Алгоритм последовательной жадной оптимизации целевой функции кодирования для создания биометрических шаблонов в задаче распознавания лиц

Горбацевич В.С. Визильтер Ю.В. Воротников А.В.

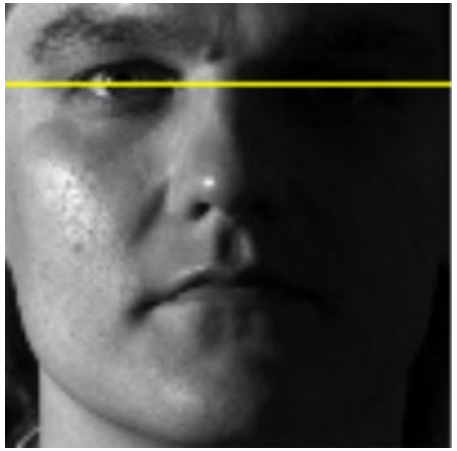
gvs@gosniias.ru

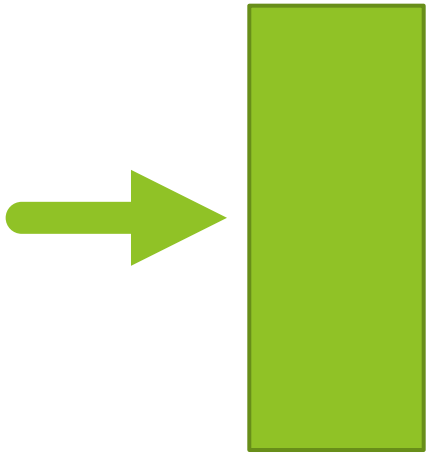
ФГУП ГосНИИАС

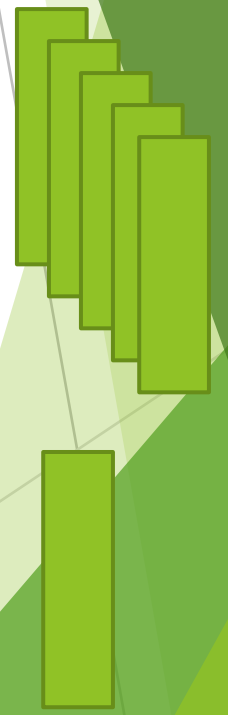
Распознавание лиц

поиск лиц

База данных







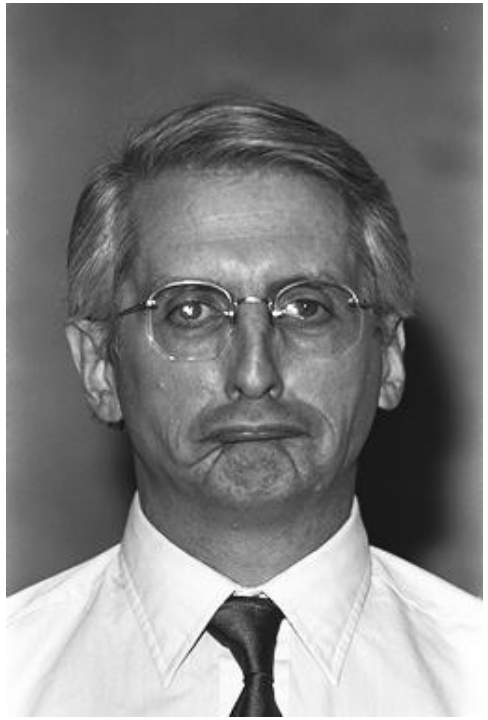
Нормализация

Построение биометрического шаблона

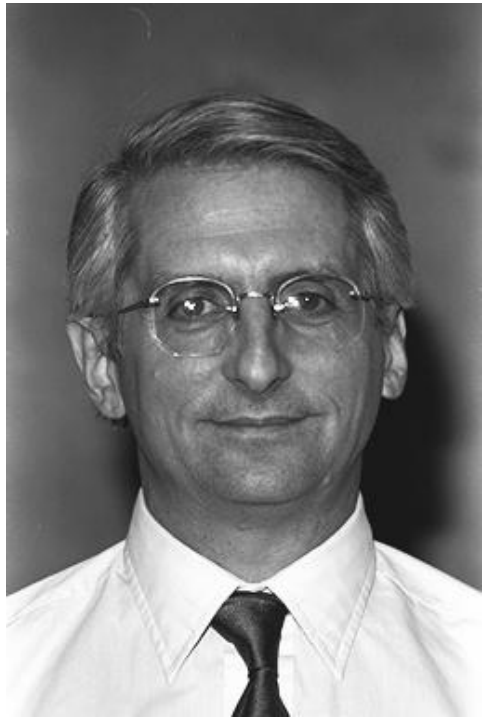
Сравнение

Распознавание лиц

Задача верификации:



?

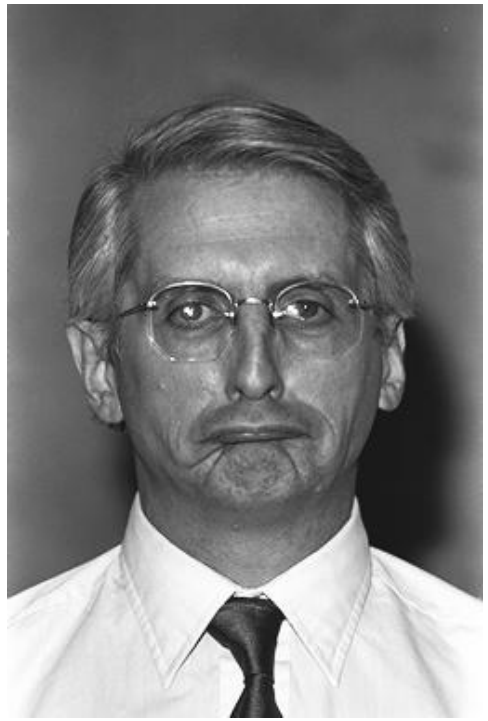


Задача верификации: попарное сравнение

характеристики FAR/FRR

Распознавание лиц

Задача идентификации:

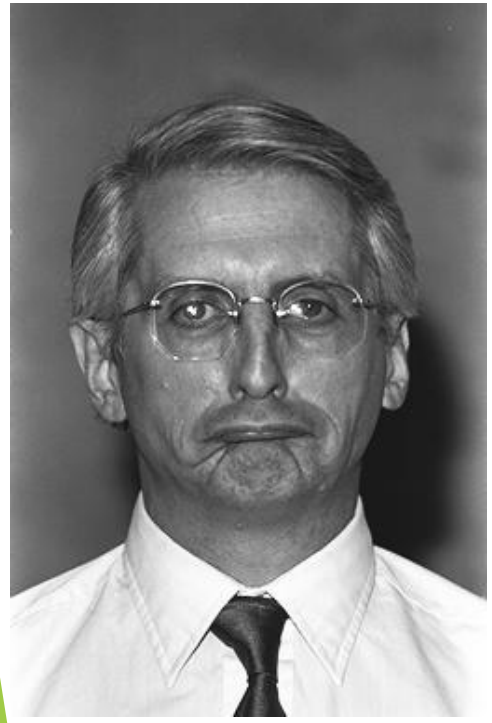


?

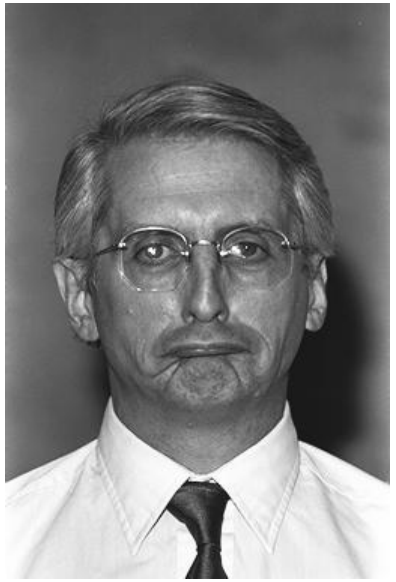
Задача идентификации : поиск в базе

характеристики CMC / RANK 1

Распознавание лиц

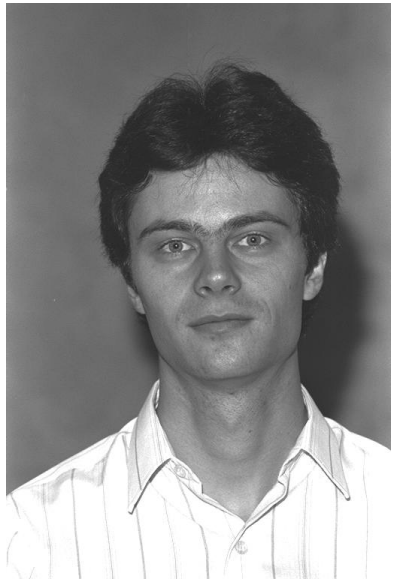


ряд кандидатов

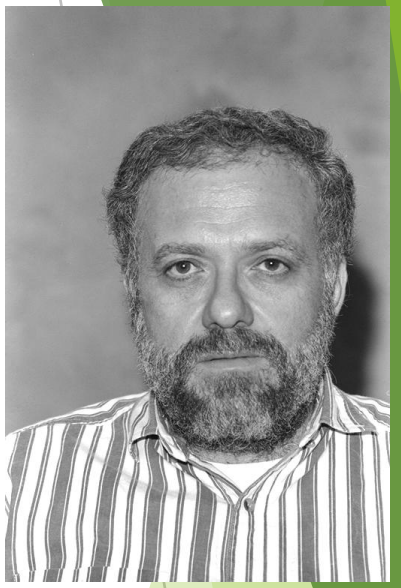


№1

№2



№3



№4

Задача идентификации : поиск в базе

характеристики CMC / RANK 1

Биометрический шаблон

Нормализация

Построение биометрического шаблона

+ компактность

+ высокое качество сравнения

+ Высокая скорость построения

Хеширование

0,001
10
2
5
0
12
35
…

Дескриптор

0
1
1
0
0
1
1
1

хэш

0,001
10
2
5
0
12
35
…

?

0,001
10
2
5
0
12
35
…

Методика сравнения

0	0
1	1
1	1
0	0
0	0
1	1
1	1
1	1

Расстояние Хемминга

+ компактность

+ крайне высокая скорость сравнения

Хеширование

0,001
10
2
5
0
12
35
…

0
1
1
0
0
1
1
1

Дескриптор

хэш

0,001
10
2
5
0
12
35
…

?

0,001
10
2
5
0
12
35
…

Методика сравнения

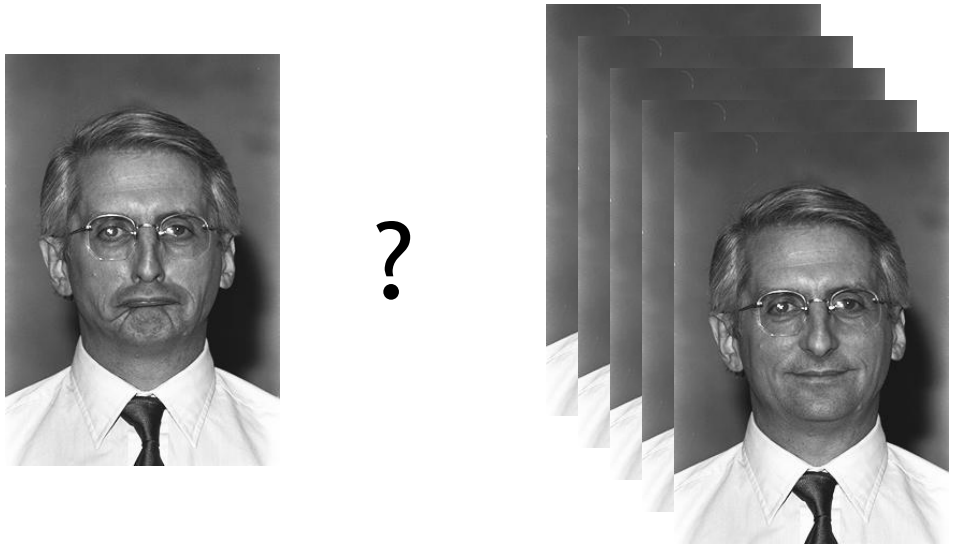
0	0
1	1
1	1
0	0
0	0
1	1
1	1
1	1

Расстояние Хемминга

Алгоритмы хеширования:

+ LSH
+ SH
+ SSH
+ Random Hyperplane Hashing
+ ITQ
+ Boost SSH

Специфика распознавания лиц:



+ очень большое число классов

+ малое число представителей одного класса

+ наличие только “парной” информации

+ огромное количество признаков различной природы

Задача кодирования:

Имеется выборка объектов $\{o_i\}_{i=1,\dots,N}$, описываемых некоторыми m-мерными векторами признаков $X = \{\mathbf{x}_i \in R^m\}_{i=1,\dots,N}$. Требуется отобразить X в n-мерное пространство с метрикой Хэмминга: $X = \{\mathbf{x}_i \in R^m\}_{i=1,\dots,N} \rightarrow Y = \{\mathbf{y}_i \in \{+1,-1\}^n\}_{i=1,\dots,N}$. Функция:

$$\mathbf{h}(\mathbf{x}): \mathbf{x} \in R^m \rightarrow \mathbf{y} \in \{+1,-1\}^n$$

называется n-битной *хэш-функцией* или n-битным *кодером*. Соответственно $\mathbf{y}=h(\mathbf{x})$ называется n-битным *хэш-кодом* $\mathbf{x}$ относительно функции h. Функция

$$h(\mathbf{x}): \mathbf{x} \in R^m \rightarrow y \in \{+1,-1\}$$

называется *элементарной хэш-функцией* или *элементарным кодером*.

Будем также использовать следующие обозначения:

$d_{ij} = \| \mathbf{x}_i - \mathbf{x}_j \|$ – *исходное расстояние* между парой векторов в пространстве признаков;

$h_{ij} = \| \mathbf{y}_i - \mathbf{y}_j \|/2$ – *расстояние Хэмминга* между соответствующей парой векторов в пространстве признаков.

Элементарный кодер

Пусть на каждой паре объектов выборки определена *функция управления парной разделимостью* (pairwise reparability control function)
$r(\mathbf{x}_i,\mathbf{x}_j)=r_{ij}\in R$, такая что:

$r_{ij}>0$, если искомая элементарная хэш-функция h должна объекты i и j отождествлять;
$r_{ij}<0$, если искомая элементарная хэш-функция h должна объекты i и j различать.

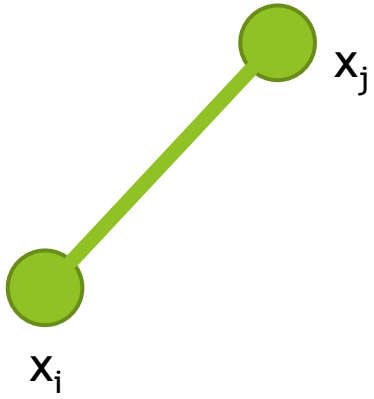
Задача обучения элементарной хэш-функции на выборке X сводится в таком случае к задаче минимизации суммарного *критерия парной разделимости выборки:*

$$J(r,h) = \sum_{i=1,\dots,N} \sum_{j=1,\dots,N} r_{ij}\, h_{ij} \to min\{h \in \mathbf{H}\},$$

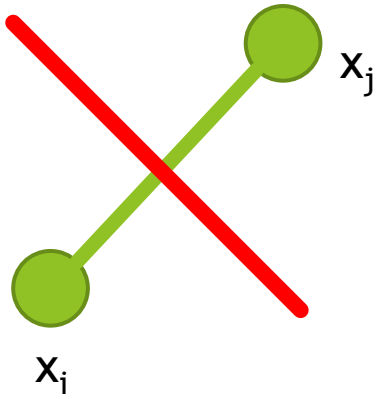
$\mathbf{H} = \{h(\mathbf{w}, t, \mathbf{x}) = sgn(\sum_{k=1,\dots,m} w_k x_k + t)\}$, класс *пороговых линейных разделителей*

Элементарный кодер:обучение

1.Поиск случайной “чужой пары”:



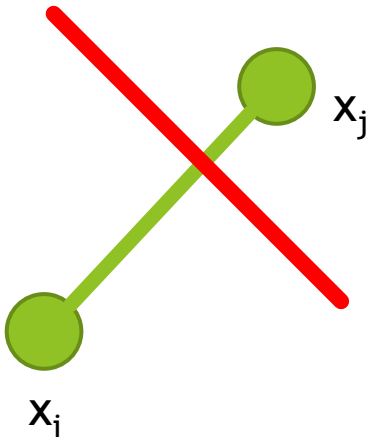
2.Выбор направления для гиперплоскости:



Направление гиперплоскости задается вектором (x_i, x_j)

3.Поиск оптимального смещения:

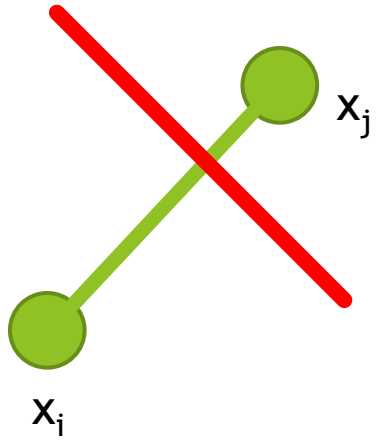
$$\mathbf{H} = \{h(\mathbf{w}, t, \mathbf{x}) = sgn(\sum_{k=1,\ldots,m} w_k x_k + t)\},$$



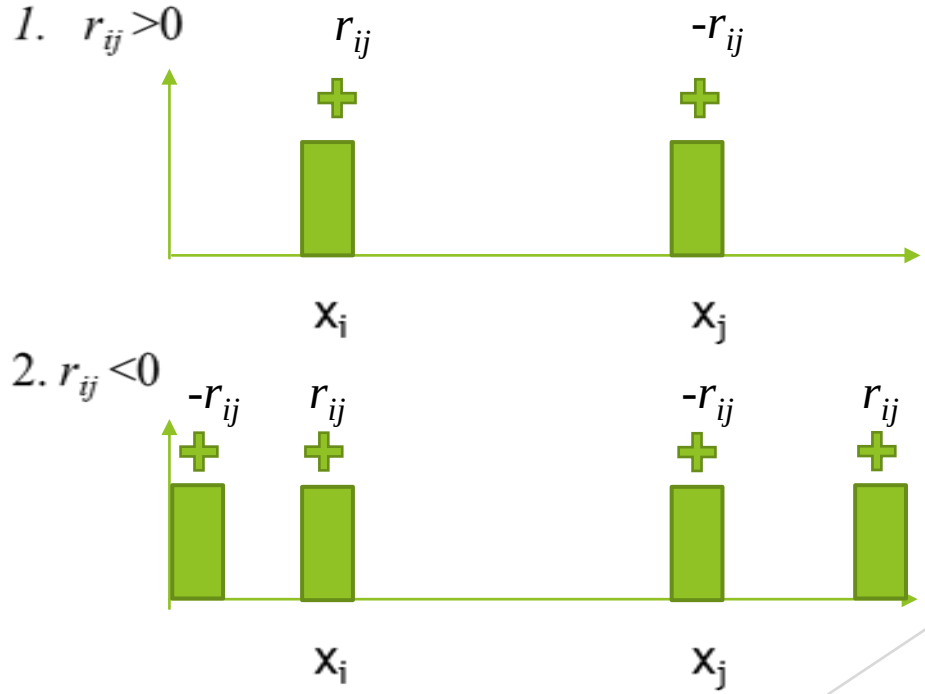
Поиск оптимального смещения проводится путём минимизации функционала $J(r,h)$ по t при фиксированных w_k

Элементарный кодер:обучение

3.Поиск оптимального смещения:



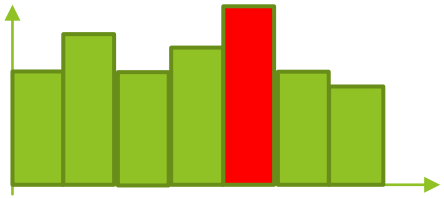
1. Вычисление проекций для каждого элемента $\mathbf{x}_i \in \mathbf{X}$ проекции на плоскость $\boldsymbol{w}$: $\mathbf{y}_i = (\boldsymbol{w}, \mathbf{x}_i)$
2. Накопление гистограммы производных $J(r,h)'_t$:



3. Восстановление $J(r,h)$:

$$J(r,h) = \int J(r,h)'_t \mathrm{dt}$$

4. Поиск оптимального смещения:



Задача кодирования:

Пусть задан некоторый *критерий кодирования, основанный на парных расстояниях* (distance-based objective function):

$$\mathcal{I}(X,\mathbf{h}) = \mathcal{I}(\mathbf{h}(\mathbf{x})\in\mathbf{H},\ \{d_{ij} = \|\ \mathbf{x}_i - \mathbf{x}_j\ \|_2,\ \mathbf{x}_i,\mathbf{x}_j\in X\}_{i,j=1,\ldots,N},\ \{h_{ij} = \|\ \mathbf{h}(\mathbf{x}_i) - \mathbf{h}(\mathbf{x}_j)\ \|_1,\ \mathbf{x}_i,\mathbf{x}_j\in X\}_{i,j=1,\ldots,N}) =$$
$$= \mathcal{J}_{\mathbf{H},n}(D_X = \{d_{ij}\}_{i,j=1,\ldots,N},\ D_H = \{h_{ij}\}_{i,j=1,\ldots,N}) \to min(D_H = \{h_{ij}\}_{i,j=1,\ldots,N}).$$

Смысл данного выражения в том, что искомая n-битная хэш-функция $\mathbf{h}(\mathbf{x})$, состоящая из однобитных хэш-функций заданного класса $\mathbf{H}$, формируется при обучении на выборке X таким образом, что в процессе обучения:

- в качестве входной информации используются не конкретные значения векторов признаков $\mathbf{x}$, а парные расстояния между ними d_{ij};
- в качестве целевых переменных контролируются не конкретные значения получаемых кодов $\mathbf{h}(\mathbf{x})$, а парные расстояния Хэмминга между кодами h_{ij}.

Задача кодирования:

(Базовый алгоритм **SDC**)

Входные данные: D_X, $\mathcal{J}_{\mathbf{H},n}(D_X,D_H)$.

Выходные данные: $\mathbf{h}^{(n)}(\mathbf{x})$: $\mathbf{x}\in R^m \rightarrow \mathbf{y}\in\{+1,-1\}^n$, $\mathbf{h}(\mathbf{x})\in\mathbf{H}$.

Инициализация:

Шаг 0. k:=0; $\mathbf{h}^{(k)}$:= ().

Итеративное повторение шагов:

k:= k+1;

Шаг 1. Формирование матрицы весов функции управления парной разделимостью:

Для всех i=1,…,N, j=1,…,N:

$$r^{(k-1)}_{ij} := \partial\mathcal{J}_{\mathbf{H},n}(D_X,D_H)/\partial h_{ij}\big|_{h_{ij} = \| \mathbf{h}^{(k-1)}(\mathbf{x}_i) - \mathbf{h}^{(k-1)}(\mathbf{x}_j)\|_1};$$

// численное значение частной производной $\mathcal{J}_{\mathbf{H},n}$ по h_{ij} на текущей k-й итерации

Шаг 2. Обучение k-го однобитового кодера.

Шаг 3. Добавление k-го однобитового кодера:

$$\mathbf{h}^{(k)}(\mathbf{x}) := (\mathbf{h}^{(k-1)}(\mathbf{x}), h^{(k)}(\mathbf{x}));$$

пока $k<n$.

Обучение на основе критерия межклассовых и внутриклассовых расстояний (верификация)

«Идеальное» (желаемое) расстояние, которого хотелось бы добиться при k-битном кодировании:

$$g^{(k)}_{ij} = \{0, \text{если } \mathrm{class}(\mathbf{x}_i) = \mathrm{class}(\mathbf{x}_j);$$
$$k \text{ – в противном случае}\}.$$

С учетом этого *целевой дистанционный критерий для задач обучения с учителем* может быть сформирован следующим образом:

$$v_{ij} = a\, s_{ij} + b\,(1\text{-}s_{ij}),\ a \geq b \geq 0.$$

$$\mathcal{J}_{\mathbf{H},n}(D_G, D_H) = \sum_{i=1,\ldots,N} \sum_{j=1,\ldots,N} v_{ij}\,(h_{ij} - g_{ij})^2 \to min(D_H = \{h_{ij}\}_{i,j=1,\ldots,N}).$$

Введём нормированные (*удельные*) расстояния Хэмминга для k-битных ключей:

$$\hat{h}^{(k)}_{i,j} = h^k_{i,j}/k$$

$$\mathcal{J}_{\mathbf{H},n}(D_G, D_H) = \sum_{i=1,\ldots,N} \sum_{j=1,\ldots,N} v_{ij}\,(\hat{h}_{i,j} - \hat{g}_{i,j})^2 \to min(D_H = \{h_{ij}\}_{i,j=1,\ldots,N}),$$

$$\hat{g}_{i,j} = \{0, \text{если } \mathrm{class}(\mathbf{x}_i) = \mathrm{class}(\mathbf{x}_j);$$
$$1 \text{ – в противном случае}\}.$$

Обучение на основе критерия межклассовых и внутриклассовых расстояний (верификация)

Ускорение процесса обучения:

- обнулять $r^{(k)}_{ij}$ для всех «своих» пар, в которых расстояние Хэмминга $h^{(k-1)}_{ij}$ на предыдущем шаге больше среднего по всем «своим» парам;

- обнулять $r^{(k)}_{ij}$ для всех «чужих» пар, в которых расстояние Хэмминга $h^{(k-1)}_{ij}$ на предыдущем шаге меньше среднего по всем «чужим» парам.

- обнулять $r^{(k)}_{ij}$ для всех «чужих» пар, в которых расстояние Хэмминга $h^{(k-1)}_{ij}$ на предыдущем шаге больше чем 3 СКО + МО для своих пар.

Так мы на каждом шаге итерации усиливаем давление на те пары, которые отличаются от остальных в нежелательную сторону (т.е. подгоняем отстающих, не трогая лидеров).

Обучение на основе критерия правильной идентификации

Назовем *рядом кандидатов* (или *последовательностью выдачи*) для запроса $\mathbf{x}_i \in X$ на основе хеш-функции $\mathbf{h}$ последовательность

$\mathbf{p}(\mathbf{x}_i,\mathbf{h}) = (\mathbf{p}_1(\mathbf{x}_i,\mathbf{h}),\ldots,\mathbf{p}_{N-1}(\mathbf{x}_i,\mathbf{h})): \forall j \in \{1,\ldots,N\text{-}1\}\ \mathbf{p}_j(\mathbf{x}_i,\mathbf{h}) = \mathbf{x}_j \in X,\ \mathbf{x}_j \neq \mathbf{x}_i,$

$\forall j \in \{1,\ldots,N\text{-}2\}\ h_{i,j} \le h_{i,j+1},\ h_{i,j} = \| \mathbf{h}(\mathbf{x}_i) - \mathbf{h}(\mathbf{x}_j) \|_1.$

Обозначим через $I_{\mathbf{h}}(\mathbf{x}_i,\mathbf{x}_j)$ номер (*индекс* или *ранг*) объекта $\mathbf{x}_j$ в последовательности выдачи $\mathbf{p}(\mathbf{x}_i,\mathbf{h})$. Очевидно, что в случае идеального хеш-кода

$\forall\ \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in X$, $\text{class}(\mathbf{x}_i) = \text{class}(\mathbf{x}_j) \neq \text{class}(\mathbf{x}_k)$: $h_{i,j} < h_{i,k} \Leftrightarrow I_{\mathbf{h}}(\mathbf{x}_i,\mathbf{x}_j) < I_{\mathbf{h}}(\mathbf{x}_i,\mathbf{x}_k)$.

Отклонение от такого идеального порядка мы будем штрафовать.

Для этого введем функции

$\sigma_i(\mathbf{x}_j,\mathbf{x}_k) = \{$-1, если $h_{i,j} < h_{i,k}$; // первая точка ближе второй

1, если $h_{i,j} \ge h_{i,k}$ // вторая точка ближе первой$\}$,

и

$\tau_i(\mathbf{x}_j,\mathbf{x}_k) = \{$ 0, если $\text{class}(\mathbf{x}_i) = \text{class}(\mathbf{x}_j) = \text{class}(\mathbf{x}_k)$; // обе точки «свои»

0, если $\text{class}(\mathbf{x}_i) \neq \text{class}(\mathbf{x}_j)$, $\text{class}(\mathbf{x}_i) \neq \text{class}(\mathbf{x}_k)$; // обе точки «чужие»

1, если $\text{class}(\mathbf{x}_i) = \text{class}(\mathbf{x}_j) \neq \text{class}(\mathbf{x}_k)$; // первая «своя», вторая «чужая»

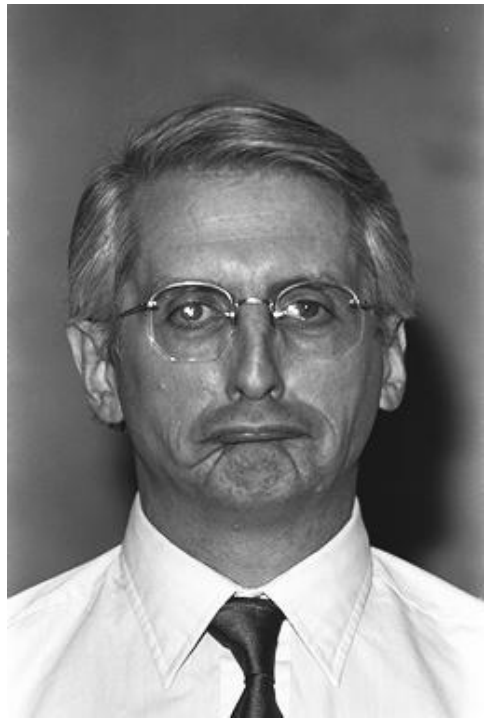
-1, если $\text{class}(\mathbf{x}_i) = \text{class}(\mathbf{x}_k) \neq \text{class}(\mathbf{x}_j)$; // вторая «своя», первая «чужая»$\}$,

целевой критерий на основе порядка выдачи кандидатов для задач обучения с учителем:

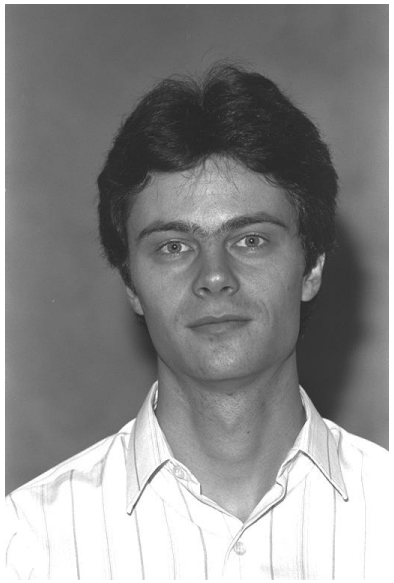
$\mathcal{J}_{Sort}(D_G, D_H) = \sum_{i=1,\ldots,N} \sum_{j=1,\ldots,N} \sum_{k=1,\ldots,N} \sigma_i(\mathbf{x}_j,\mathbf{x}_k)\ \tau_i(\mathbf{x}_j,\mathbf{x}_k) \to min(D_H = \{h_{ij}\}_{i,j=1,\ldots,N})$.

Обучение на основе критерия правильной идентификации

эталон



ряд кандидатов



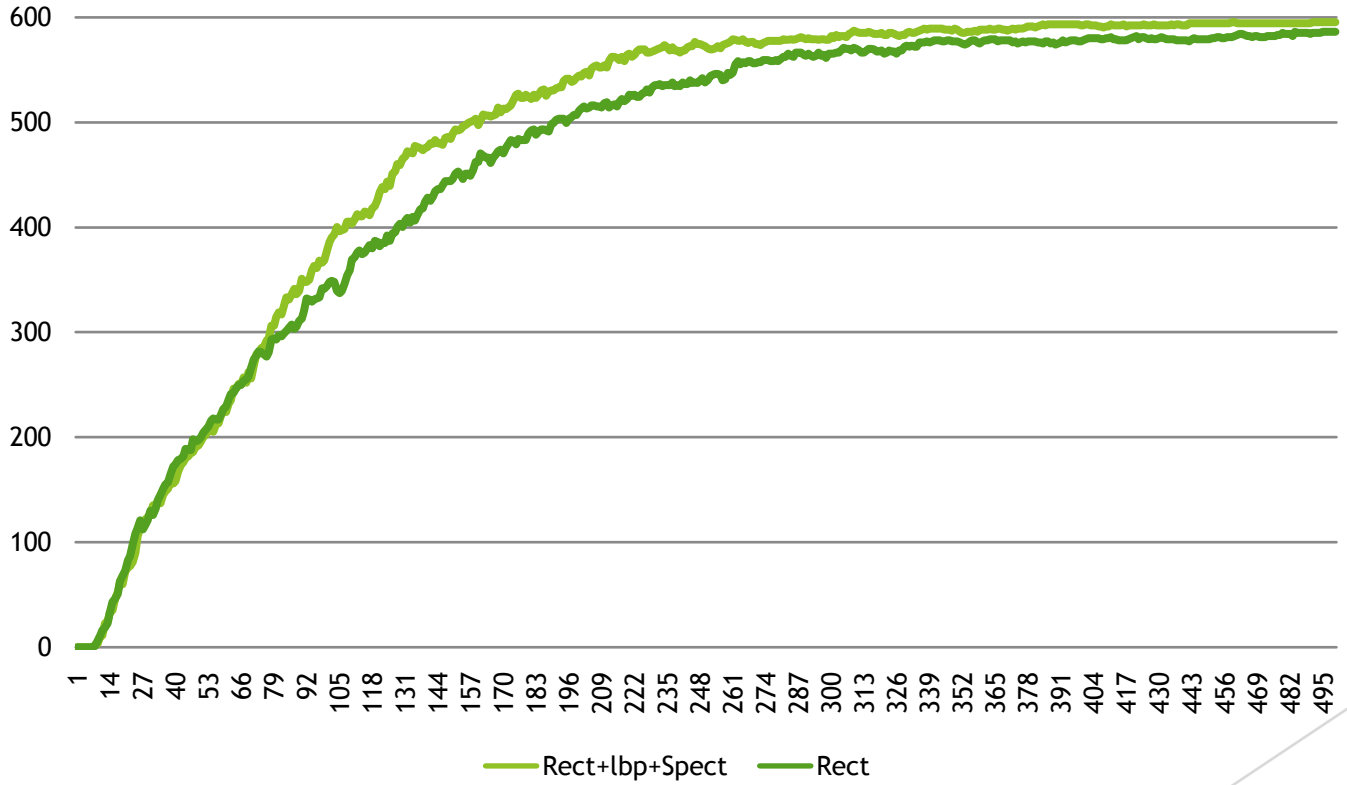
рассматривать

не рассматривать/
ослаблять

Результаты экспериментов:отбор признаков

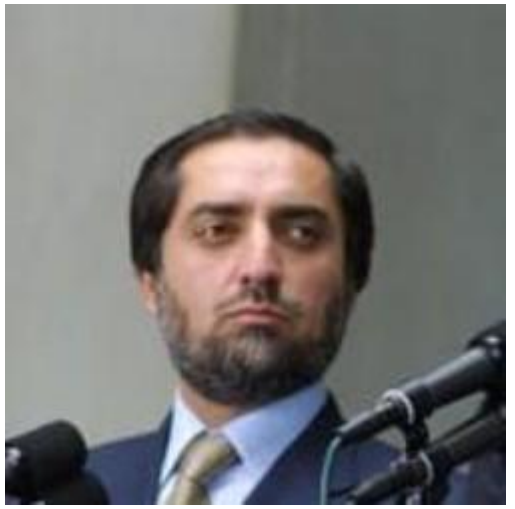
База FERET 600 изображений

Rank1

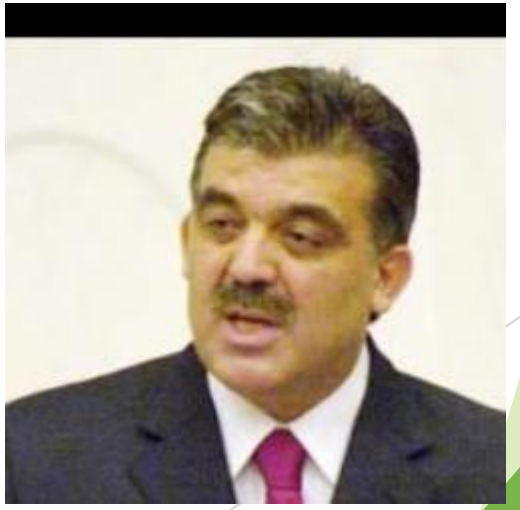


База lfw





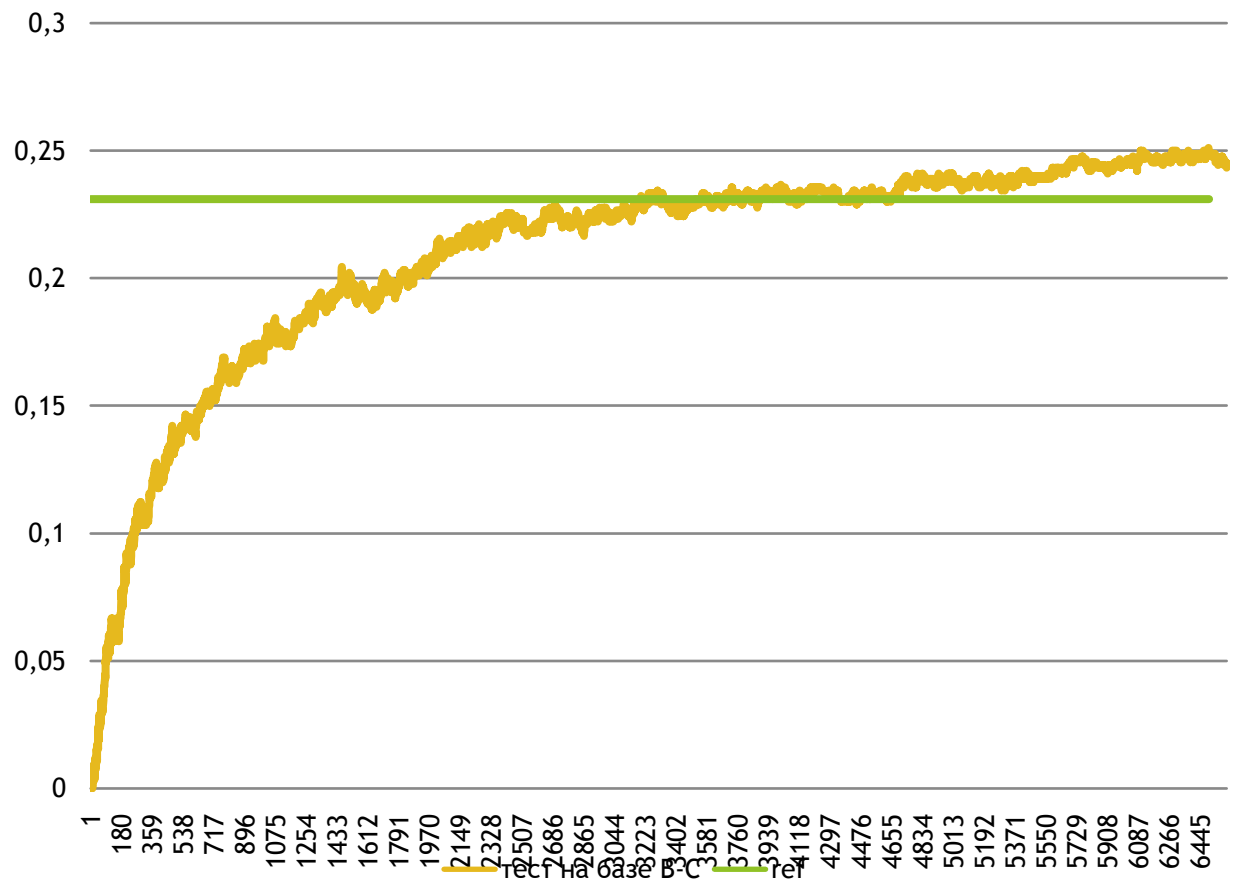





Результаты экспериментов:по базе lfw

Дескриптор lbp[1] размерность 180000 бит Image-Restricted, No Outside Data

Rank1

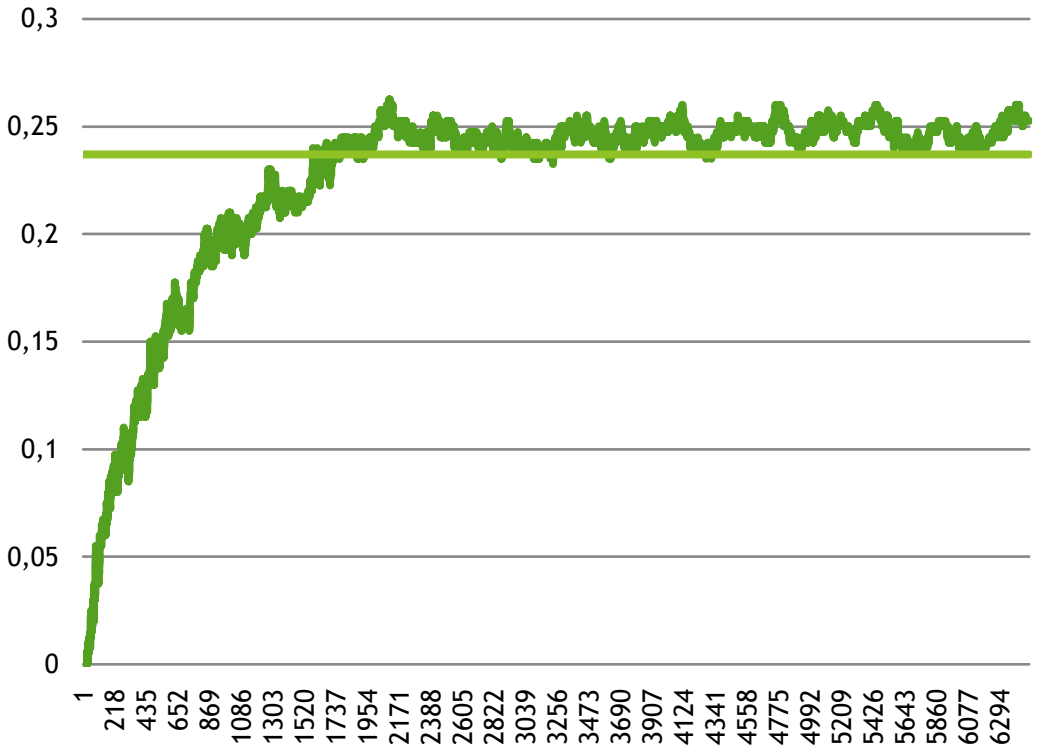


[1] Bayesian Face Revisited: A Joint Formulation ECCV'12

Результаты экспериментов:по базе lfw

Дескриптор le[1] размерность 663552 бит Image-Restricted, No Outside Data

Rank1



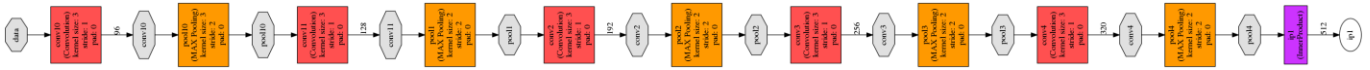
Face Recognition with Learning-based Descriptor *ERI Procedia, Volume 2, 2012, Pages 114-119*

Результаты экспериментов:по базе lfw

Дескриптор DeepID для сети:

Конв. слоев 5

Пулинг 3



Размерность вектора признаков 16384 бит

Rank1

Net: 0.81

Hash: 0.84

EER

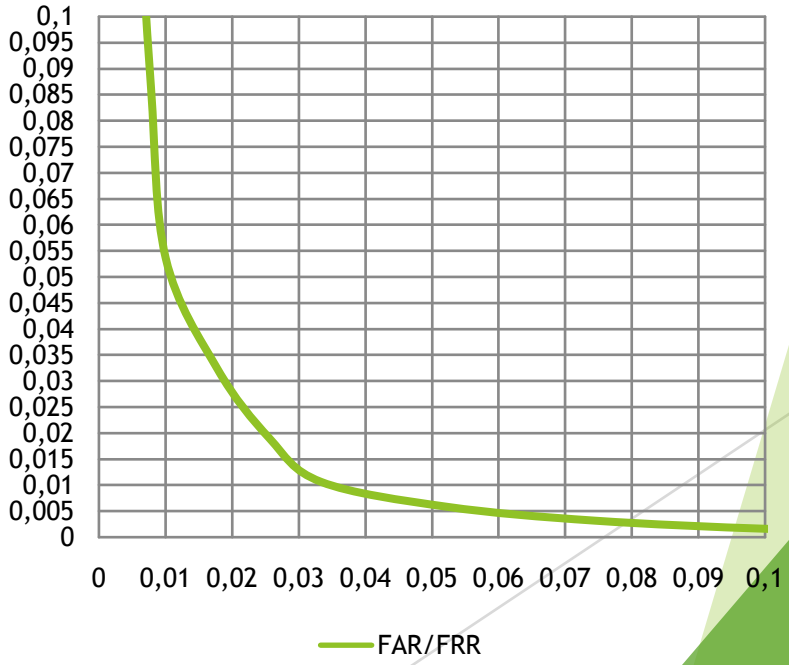
Net: 0.97

Hash: 0.981

DeepID: 0.975

Baidu: 0.997

Rank1



Выводы:

+ Позволяет строить крайне компактные бинарные описания.

+ Может работать практически с любыми пространствами признаков.

+ Позволяет строить бинарные описания с использованием различных признаков различной природы.

Спасибо за внимание