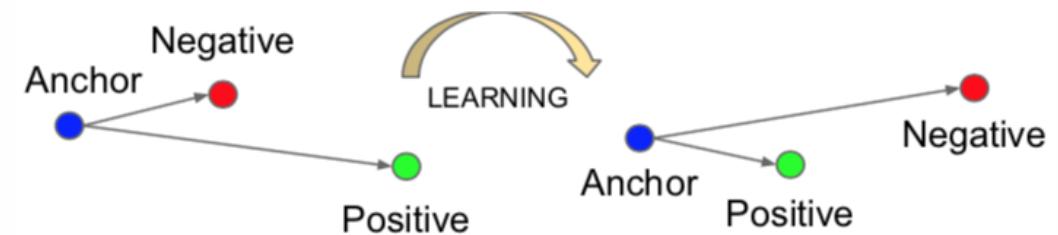
2. Фаза верификации (**Contrastive loss**)

$$\text{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \begin{cases} \frac{1}{2} \|f_i - f_j\|_2^2 & \text{if } y_{ij} = 1 \\ \frac{1}{2} \max(0, m - \|f_i - f_j\|_2)^2 & \text{if } y_{ij} = -1 \end{cases}$$

Triplet loss

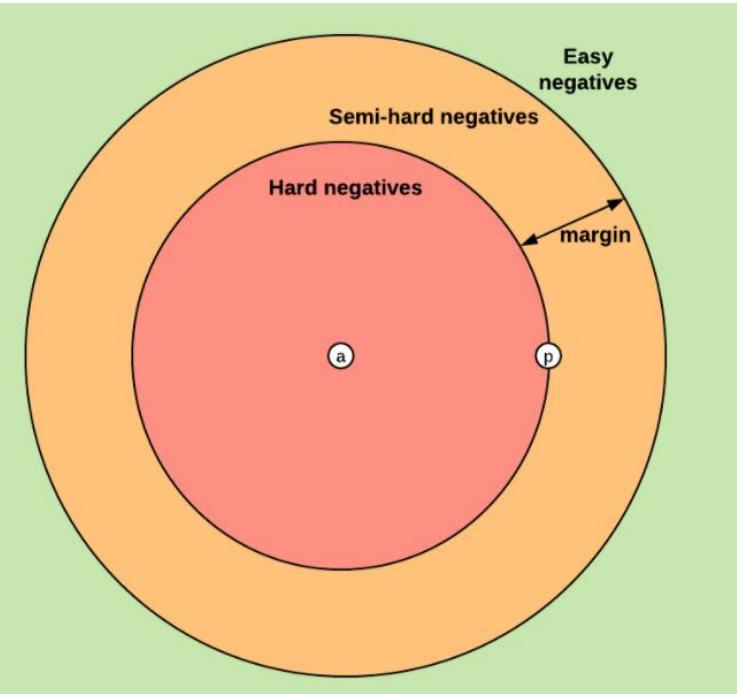
Изображение лица (anchor, a) должно быть ближе ко всем остальным фотографиям лица этого человека (positive, p), чем к изображениям других лиц (negative, n)

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$



Выбор триплетов

1. **easy triplets:** $d(a,p)+\text{margin} < d(a,n)$
2. **hard triplets:** $d(a,n) < d(a,p)$
3. **semi-hard triplets:** $d(a,p) < d(a,n) < d(a,p)+\text{margin}$



Нормализация

- FaceNet (InceptionResNet), <https://github.com/davidsandberg/facenet>

Нормализация входного изображения	LFW Rank-1 accuracy (%)
Prewiten: $(\text{image}-\text{mean}(\text{image}))/\max(\text{std}(\text{image}), \epsilon)$	81.6%
Fixed image standardization: $(\text{image}-127.5)/128$	95.5%

- Идентификация лиц на видео, IJB-A dataset (Sokolova et al 2017)

	Lightened CNN			VGGNet		
	AUC (%)	ERR (%)	FRR@FAR = 1%	AUC (%)	ERR (%)	FRR@FAR = 1%
L ₂ -norm -> Distance (1)	87.9 ± 0.5	20.5 ± 0.9	67.9 ± 3.3	97.5 ± 0.4	8.0 ± 0.4	30.3 ± 4.6
Medoid (2) -> L ₂ -norm	76.6 ± 0.4	30.0 ± 1.2	77.1 ± 4.0	92.4 ± 0.7	15.5 ± 0.6	50.0 ± 6.9
L ₂ -norm -> AvePool (3)	79.6 ± 0.7	27.8 ± 0.8	67.5 ± 4.4	96.1 ± 0.4	13.6 ± 0.8	40.0 ± 4.4
AvePool (3) -> L ₂ -norm	88.2 ± 0.4	20.0 ± 0.4	59.3 ± 2.9	97.7 ± 0.3	8.0 ± 0.3	26.8 ± 4.2

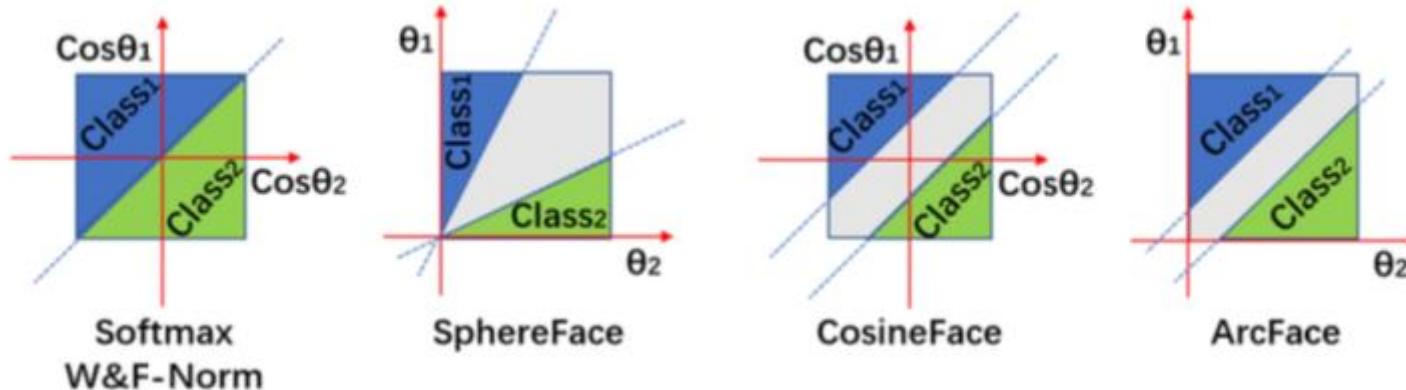
Angular space

- **Sphere loss** [Liu et al, CVPR 2017]
- **Cosine loss in FaceCNN** [Wang et al, CVPR 2018]
- **ArcFace** [Deng et al, 2018], <https://github.com/AllnAi/tf-insightface>

Пример: бинарная классификация

Loss Functions	Decision Boundaries
Softmax	$(W_1 - W_2)x + b_1 - b_2 = 0$
W-Norm Softmax	$\ x\ (\cos \theta_1 - \cos \theta_2) = 0$
SphereFace [23]	$\ x\ (\cos m\theta_1 - \cos \theta_2) = 0$
F-Norm SphereFace	$s(\cos m\theta_1 - \cos \theta_2) = 0$
CosineFace [44, 43]	$s(\cos \theta_1 - m - \cos \theta_2) = 0$
ArcFace	$s(\cos(\theta_1 + m) - \cos \theta_2) = 0$

- θ_i – угол между W_i и x .
- Параметр $m > 1$ в SphereFace требует, чтобы $\cos(m\theta_1) > \cos(\theta_2)$
- m – запас (margin) в ArcFace/CosineFace

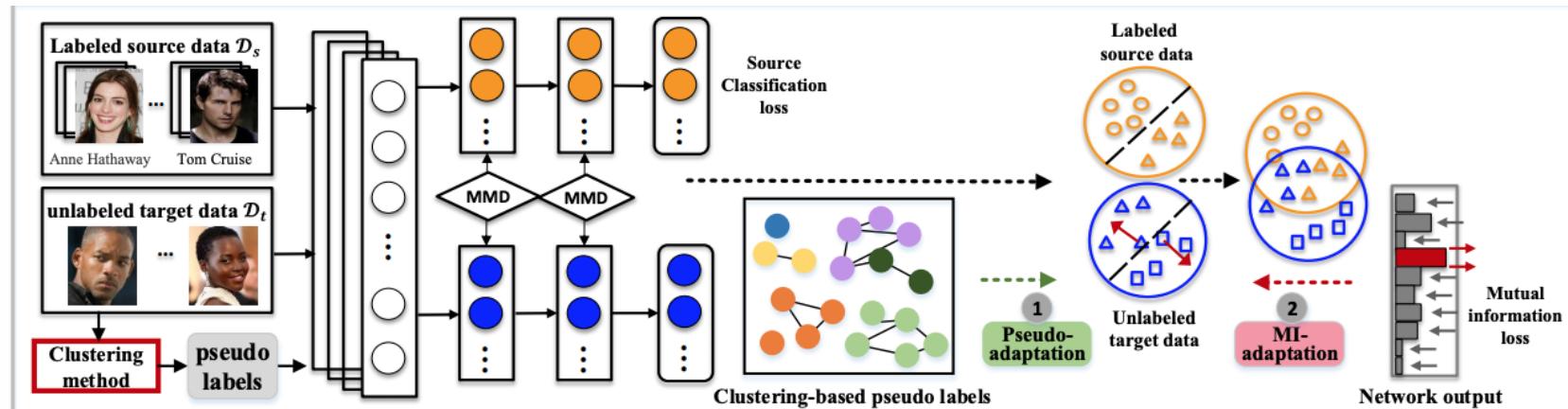


Racial bias

Набор данных RFW



Information maximization adaptation network (IMAN)



Обучение без учителя:

1. Кластеризация набора данных и обучение сети с softmax-loss на «надежных» кластерах (с высокой cosine similarity)
2. Функция потерь: выход классификатора должен быть близким к one-hot encoding, но при этом все кластеры равномерно представлены
3. Адаптация сети, обученной на другом наборе данных: MMD (Maximum mean Discrepancy)

$$\mathcal{L} = \mathcal{L}_C(X_s, Y_s) + \alpha \sum_{l \in \mathcal{L}} MMD^2(D_s^l, D_t^l) + \beta \mathcal{L}_M(X_t)$$

<https://arxiv.org/pdf/1812.00194.pdf>

	Model	RFW			
		Caucasian	Indian	Asian	African
commercial API	Microsoft [5]	87.60	82.83	79.67	75.83
	Face++ [4]	93.90	88.55	92.47	87.50
	Baidu [3]	89.13	86.53	90.27	77.97
	Amazon [1]	90.45	87.20	84.87	86.27
	mean	90.27	86.28	86.82	81.89
SOTA algorithm	Center-loss [65]	87.18	81.92	79.32	78.00
	Sphereface [39]	90.80	87.02	82.95	82.28
	Arcface ¹ [21]	92.15	88.00	83.98	84.93
	VGGface2 [15]	89.90	86.13	84.93	83.38
	mean	90.01	85.77	82.80	82.15

Метод максимально правдоподобных рассогласований

$$f(\rho_1(\mathbf{x}), \dots, \rho_c(\mathbf{x})|W_c) = f(\rho_c(\mathbf{x})|W_c) \cdot \prod_{i=1, i \neq c}^c f(\rho_i(\mathbf{x})|W_c)$$

Рассогласование Кульбака-Лейблера асимптотически распределено как нецентральный хи-квадрат. Используется аппроксимация нормальным распределением

$$\mathcal{N}\left(\rho_{KL}(\mathbf{x}_r, \mathbf{x}_i) + \frac{D-1}{WH}, \frac{4WH\rho_{KL}(\mathbf{x}_r, \mathbf{x}_i) + (D-1)}{2(WH)^2}\right)$$



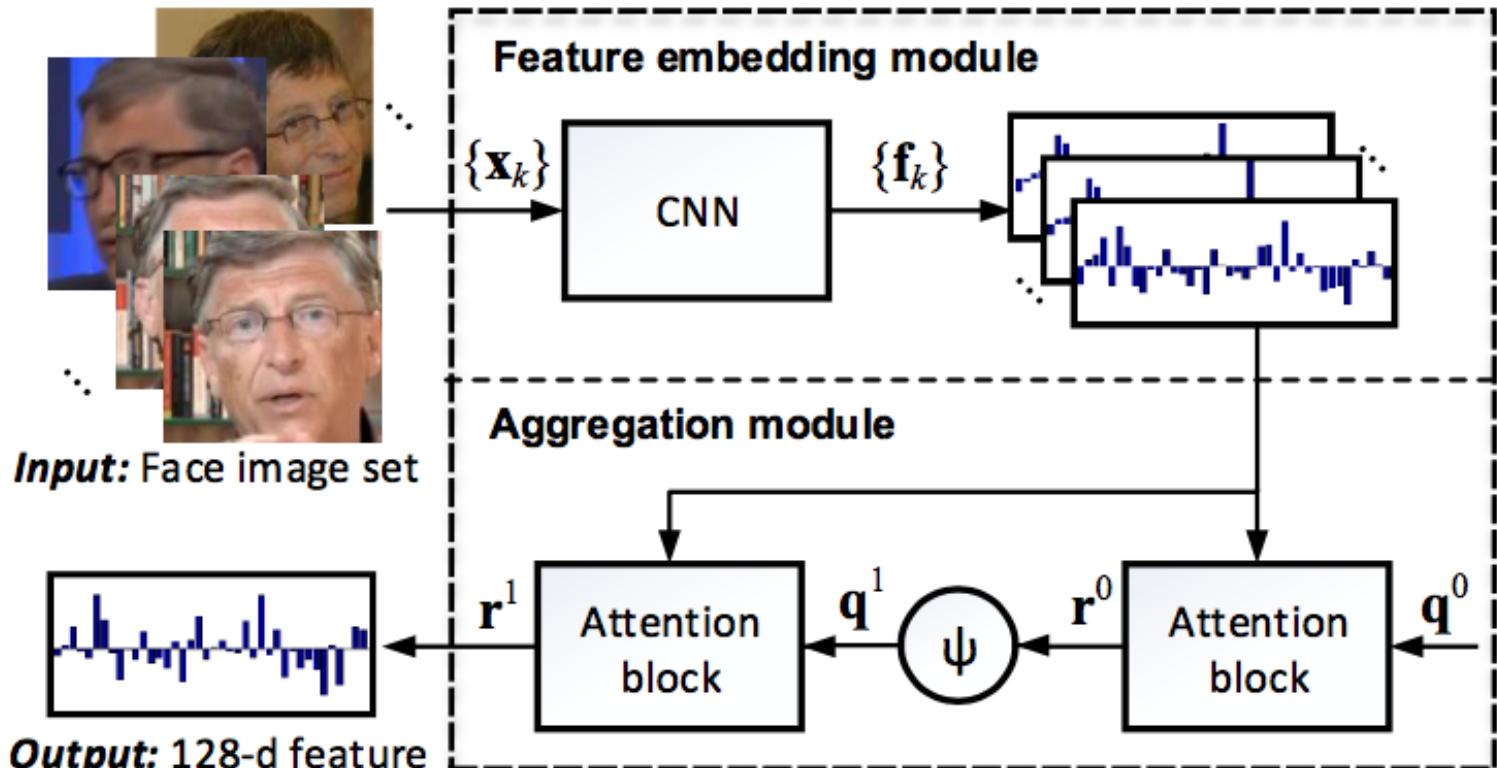
Ближайший сосед с регуляризацией

$$\min_{c \in \{c_1, \dots, c_M\}} \left(\rho_c(\mathbf{x}) + \frac{\lambda}{C} \sum_{i=1, i \neq c}^c \frac{(\rho_i(\mathbf{x}) - \rho_{c;i} - \frac{D-1}{WH})^2}{\rho_{c;i} + \frac{D-1}{4WH}} \right)$$



SubjectId	L_2 distance	Regularization term	Total objective
1001	0.0236	0.0152	0.0247
1303	0.0238	0.0152	0.0249
314	0.0239	0.0113	0.0247
1980	0.0240	0.0166	0.0252
387	0.0241	0.0300	0.0262
1289	0.0241	0.0140	0.0251
942	0.0242	0.0059	0.0246
1287	0.0242	0.0174	0.0258

Распознавание лиц на видео



Несколько последовательных блоков attention

$$q^1 = \tanh(Wr^0 + b)$$

Блок Attention

$$e_k = \mathbf{q}^T \mathbf{f}_k$$

$$a_k = \frac{\exp(e_k)}{\sum_j \exp(e_j)}$$

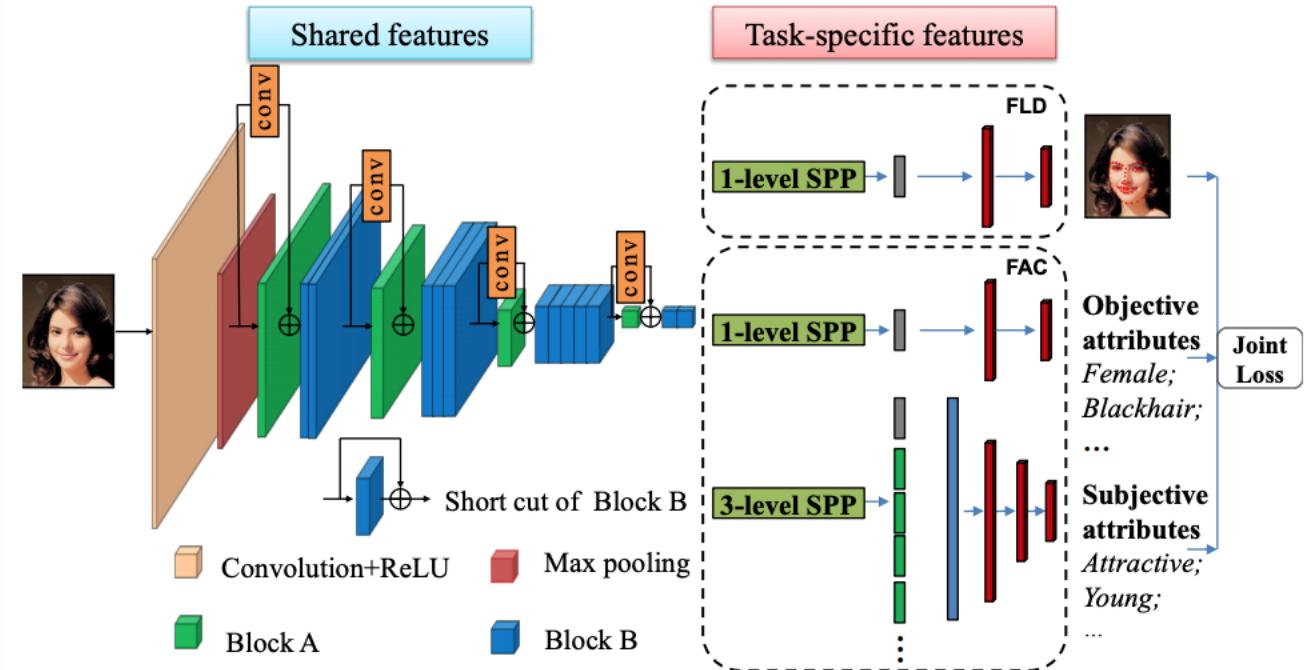
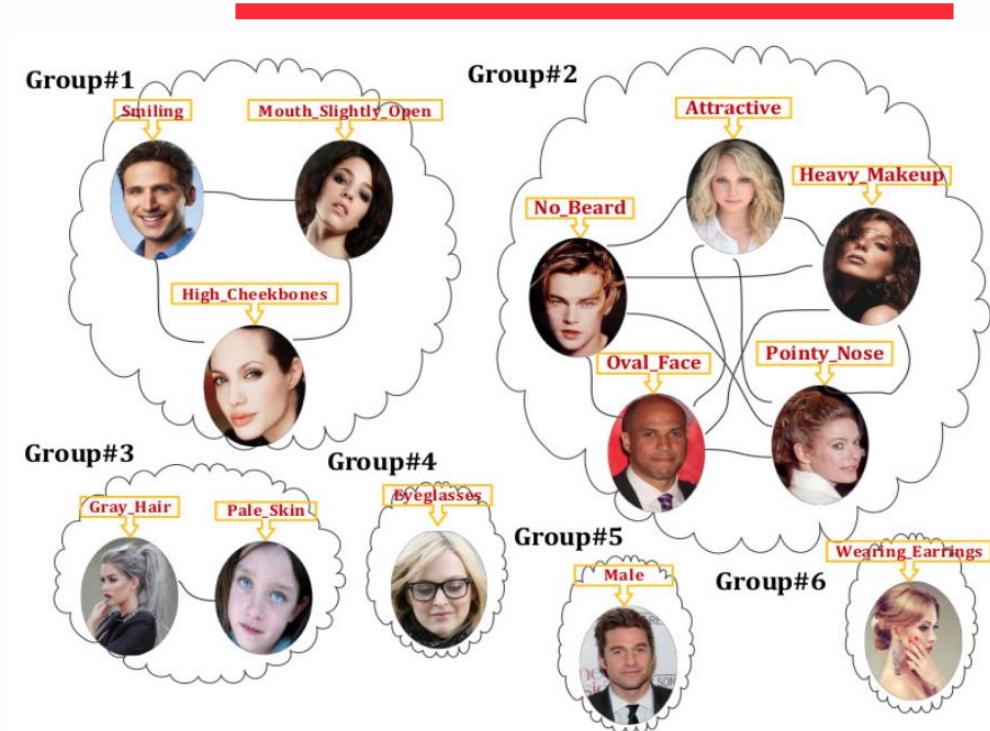


Агрегация
признаков кадров

$$\mathbf{r} = \sum_k a_k \mathbf{f}_k$$

Распознавание атрибутов лиц

Атрибуты лиц. Набор данных CelebA



FLD (facial landmark detection)
 FAC – facial attribute (multi-class) classification

$$L = \sum_{j=1}^J \lambda_t^j L_{FAC}^j + \beta L_{FLD}$$

$$L_{FLD} = \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i^{FLD} - \mathbf{y}_i^{FLD}\|^2$$

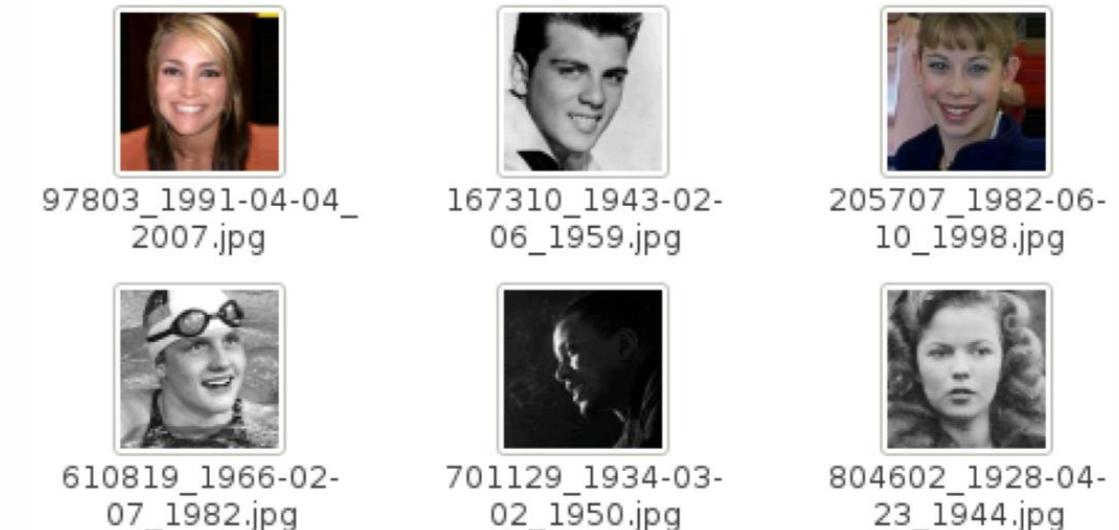
$$L_{FAC}^j = \frac{1}{N} \sum_{i=1}^N (\hat{y}_{i,j}^{FAC} - y_{i,j}^{FAC})^2$$

Пол и возраст. Наборы данных Adience и IMDB/Wiki

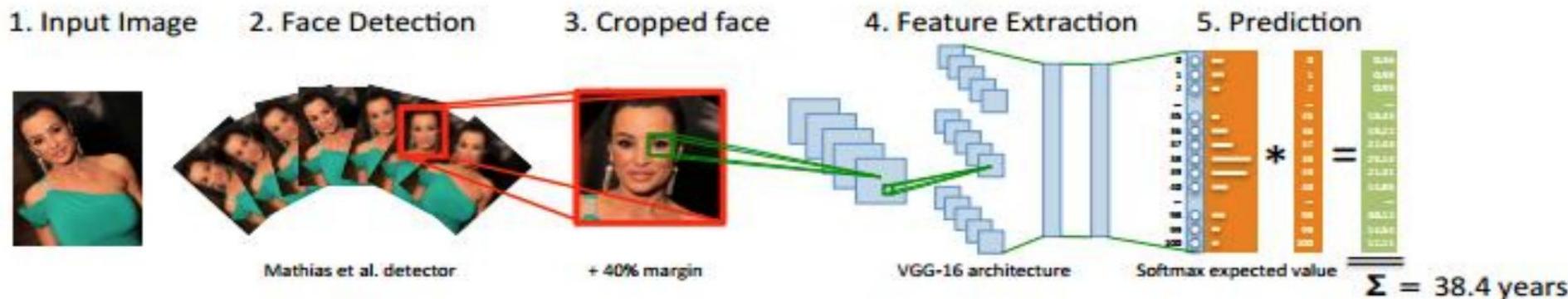
Набор данных Adience



Набор данных IMDB/Wiki

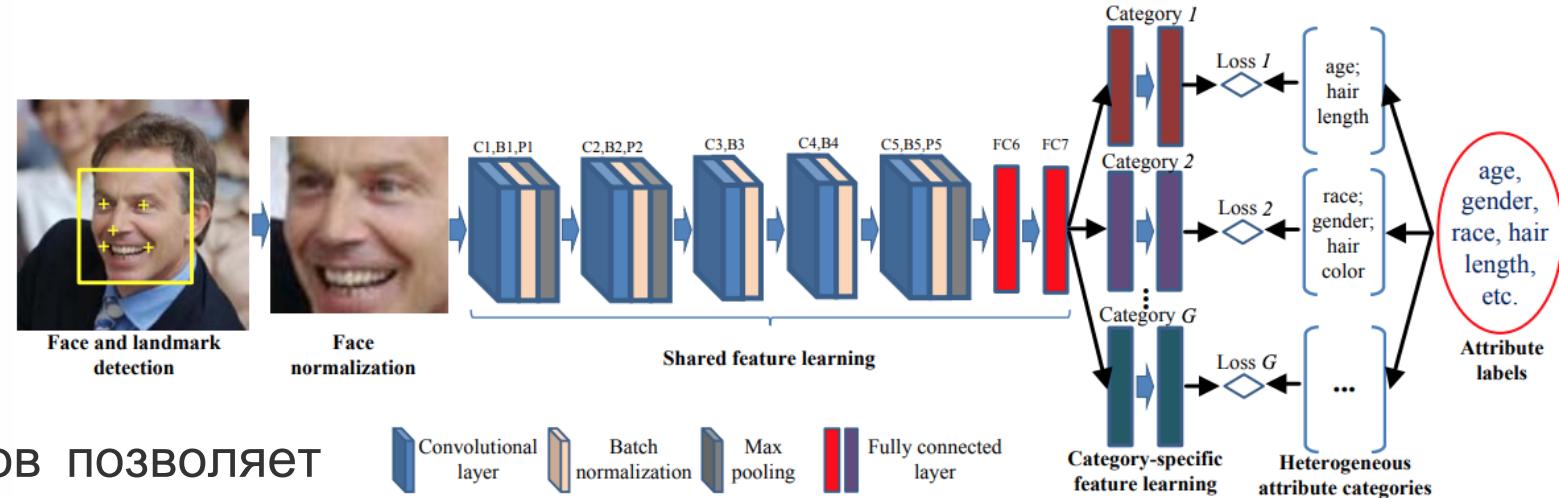


DEX: Deep EXpectation of apparent age from a single image, 2015 (VGG16-Net)

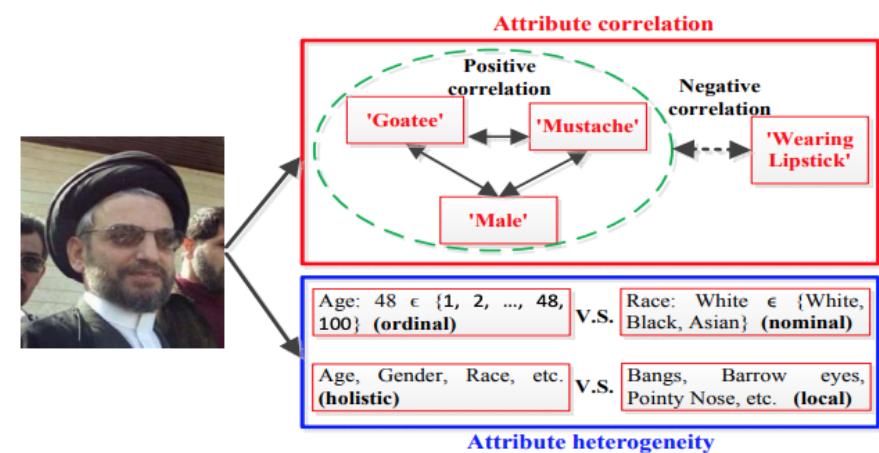
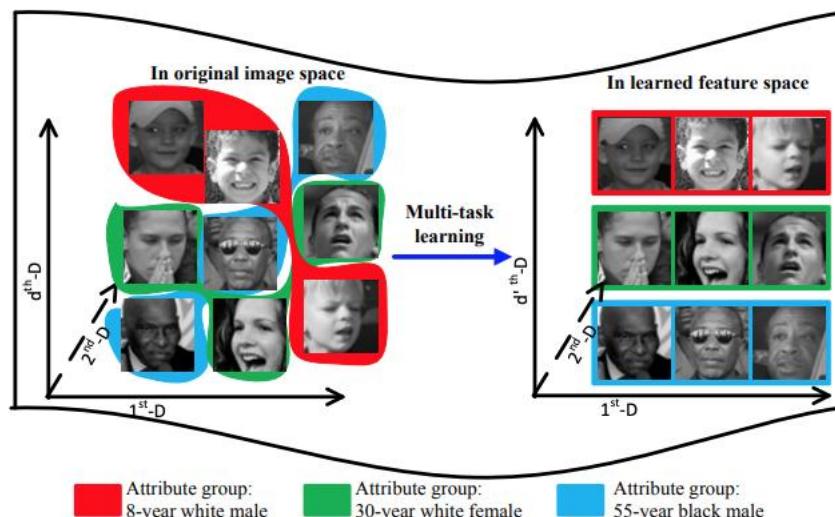


Распознавание атрибутов лиц: multi-task CNN

$$\arg \min_{W_c, \{W^j\}_{j=1}^M} \sum_{j=1}^M \sum_{i=1}^N \mathcal{L}(y_i^j, \mathcal{F}(X_i, W^j \circ W_c)), \\ + \gamma_1 \Phi(W_c) + \gamma_2 \Phi(W^j)$$



Использование общих дескрипторов позволяет лучше разделить классы разных атрибутов



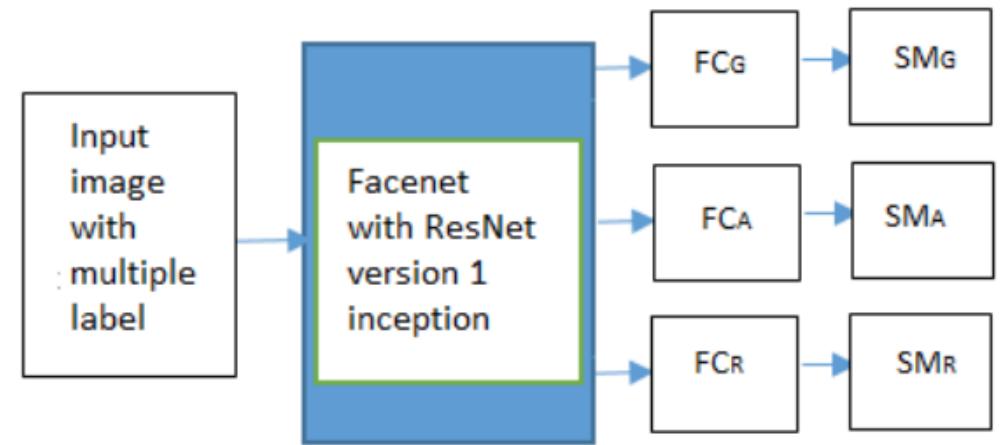
Распознавание пола, возраста и расы

Набор данных UTKFace

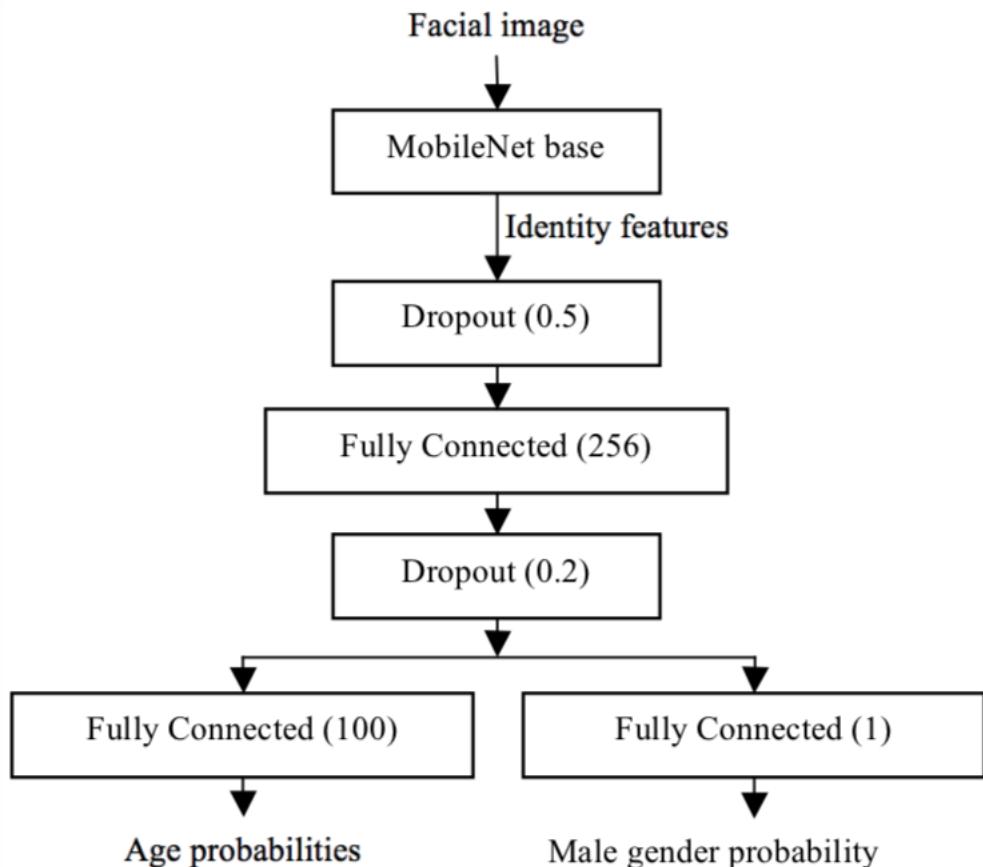
(<https://susanqq.github.io/UTKFace/>)



Multi-task CNN



Multi-task Mobile CNN



1. Обучить базовую сеть задаче идентификации (softmax loss, VGGFace2 dataset)
2. Обучить выходы для предсказания пола и возраста (alternating batches)

Предсказание возраста

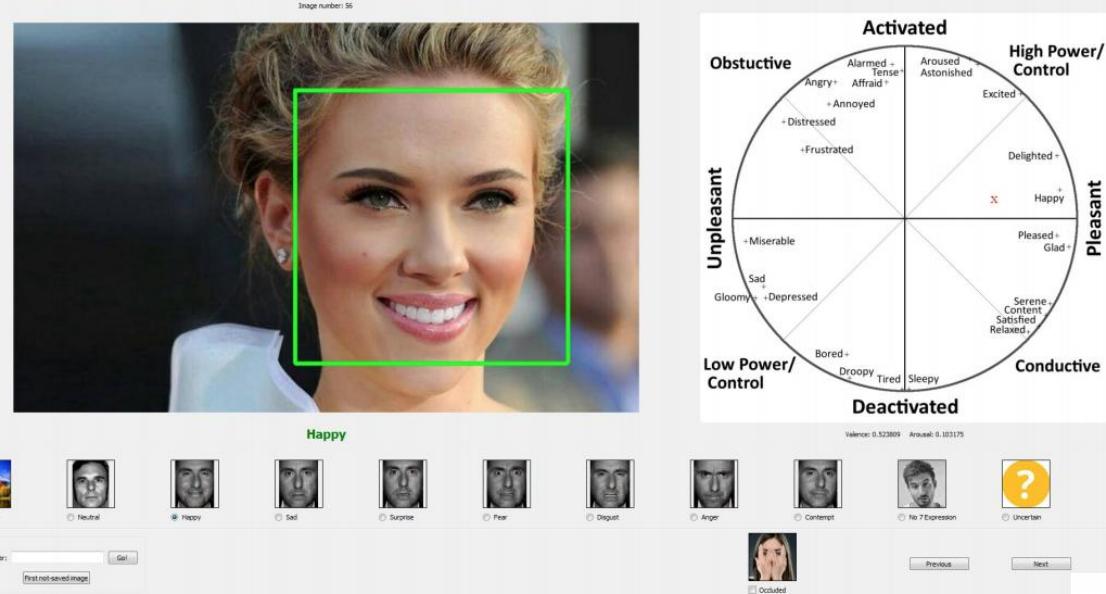
$$l^* = \frac{\sum_{k=1}^K l_{(k)} \cdot P(l_{(k)} | X(t))}{\sum_{k=1}^K P(l_{(k)} | X(t))}.$$

$$P(l_{(1)} | X(t)) \geq P(l_{(2)} | X(t)) \geq \dots \geq P(l_{(L)} | X(t))$$

- Savchenko, PeerJ-CS 2019
- https://github.com/HSE-asavchenko/HSE_FaceRec_tf/tree/master/age_gender_identity

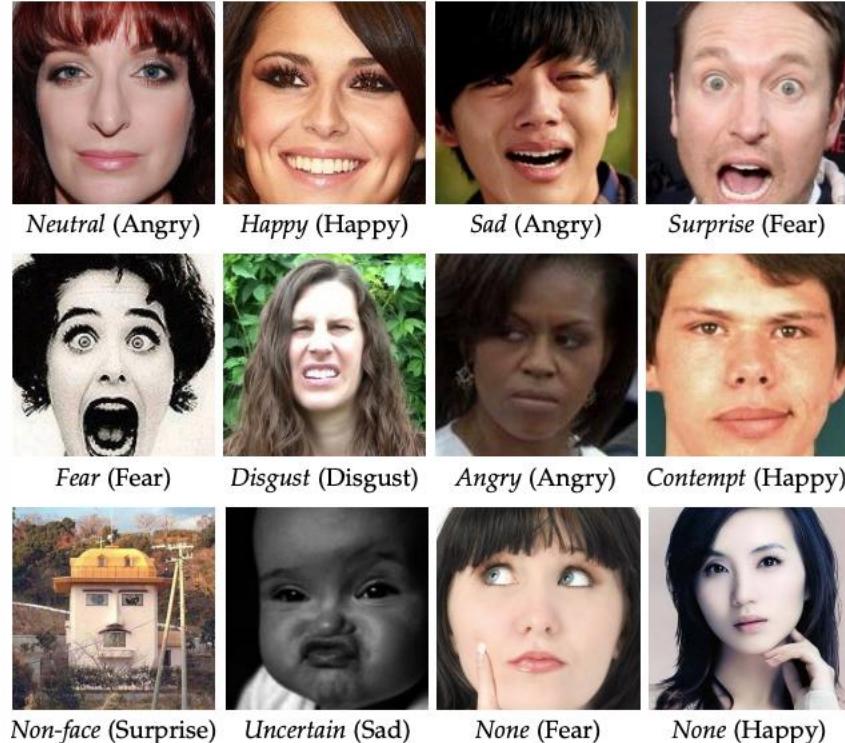
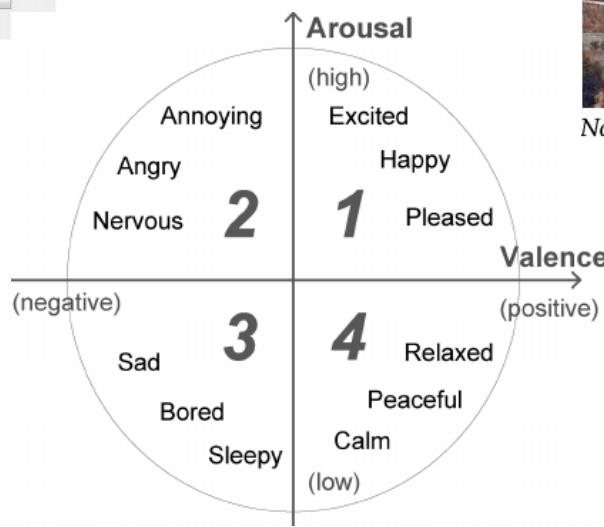
Распознавание эмоций

Набор данных AffectNet



Модель Рассела

«**Valence (знак)** refers to how positive or negative an event is, and **Arousal (интенсивность)** reflects whether an event is exciting/agitating or calm/soothing»

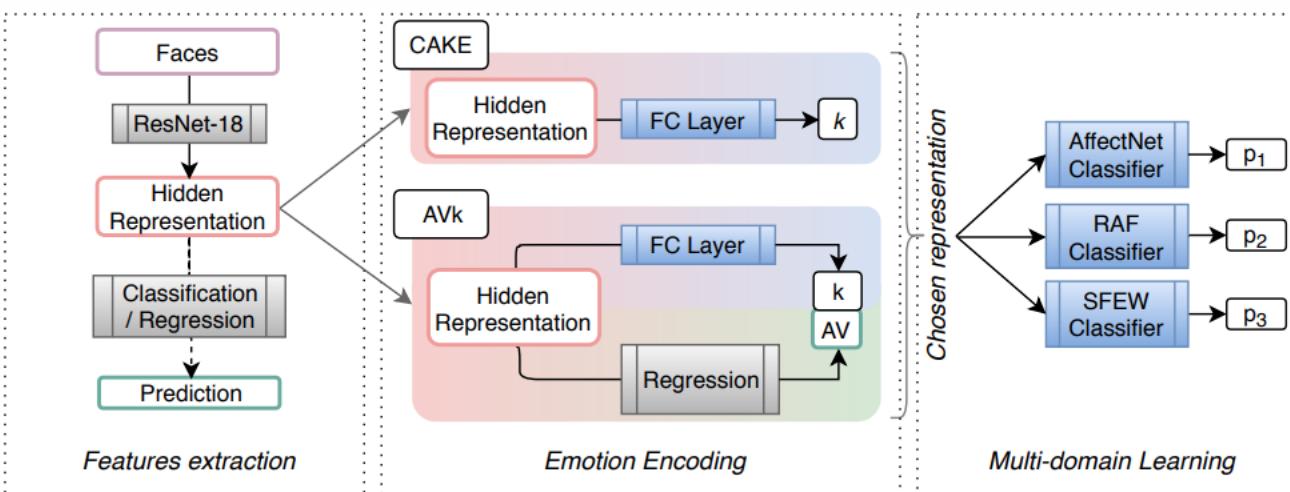


<http://mohammadmahoor.com/affectnet/>

Mollahosseini A. et al, IEEE Transactions on Affective Computing, 2017

Распознавание эмоций на статических изображениях

CAKE: Compact and Accurate K-dimensional representation of Emotion



Algorithm 1: Packing and Expanding (PAE)

Input: given task 1 and an original model trained on task 1.

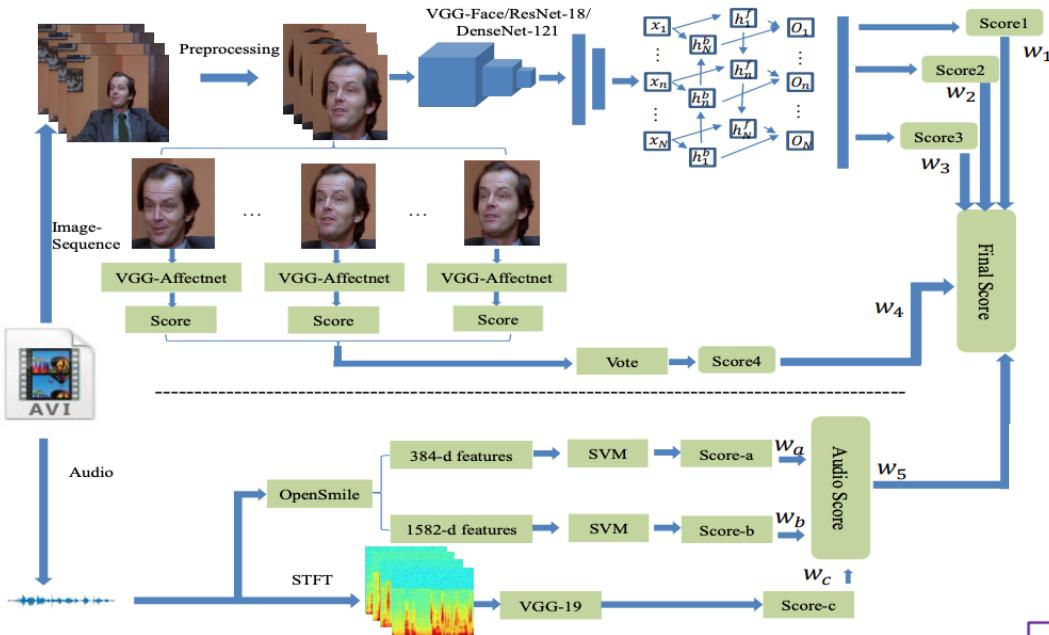
- 1 Set an accuracy goal for task 1;
- 2 Alternately remove small weights and re-train the remaining weights for task 1 via iterative pruning [29], until meeting the accuracy goal;
- 3 Let the model weights reserved for task 1 be W_1 (referred to as task-1 weights), and those that are removed by the iterative pruning be W_1^r (referred to as the saved weights);
- 4 **for** $task i = 2 \cdots K$ (*let the saved weights of task i be W_{i-1}^r*) **do**
- 5 Set an accuracy goal for task i ;
- 6 Use the weights W_1 and W_{i-1}^r to train task i , with W_1 fixed;
- 7 If the accuracy goal is not achieved by the trained model, expand the number of filters (weights) in the model, and reset $W_{i-1}^r \leftarrow W_{i-1}^r \cup W_E$, where W_E denotes the expanded weights;
- 8 Alternately remove small weights from W_{i-1}^r and re-train the remaining weights (with W_1 fixed) for task i via iterative pruning, until meeting the accuracy goal;
- 9 **end**

Multiple tasks:

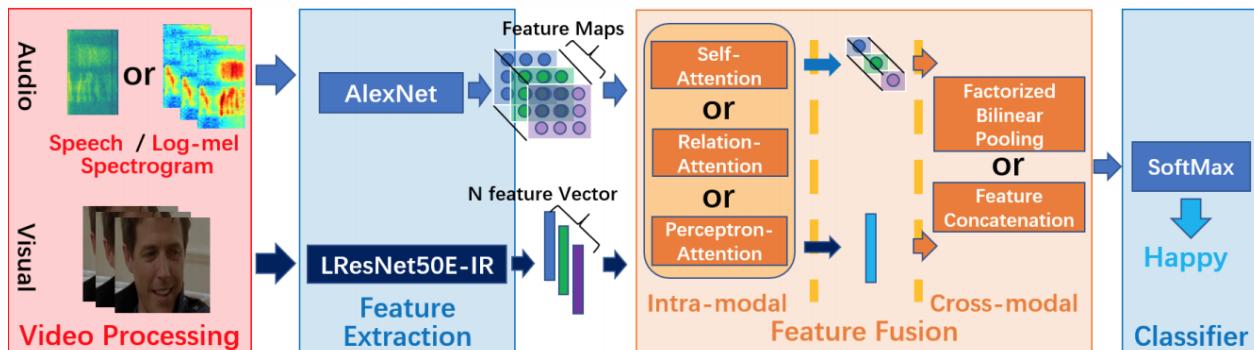
- face verification (99.67% on LFW)
- age prediction (57.30% on Adience)
- gender classification (92.23% on Adience)
- emotion recognition (65.29% on AffectNet)

Аудиовизуальное распознавание эмоций. EmotiW Challenge (AFEW)

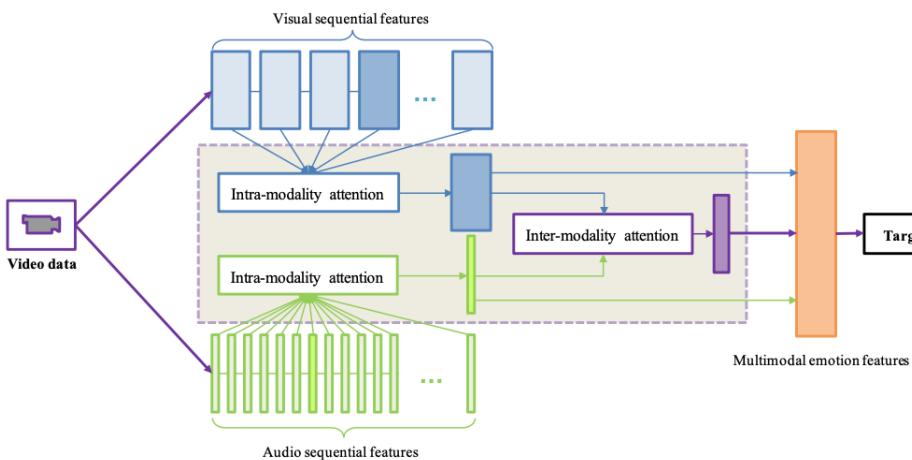
Bi-modality Fusion for Emotion Recognition in the Wild



Exploring Emotion Features and Fusion Strategies for Audio-Video Emotion Recognition



Multi-Attention Fusion Network



$$H_n^T = BLSTM(BLSTM(X_n^T))$$

$$A_n^T = \text{softmax}(W_{s2} \text{relu}(W_{s1} H_n^T))$$

$$M_n = A_n^T H_n^T$$

$$\text{Inter_}A = \text{softmax}(W_{m2} \text{relu}(W_{m1} Z))$$

$$\text{Inter_}M = \text{Inter_}A * Z$$

- A. Dhall et al, IEEE Multimedia 2012
- B. Sunan Li et al, ICMI 2019
- C. Hengshun Zhou et al, ICMI 2019
- D. Wang et al, ICMI 2019

Распознавание эмоций группы лиц

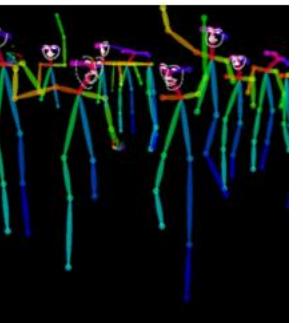
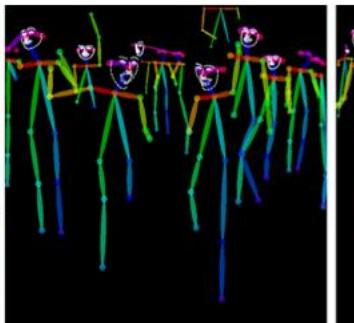
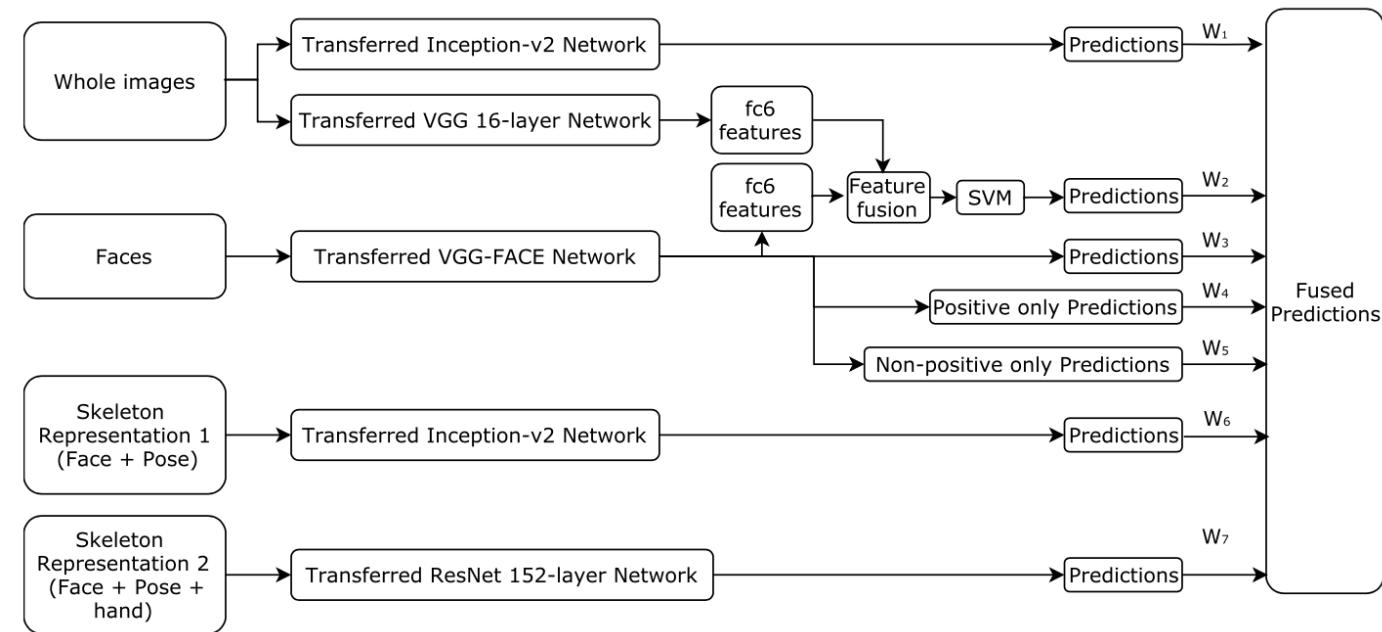
“positive”



“neutral”



“negative”



Dhall et al, ICMI 2017
Guo et al ICMI 2017

Cohesion prediction



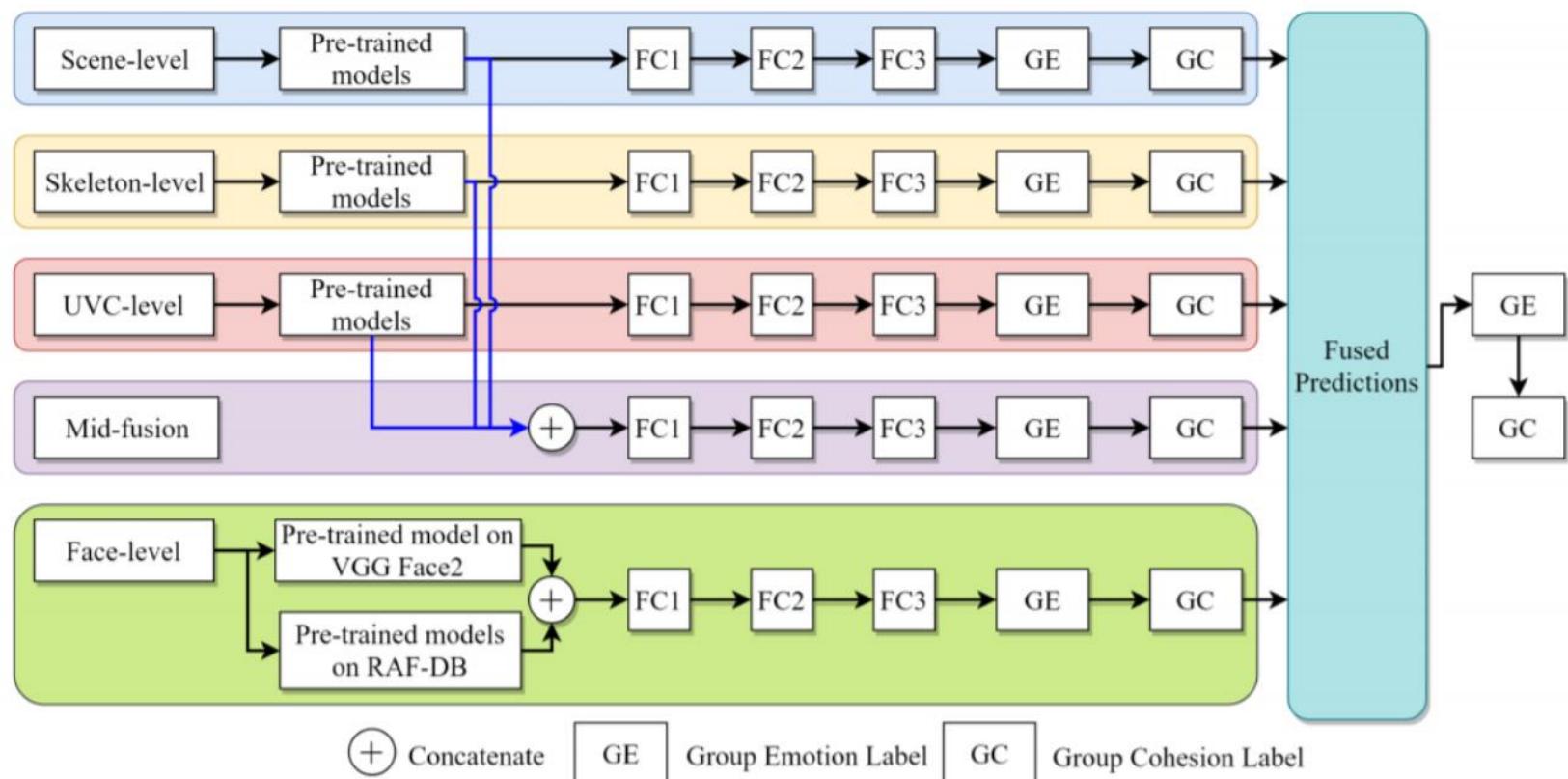
(a) Strongly disagree

(b) Disagree

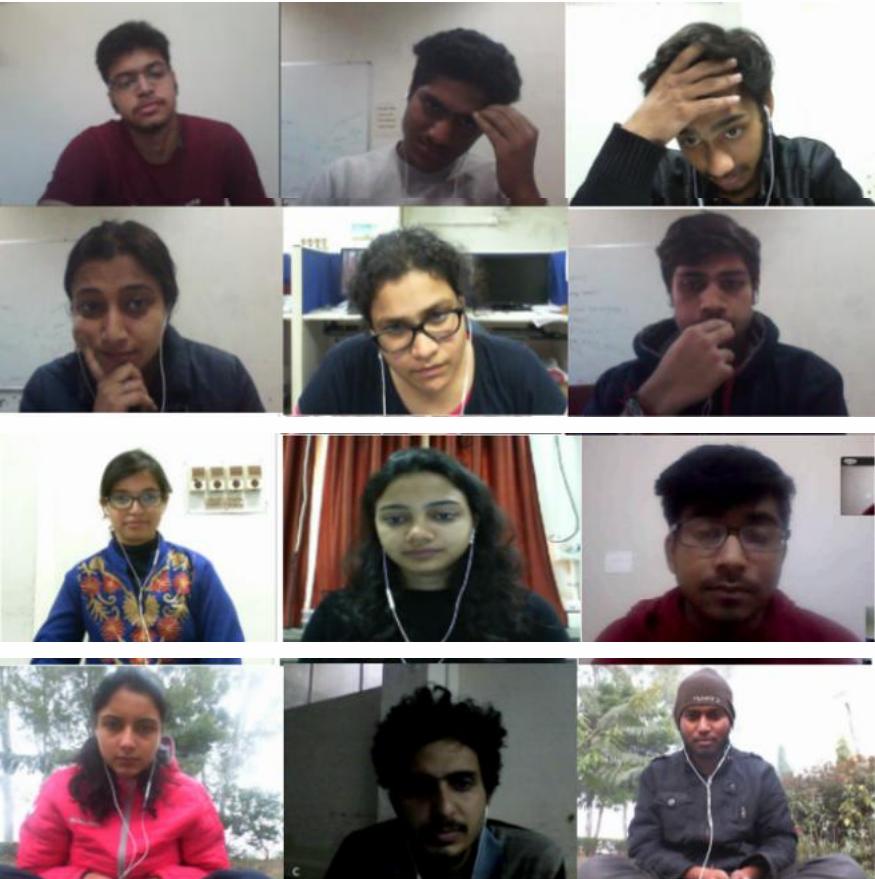


(c) Agree

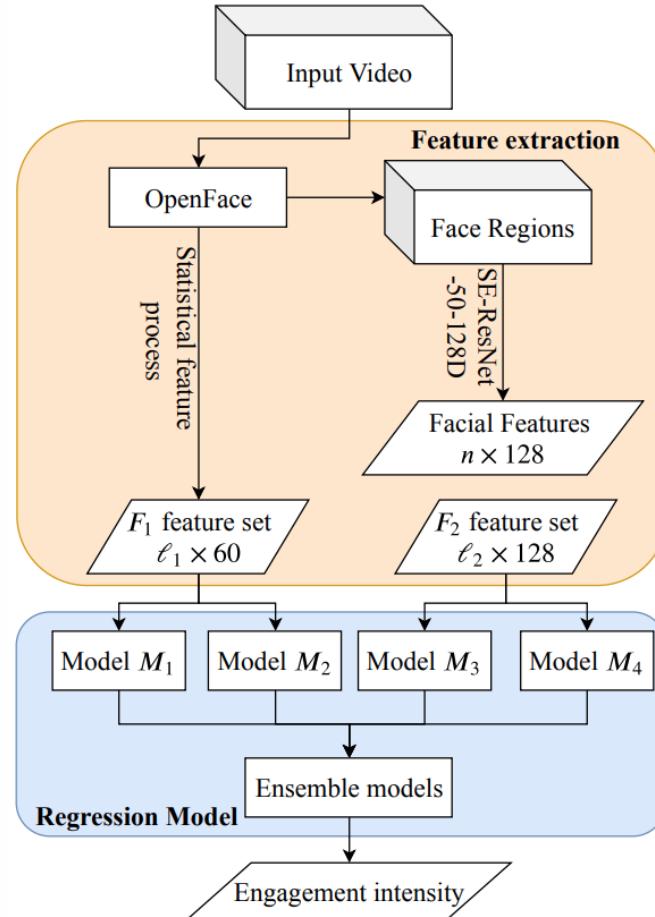
(d) Strongly agree



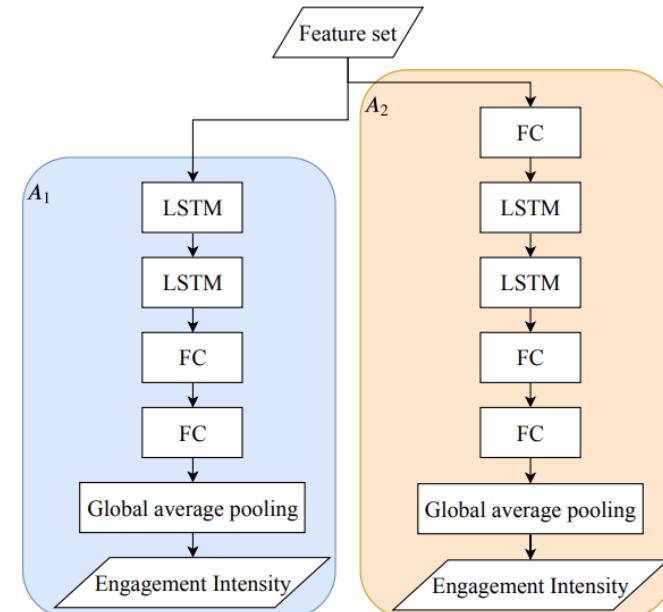
Степень вовлеченности



Engagement prediction



Feature type	Feature information	Dim
Gaze direction	mean, standard deviation	16
Eye landmarks 2D and 3D, distance from eye to camera	mean, standard deviation, coefficient of variation, ratio between the min and max values	32
Head pose	mean, standard deviation	12





Перейдем к примерам

<https://github.com/HSE-asavchenko/MADE-mobile-image-processing/tree/master/lesson6/src>



Домашнее задание

Разработать мобильное приложение для упорядочивания фотографий в галерее на основе кластеризации лиц. Минимум для сдачи работы:

1. Не менее двух различных методов извлечения дескриптора лица
2. Нейросетевой детектор лиц
3. Реализовать любой метод кластеризации, которому не нужно задавать число кластеров
4. Визуализировать результаты группировки фотографии каждого человека в фотоальбоме, распознать и отобразить пол, возраст, эмоции

Дополнительные баллы – за реализацию нескольких различных нейросетевых детекторов лиц, методов распознавания пола, возраста, эмоций, кластеризации именно лиц (не произвольных векторов признаков, например,

https://personal.ie.cuhk.edu.hk/~ccloy/files/aaai_2018_merge.pdf)

Для того, чтобы сдать задание, необходимо исходный код выложить на github (публичный или приватный с доступом для HSE-asavchenko), сделать pull request из бранчи в мастер, ссылку на PR прислать мне.

Сдать задание нужно до 06.11.2020 (19:00)