

Распознавание лиц численными методами линейной алгебры

Гиганты мысли

Илья Белько, Данила Андреев, Владислав Муджиков

Постановка задачи

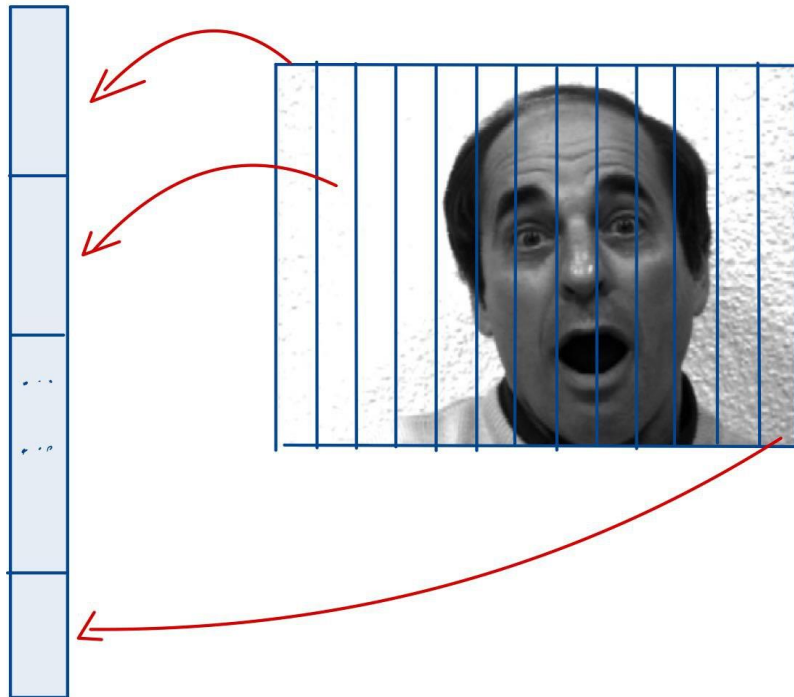
Цель проекта - исследование эффективности применения изученных численных методов линейной алгебры в задаче распознавания лиц и оптимизации хранения фотографий.

Хотим получить как можно менее объемную по занимаемому месту базу данных, наиболее быстрое и точное исполнение алгоритма распознавания лиц.

Описание методов

Наивный метод:

1. Из столбцов каждой фотографии собрать новый вектор-столбец и в таком виде их хранить в базе данных.
2. При поступлении фотографии на распознавание составить из нее вектор столбец и посчитать норму разницы между каждой фотографией в базе данных.



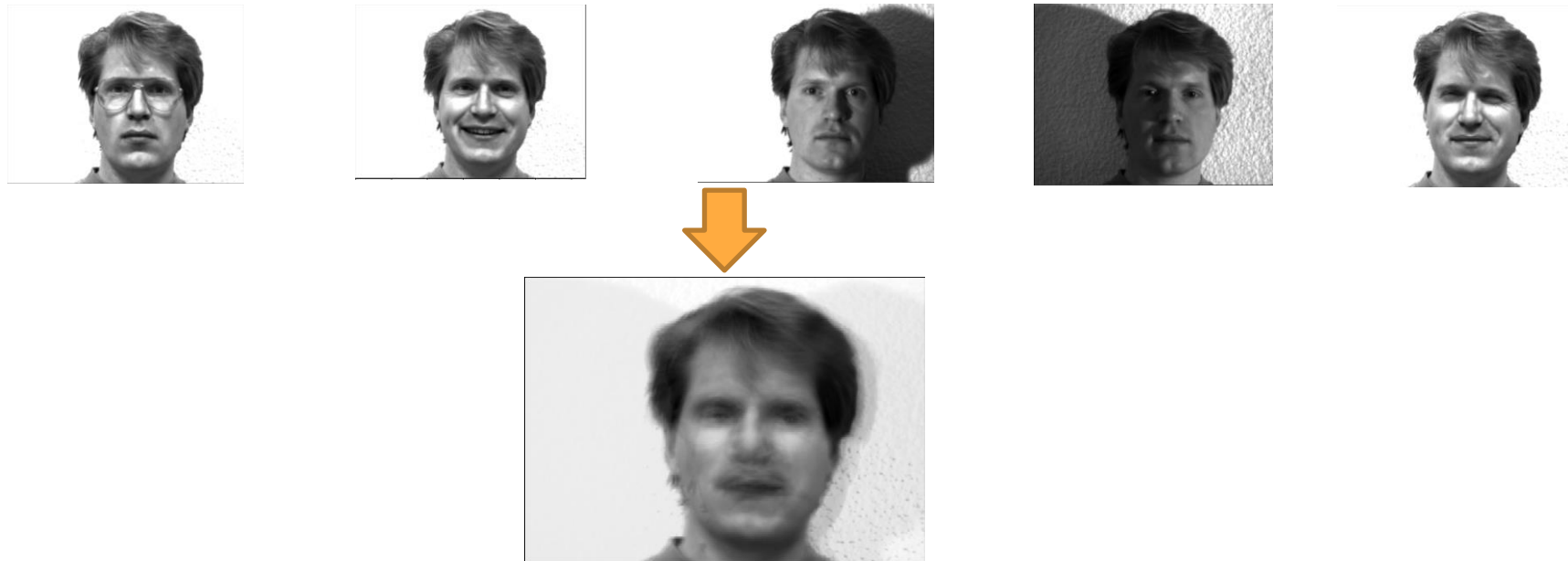
3. Взять k наименьших норм из всех рассчитанных и из них взять самый частый класс. При k=1:

$$i : ||v_{input} - v_{database[i]}||_2 = \min$$

На удивление метод дает не такое уж и плохое качество – 80% правильных распознаваний на нецентрированных снимках и 85% на отцентрированных.

Проблема наивного метода, помимо недостаточной точности в том, что нужно хранить каждую фотографию и вычислять норму разницы для каждой фотографии, при увеличении базы данных могут возникнуть проблемы.

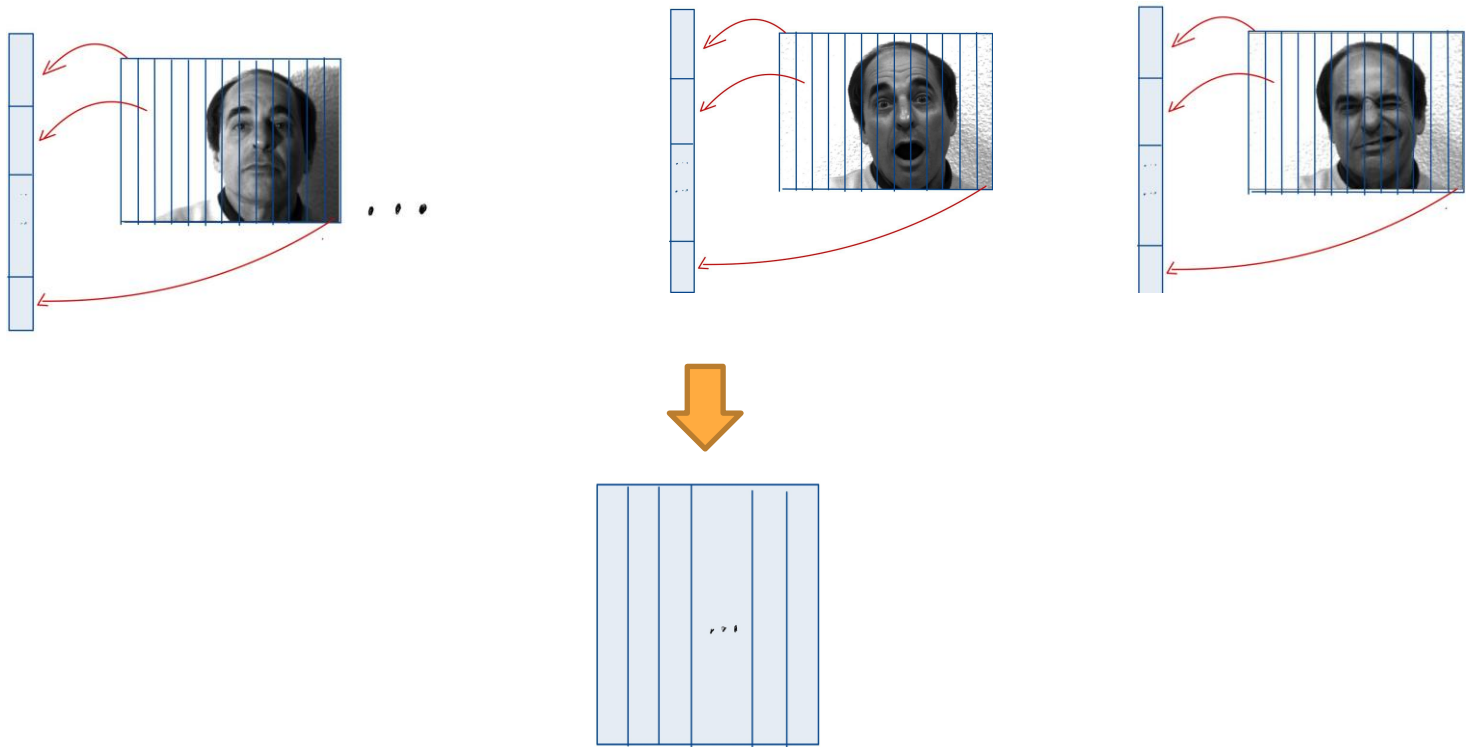
Самое простое улучшение для наивного метода – вместо хранения всех фотографий можно просто составить по одной усредненной фотографии для каждого человека в базе данных.



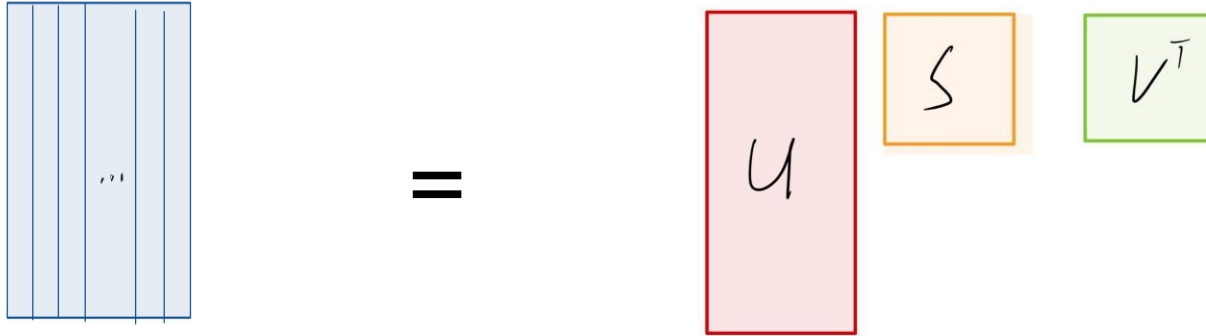
Хоть и добились сильной экономии по памяти и времени, полученная точность 66% нас не устраивает

Распознавание лиц с помощью SVD

1. Для каждого человека в базе данных составляем матрицу из векторов-столбцов всех его фотографий



2. Вычисляем SVD разложение этой матрицы



The diagram illustrates the SVD decomposition of a matrix. On the left is a blue rectangle representing the input matrix, divided into five vertical columns with an ellipsis in the center. This is followed by an equals sign. To the right of the equals sign are three matrices: a tall red rectangle labeled U , a small orange square labeled Σ , and a green square labeled V^T .

Любопытно посмотреть на малоранговую аппроксимацию в контексте фотографий.
Для этого рассмотрим столбцы матрицы U .

$\delta = 176267$



$\delta = 25107$



$\delta = 10935$



$\delta = 7999$



$\delta = 5385$



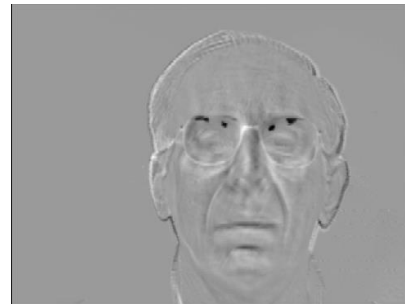
$\delta = 4326$



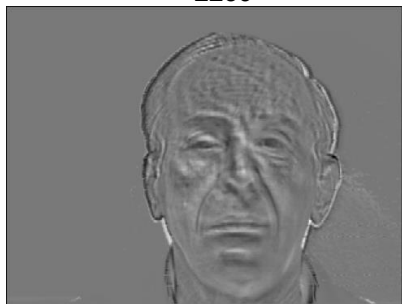
$\delta = 3292$



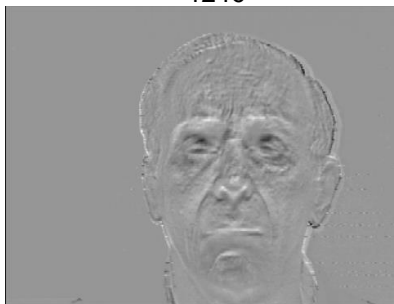
$\delta = 3008$



$\delta = 2239$



$\delta = 1246$



В базе данных храним полученные матрицы U для каждого человека, при поступлении фото для классификации, раскладываем его на вектор и вычисляем наилучшее решение методом наименьших квадратов

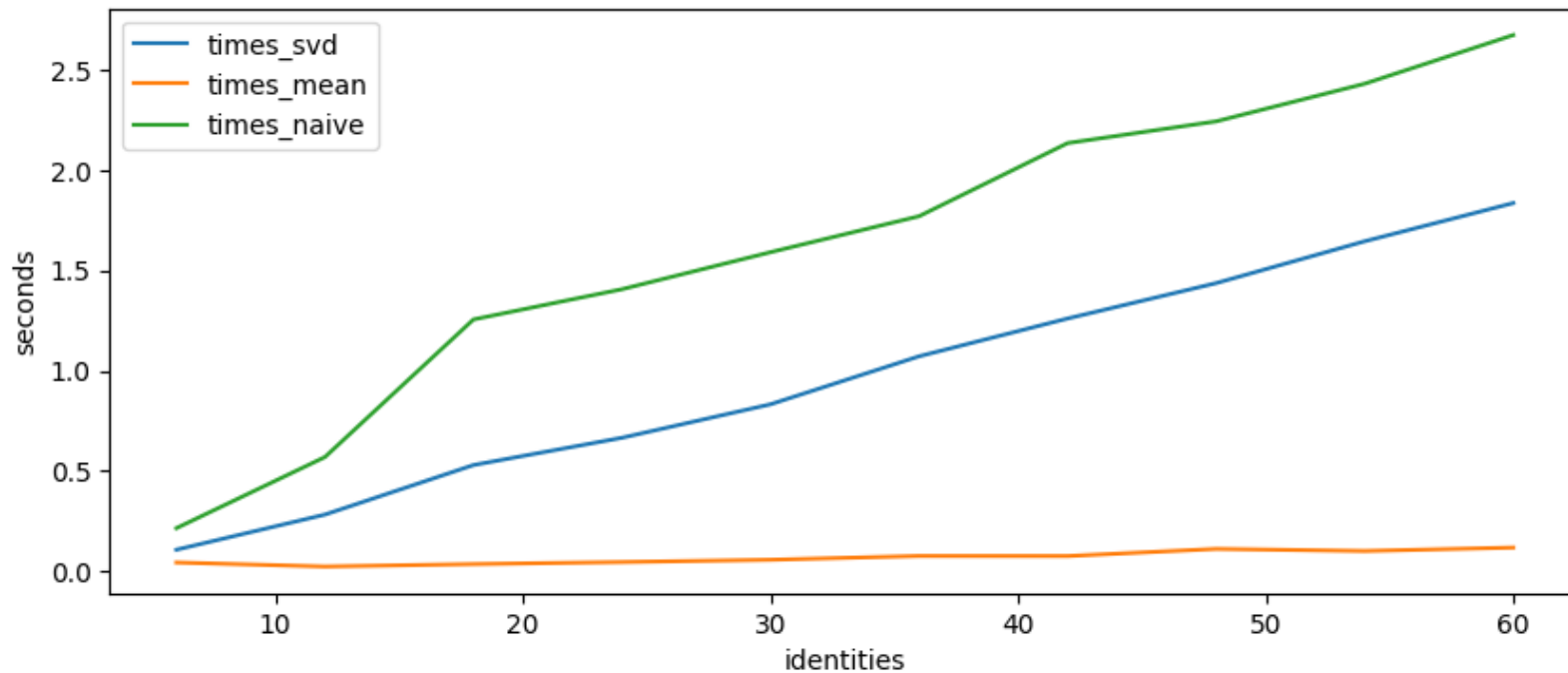
$$i : ||v_{input} - U_i U_i^T v_{input}||_2 = \min$$

Выбираем наименьшую по значению полученную норму из всех и готово, мы классифицировали наше фото.

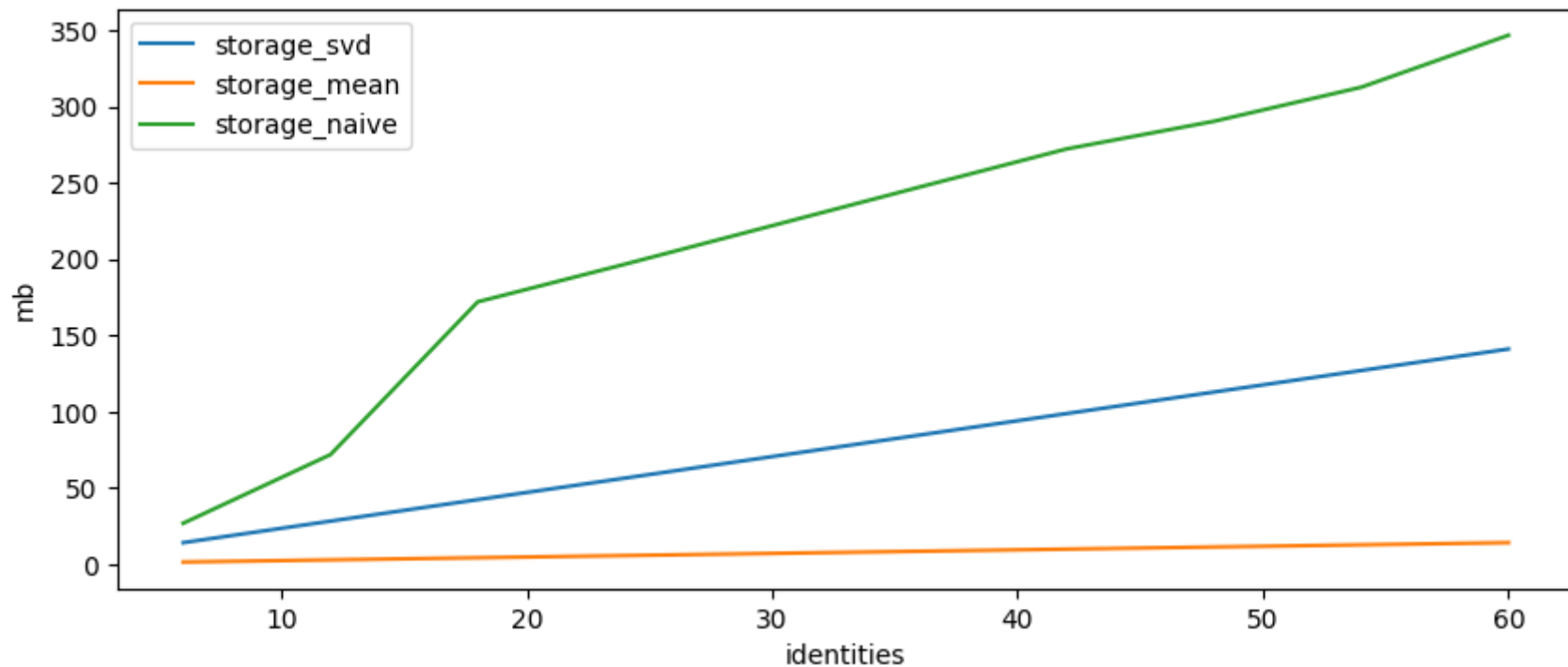
При хранении матрицы U ранга 3 получаем наилучшую точность – 94% верно классифицированных фото, далее при увеличении ранга точность не растет. Хранение U ранга 3 эквивалентно хранению трех фотографий.

Сравнение по занимаемой памяти и скорости методов

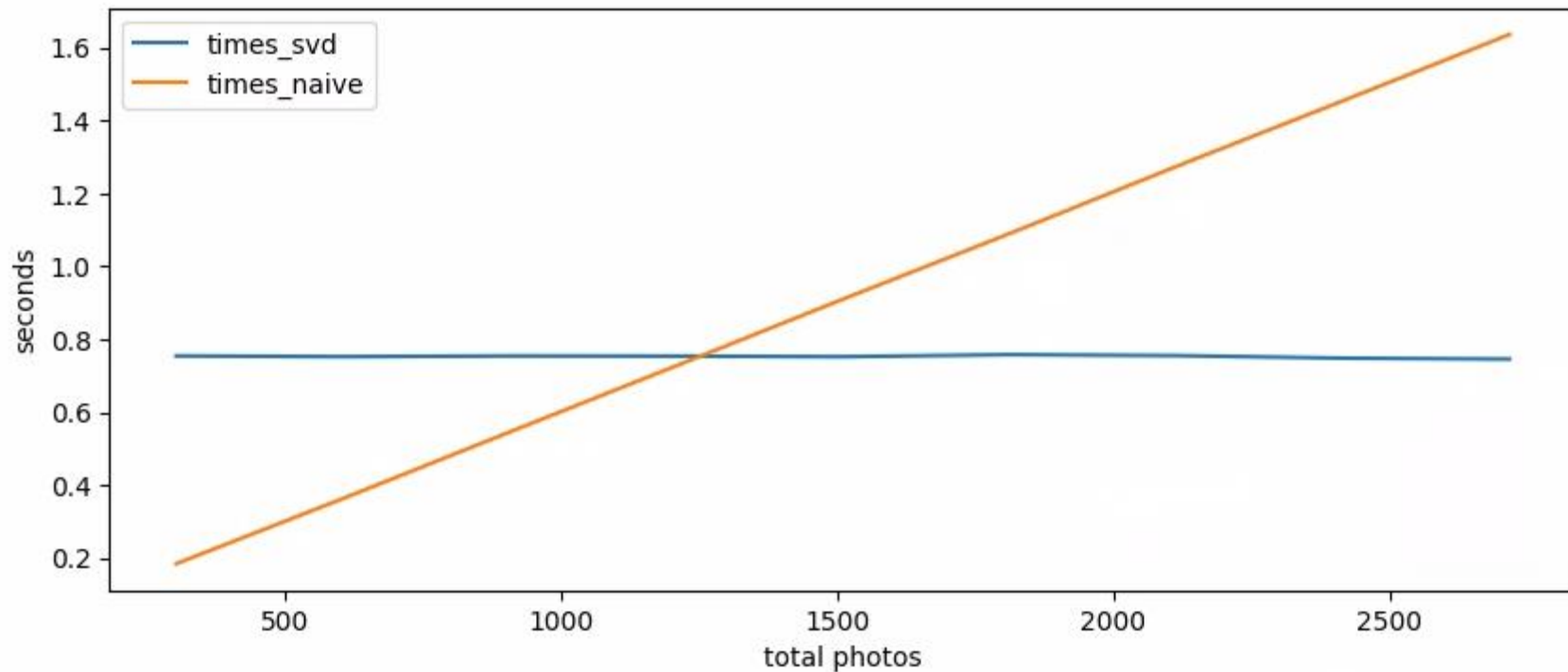
Время классификации по количеству человек в базе данных



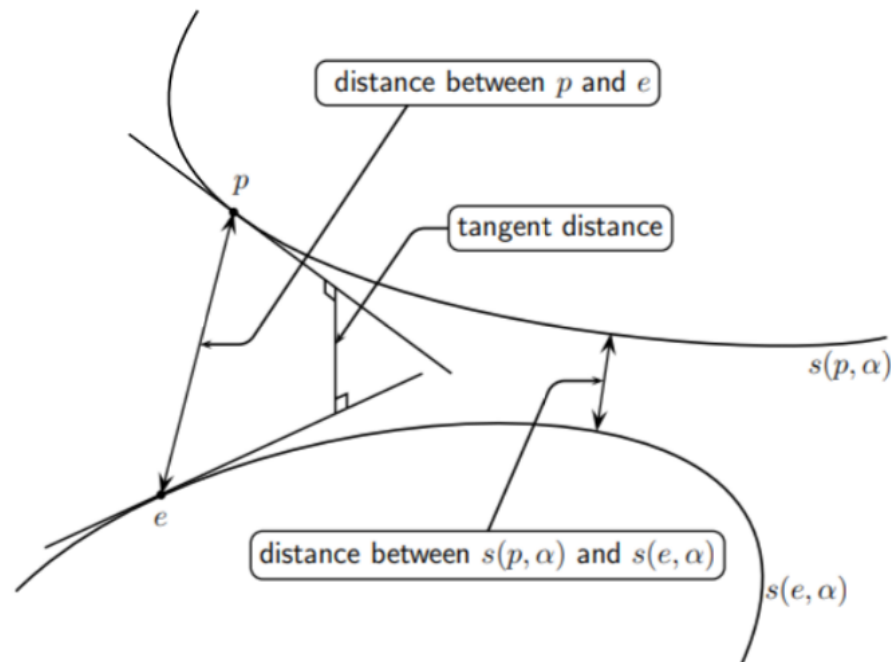
Размер базы данных по количеству человек в ней



Время классификации по количеству фото в базе данных



Распознавание лиц с помощью Tangent Distance



Изображение размером $n \times m$ представим как точку в пространстве $R^{n \times m}$. Сначала рассмотрим случай одного преобразования в направлении x . Разложим кривую трансформации для p по первым двум членам ряда Тейлора:

$$s(p, \alpha) = s(p, 0) + \frac{ds}{d\alpha}(p, 0) \alpha + O(\alpha^2) \approx p + t_p \alpha,$$

Аналогично, для e :

$$s(e, \alpha) \approx e + t_e \alpha.$$

Теперь если мы посчитаем обычное евклидово расстояние, мы решим задачу наименьших квадратов:

$$\min_{\alpha_p, \alpha_e} \|p + t_p \alpha_p - e - t_e \alpha_e\|_2 = \min_{\alpha_p, \alpha_e} \left\| (p - e) - (-t_p \quad t_e) \begin{pmatrix} \alpha_p \\ \alpha_e \end{pmatrix} \right\|_2$$

Figure 10.12. The distance between the points p and e , and the tangent distance.

В более общем случае, мы можем двигаться вдоль l - различных кривых, параметризованных $\alpha = (\alpha_1 \cdots \alpha_l)^T$.

Это эквивалентно тому, что мы будем перемещать картинку вдоль l -мерного многообразия. В этом

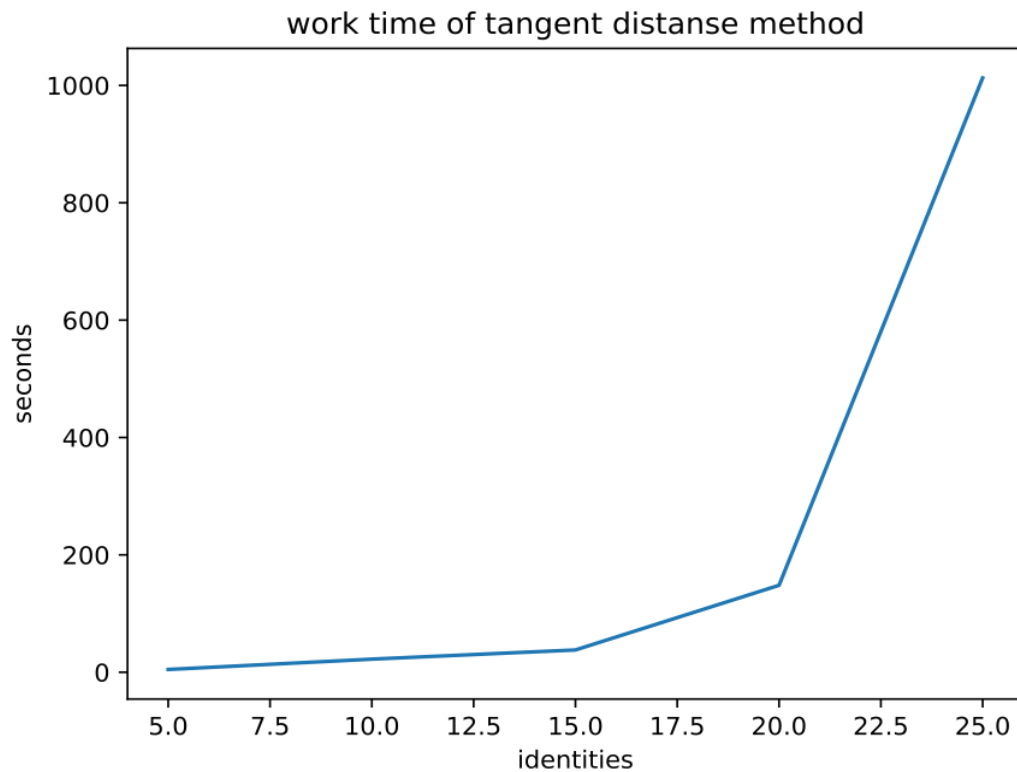
случае разложение для кривой будет: $s(p, \alpha) = s(p, 0) + \sum_i^l \frac{ds}{d\alpha_i}(p, 0) \alpha_i + O(\|\alpha\|_2^2) \approx p + T_p \alpha$, где $T_p = \left(\frac{ds}{d\alpha_1} \quad \frac{ds}{d\alpha_2} \quad \cdots \quad \frac{ds}{d\alpha_l} \right)$, все производные взяты в точке $(p, 0)$.

И наконец считаем расстояние:

$$\min_{\alpha_p, \alpha_e} \|p + T_p \alpha_p - e - T_e \alpha_e\|_2 = \min_{\alpha_p, \alpha_e} \left\| (p - e) - \begin{pmatrix} -T_p & T_e \end{pmatrix} \begin{pmatrix} \alpha_p \\ \alpha_e \end{pmatrix} \right\|_2.$$

В заключении, отметим, что хотя алгоритм довольно хорош с точки зрения классификации, он дорогостоящий: у каждой тестовой фотографии, необходимо будет посчитать матрицу частных производных, и дальше сравнивать со всеми обучающими фотографиями.

Время классификации по количеству фото в базе данных



Выводы

- Реализовали алгоритмы исполнения методов.
- Протестировали по качеству, времени выполнения и занимаемой памяти.
- Поняли, что метод с SVD неплохо справляется с этой задачей, но в основном задача классификации лиц требует максимальной точности, к сожалению, без применения нейронных сетей вряд ли получится добиться этого. Также остается проблема с масштабируемостью базы данных.

Ссылка на Гитхаб:

https://github.com/CyberKachok/Giants-of-thought_Face_recognition