Санкт-Петербургский Государственный Университет

Программная инженерия

Чернявский Олег Николаевич

Анализ решений задачи кластеризации изображений лиц в сфере киберкриминалистики

Курсовая работа

Научный руководитель:
доцент кафедры СП СПбГУ, к. т. н. Литвинов Ю. В.

Научный консультант:
рук. отд. раз. ПО, ООО “Белкасофт” Тимофеев Н. М.

Санкт-Петербург
2019

Оглавление

Введение

Видеонаблюдение, как метод решения задач криминалистики, становится все более и более востребованным в организациях, связанных с правоприменением. Например, уже сейчас правоохранительные органы по всему миру проводят расследования преступлений, находят подозреваемых и определяют их причастность или непричастность к совершению правонарушения с помощью записей с камер видеонаблюдения. Эти же видеозаписи могут быть потом рассмотрены в суде в качестве доказательств. С другой стороны, все чаще и чаще в ходе расследования используются данные (в том числе и фотографии) с мобильного телефона жертвы или подозреваемого.

С ростом популярности данных подходов число подобных записей и фотографий неуклонно растет, что делает практически невозможным просмотр вручную всех кадров, потенциально представляющих интерес для криминалиста. Возникает необходимость в автоматизации данного процесса.

Решению подобных задач посвящен такой раздел теории распознавания образов (pattern recognition), как распознавание лиц (face recognition). В рамках этой области рассматриваются следующие проблемы:

- верификация лиц (face verification) – определить, принадлежат ли лица на разных фотографиях одному и тому же человеку;
- идентификация лиц (face identification) – выяснить, есть ли данный человек в базе, и если да, то кем конкретно он является;
- кластеризация лиц (face clustering) – разбить лица людей, обнаруженные на одной или нескольких фотографиях, в кластеры, каждый из которых соответствует конкретному человеку.

В данной работе рассматриваются решения только третьей задачи из этого списка. В то же время первые две проблемы также являются крайне распространенными в технологическом мире, и для них уже

было предложено большое количество решений (см., например, [18] и [19]).

Задача кластеризации изображений лиц встречается в сфере киберкриминалистики достаточно часто. Например, это может быть ситуация, когда необходимо обработать запись с камеры видеонаблюдения и отследить, где на каждом кадре находится конкретный человек. Другая возможная задача: имеется телефон подозреваемого, содержащий несколько тысяч фотографий. Необходимо сгруппировать лица людей на данных изображениях, чтобы упростить работу криминалиста по поиску конкретного человека. Визуализацию решаемой задачи можно увидеть на рис. 1.

Рис. 1: Пример задачи кластеризации изображений лиц [16]

Чтобы быть применимой в сфере криминалистики, система кластеризации изображений лиц должна обладать следующими качествами:

- способность делать правильные предсказания в условиях практически полного отсутствия информации о характеристиках лиц на

фотографиях (заранее неизвестное количество личностей, разное число имеющихся лиц для каждого человека, меняющиеся позы и ориентация лиц и т.д.);

- быстродействие при работе на CPU. Данный критерий связан с практически повсеместной невозможностью использования криминалистами графических ускорителей при работе с изображениями (по данным компании Belkasoft).

Именно на эти ключевые факторы и будет обращаться внимание в первую очередь при оценке систем кластеризации лиц в рамках данной работы.

Работа над данным исследованием проводится совместно с компанией Belkasoft, специализирующейся на создании программного обеспечения в сфере киберкриминалистики. Специалисты компании оказывают помощь в сборе необходимых данных и установлении контактов с представителями индустрии. Также работа над задачей осуществляется совместно с Федором Жилкиным, который в рамках этого исследования занимается обнаружением лиц (face detection) на фотографиях.

1. Постановка задачи

Целью данной работы является анализ и сравнение существующих решений задачи кластеризации лиц людей на фотографиях на основе двух ключевых факторов, описанных в предыдущем разделе. Также по результатам исследования будет создана система кластеризации лиц, удовлетворяющая приведенным выше условиям.

Для достижения данной цели были поставлены следующие задачи:

- рассмотреть существующие архитектуры систем кластеризации лиц;
- выбрать, возможно модифицировав, архитектуру, наиболее удовлетворяющую приведенным выше критериям;
- изучить применимые для работы с изображениями лиц алгоритмы кластеризации;
- создать систему кластеризации лиц на основе выбранных алгоритмов и решений;
- протестировать решения на наборе данных Labeled Faces in the Wild [12] и сравнить результаты на основе приведенных выше ключевых факторов.

2. Обзор

2.1. Конвейер для кластеризации лиц

Первым шагом в решении задачи кластеризации лиц является обнаружение их на входном изображении. Этим этапом обработки фотографий занимается в рамках курсовой работы Федор Жилкин, поэтому в данном обзоре будут рассмотрены только шаги кластеризации лиц, следующие за ним (то есть обработка изображений извлеченных лиц).

Для кластеризации лиц необходимо выполнить несколько операций: представить данные в удобном для работы формате, извлечь некоторые особые характеристики (features) лица, уменьшить размерность данных и, наконец, применить сам алгоритм кластеризации. Распространенным способом оформления данного процесса является конвейер распознавания визуальных образов (visual pattern recognition pipeline). Данный подход был рассмотрен еще в 1972 году в [14], и, хотя с тех пор прошло уже почти 50 лет, концептуально он не изменился и сегодня. Устройство конвейера представлено на рис. 2.

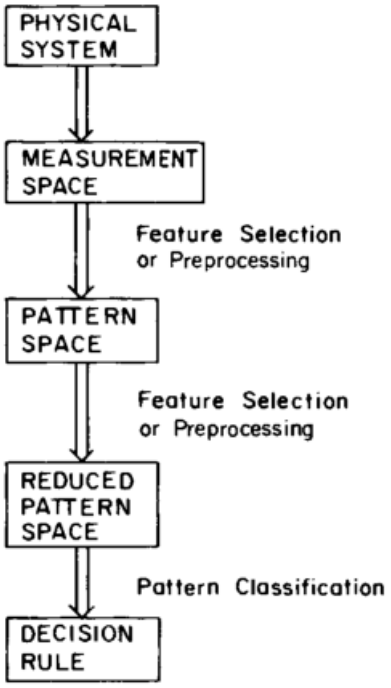

Рис. 2: Конвейер распознавания визуальных образов [14]

В процессе обработки изображений лиц будут выполняться следующие шаги:

- извлекается математическое представление фотографии — матрица пикселей изображения;
- извлекаются характеристики изображения лица, представляющие собой многомерный вектор из некоторого пространства характеристик, а также уменьшается размерность данных;
- используются алгоритмы кластеризации, отбирающие в один кластер векторы, находящиеся «рядом» в пространстве характеристик.

2.2. Создание векторных представлений изображений лиц

2.2.1. FaceNet

В относительно недавней статье [20] группа исследователей из Google Inc. предложила метод извлечения характеристик лица FaceNet, при котором каждому изображению сопоставляется 128-мерный вектор из евклидового пространства. Таким образом, евклидово расстояние между двумя векторами является мерой схожести соответствующих им лиц.

Чтобы добиться такого результата, была использована глубокая нейронная сеть, обученная с использованием функции потерь на основе троек (triplet-based loss function). Эта функция принимает в качестве аргументов представления трех изображений: «якорное» изображение некоторого человека ($f(x_i^a)$), еще одну фотографию с тем же лицом ($f(x_i^p)$) и изображение другого человека ($f(x_i^n)$). Чтобы получить нужный результат, необходимо в процессе обучения добиться выполнения следующего условия:

$$\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 + \alpha < \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2$$
$$\forall \left(f\left(x_i^a\right), f\left(x_i^p\right), f\left(x_i^n\right) \right) \in \mathcal{T} \quad , \tag{1}$$

где α – «зазор» между парами фотографий одного человека и парами изображений разных людей, $\mathcal{T}$ – набор всех возможных троек.

Так как выполнение данного условия для всех существующих троек требует слишком сложного вычислительного процесса, в набор данных для обучения входят только вручную отобранные тройки, для которых это неравенство нарушается.

Данный метод крайне удобен тем, что позволяет использовать классические алгоритмы кластеризации (например, k-means) без дополнительных модификаций.

Более того, попытка сделать это предпринималась B.E.Jouke [3]. В данном исследовании применялась несколько модифицированная версия алгоритма FaceNet – использовались 512-мерные векторы характеристик лиц, вместо 128-мерных. Было проведено сравнение алгоритмов кластеризации K-means [13], Mean Shift [4], Threshold Clustering [20], DBSCAN [6] и уже упомянутого выше Approximate Rank-Order [16].

Результаты, полученные на наборе данных Labeled Faces in the Wild (LFW) [12], представлены в таблице 1.

clustering method	f-measure	amount of clusters	false positives	precision	recall
known clustering	1.00	4935	0	1.00	1.00
Threshold clustering	0.21	11578	130	0.99	0.12
Mean shift	0.14	12759	47	0.99	0.08
DBSCAN	0.18	12652	130	0.99	0.09
App. Rank-Order	0.13	9642	702	0.96	0.07

Таблица 1: Результаты работы разных алгоритмов кластеризации на наборе данных LFW [3]

2.2.2. OpenFace

В 2016 году в университете Карнеги-Меллона была предложена [1] архитектура системы для создания векторных представлений лиц OpenFace, ориентированная на работу в реальном времени, в том числе на мобильных устройствах, при относительно малом количестве доступных данных (фотографий) для обучения.

При разработке системы исследователи ориентировались на подход, использованный в FaceNet [20]. Была использована глубокая нейронная сеть той же архитектуры, при обучении применялась функция потерь на основе троек.

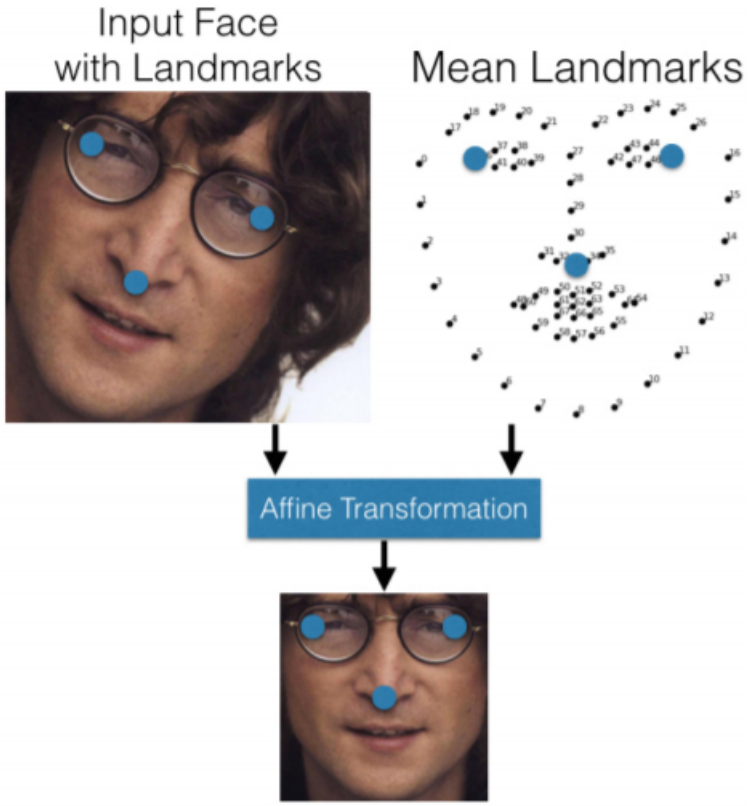

Рис. 3: Процесс выравнивания изображения лица, используемый создателями OpenFace [1]

В то же время, чтобы компенсировать небольшой размер набора данных для обучения, проводится дополнительное выравнивание изображения лица. С помощью детектора ключевых точек лица (face

landmark detector) библиотеки dlib[1] на изображении обнаруживаются точки, соответствующие углам глаз и носу человека на фотографии. Затем они аффинными преобразованиями выравниваются в соответствии с нормализованным представлением лица. После этого размер изображения изменяется до 96x96 пикселей. Визуализацию данного процесса можно увидеть на рис. 3.

Используя данную систему, исследователям удалось добиться на наборе данных LFW [12] сравнимых с FaceNet результатов в задачах верификации и классификации лиц. Результаты применения OpenFace для решения задач кластеризации не были описаны.

2.2.3. Visual Geometry Group (VGG)

В 2015 году исследователи группы визуальной геометрии (Visual Geometry Group) Оксфордского университета, широко известной своими исследованиями в области применения глубокого обучения для распознавания изображений, выпустили статью [17], в которой предложили подход к созданию наборов данных для обучения глубоких нейронных сетей для распознавания лиц.

Помимо этого подхода был опубликован набор данных, содержащий более двух миллионов изображений лиц, архитектура сверточной нейронной сети (позже получившая название VGGFace-16) для создания векторных представлений лиц и методика ее обучения. Последняя представляет особенный интерес, так как она кардинально отличается от предложенной создателями FaceNet.

Вместо того, чтобы обучать нейронную сеть, используя функцию потерь на основе троек, сначала обучается классификатор, способный распознать 2622 личности из набора данных для обучения. Затем последний «классифицирующий» слой удаляется, и дескрипторы, являющиеся выходами предыдущего слоя, используются для распознавания лиц вычислением евклидового расстояния между ними.

Однако, так как при обучении на дескрипторы не накладывалось

[1] `http://dlib.net/` – домашняя страница библиотеки dlib (дата обращения – 14.05.2020)

никаких специальных ограничений, на данном этапе результат в некоторых случаях все еще может оказаться неудовлетворительным. Поэтому предлагается калибровать модель, используя функцию потерь на основе троек.

Таким образом исследователям удалось сделать процесс обучения модели для создания векторных представлений лиц более простым и эффективным по времени.

Используя данный подход к обучению, на наборе данных, собранном исследователями по предложенной ими методике, была обучена сверточная нейронная сеть. Эта модель в решении задачи верификации на наборе данных LFW [12] смогла достичь результатов, сравнимых с лучшими на тот на момент.

В 2017 году была опубликована [21] новая версия набора данных с изображениями лиц людей от той же группы исследователей. На нем были обучены нейронные сети архитектуры ResNet-50 [5] и SeNet-50 [10], превзошедшие лучшие на тот момент результаты в задачах верификации и классификации. В то же время тестирование данных моделей для решения задачи кластеризации лиц не проводилось.

2.3. Кластеризация изображений лиц

2.3.1. K-means

Метод K-средних (K-means) [13] является одним из старейших методов векторной кластеризации, что не мешает ему до сих пор пользоваться большой популярностью. Данный алгоритм стремится распределить наблюдения (d-мерные векторы) в k кластеров так, чтобы минимизировать суммарное квадратичное отклонение этих наблюдений от центров этих кластеров.

Более формально: имеется набор наблюдений $(x_1, x_2, ..., x_n)$, где каждое наблюдение представляет собой d-мерный вещественный вектор. Необходимо разбить эти n наблюдений на $k(\leq n)$ множеств $C = \{C_1, C_2, ..., C_k\}$ так, чтобы минимизировать

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right), \tag{2}$$

где μ_j – среднее наблюдений, лежащих в кластере C_j.

Далее выполняются два шага:

- **шаг инициализации**, когда каждое наблюдение помещается в кластер с ближайшим к нему средним.
- **шаг обновления**, когда среднее каждого кластера пересчитывается по формуле

$$\mu_i^{(t+1)} = \frac{1}{\left| C_i^{(t)} \right|} \sum_{x_j \in C_i^{(t)}} x_j \tag{3}$$

Данные шаги продолжают повторяться, пока на этапе инициализации происходят изменения. В противном случае алгоритм останавливается.

2.3.2. Mean Shift

Сдвиг среднего (Mean Shift) [4] – алгоритм, стремящийся найти области повышенной плотности в имеющемся множестве наблюдений.

На каждой итерации алгоритма для каждой области находится ее центр масс (центроид), являющийся средним значений принадлежащих ей точек. Затем эти центры фильтруются на этапе постобработки, устраняя слишком близкие точки, и формируется итоговый набор центроидов.

Более формально: пусть x_i – потенциальный центр масс на итерации t. Тогда для итерации $t+1$ значение обновляется следующим образом:

$$x_i^{t+1} = m\left(x_i^t\right). \tag{4}$$

Здесь m – вектор сдвига среднего, направленный в сторону максимального увеличения плотности наблюдений. Он вычисляется следующим образом:

$$m\left(x_i\right) = \frac{\sum_{x_j \in N(x_i)} K\left(x_j - x_i\right) x_j}{\sum_{x_j \in N(x_i)} K\left(x_j - x_i\right)}, \tag{5}$$

где $N(x_i)$ – множество соседних x_i векторов, находящиеся не дальше некоторого заданного расстояния, а $K(x_j - x_i)$ – ядерная функция, определяющая вес ближайших точек для переоценки центра масс.

Процесс обновления центров масс происходит, пока $m(x)$ не сойдется.

2.3.3. Chinese Whispers

Алгоритм Chinese Whispers [2] является рандомизированным методом кластеризации взвешенных неориентированных графов, получившим свое название от игры, русскоязычным вариантом которой является "сломанный телефон". Данный алгоритм впервые был придуман для решения задач обработки естественного языка, но его возможно применить и для кластеризации изображений лиц.

Идея алгоритма крайне проста и описывается в следующих этапах:

1. Каждая вершина графа помещается в отдельный кластер.
2. Выбирается случайный порядок обхода вершин графа.
3. Каждая вершина определяется в кластер вершины-соседа, имеющей наибольший ранг. Если вершин с наибольшим рангом несколько, случайным образом выбирается одна из них.

Рангом соседней вершины здесь называется сумма весов ребер, соединяющих рассматриваемую вершину с вершинами из кластера ее соседа.

Данный метод является итеративным, поэтому второй и третий шаг повторяются либо до тех пор, пока изменения в структуре кластеров не станут незначительными, либо в течение заданного числа итераций.

Для применения данного алгоритма для решения задачи кластеризации изображений лиц, необходимо построить полный граф, где вершины – векторные представления лиц, а веса ребер – расстояния между ними. Затем данный граф кластеризуется с помощью Chinese Whispers. При этом необходимо указать некоторое пороговое значение, при превышении которого вершине переназначается кластер, чтобы избежать объединения в один класс лиц разных людей.

2.3.4. DBSCAN

Основанная на плотности пространственная кластеризация для приложений с шумами (Density-based spatial clustering of applications with noise или DBSCAN) [6] – алгоритм кластеризации, основанный на поиске областей с высокой плотностью точек и помечающий наблюдения, попавшие в область с маленьким числом соседей, как выбросы. Данный алгоритм был опубликован еще в 1996 году, но до сих пор является одним из самых популярных и упоминаемых в научной литературе алгоритмов кластеризации.

Центральными в DBSCAN являются понятия трех видов точек:

1. Основные точки (core points) – находятся в области высокой плотности (имеют много соседей) на малом расстоянии друг от друга; они и образуют кластер.

2. Пограничные точки (boundary points) – наблюдения из области, имеющей низкую плотность, но при это находящиеся близко к основным точкам; попадают в кластер к ближайшей основной точке.

3. Шум (noise points) – точки в области низкой плотности, расположенные далеко от основных точек; не попадают в кластер и помечаются как выбросы.

Для работы алгоритма необходимо задать два параметра: число ϵ, определяющее окрестность, в которой происходит поиск соседей, и минимальное число точек $MinPts$, необходимое для образования кластера.

В общем виде ход работы алгоритма можно описать следующими шагами:

1. Рассматривается ϵ-окрестность каждой точки, выделяются основные точки с не менее чем $MinPts$ соседями.

2. Компоненты связности основных точек образуют основу кластеров.

3. Каждая пограничная точка назначается в кластер к ближайшей основной точке из ϵ-окрестности. Если таких нет, точка считается шумом.

2.3.5. Threshold Clustering

Алгоритм кластеризации по пороговому значению (Threshold Clustering), описанный в [3], является наивным прямолинейным подходом к решению задачи кластеризации изображений лиц.

На каждой итерации между векторным представлением еще не рассмотренного лица и уже распределенными по кластерам лицами считается евклидово расстояние. Если расстояние до некоторого вектора характеристик оказывается меньше выбранного значения порога t, текущее векторное представление помещается в его кластер. В противном случае, если таких лиц не нашлось, создается новый кластер и рассматриваемое представление попадает в него.

Таким образом, выбор численного значения порога t имеет огромное значение. Если оно слишком мало, небольшое количество лиц будут попадать в один и тот же кластер и следовательно изображения одного и того же человека будут определены как принадлежащие разным личностям. В то же время слишком большое пороговое значение приведет к тому, что лица разных людей часто будут оказываться в одном кластере.

2.3.6. Rank-Order Clustering

В 2011 году группа исследователей из Microsoft Research Asia и Университета Цинхуа предложила подход к кластеризации изображений лиц [23]. Ими был представлен алгоритм кластеризации лиц на основе ранжированного расстояния (Rank-Order distance). Его принцип основан на наблюдении, что изображения лица одного и того же человека, как правило, имеют одинаковых ближайших «соседей» в терминах абсолютного расстояния (манхеттенское или евклидово) между векторами характеристик, соответствующих этим фотографиям. Для извлече-

ния характеристик лица используется метод, описанный в [8].

Пусть имеются два изображения лица a и b. Создаются два списка лиц O_a и O_b, отсортированных согласно абсолютному расстоянию до соответствующего рассматриваемого лица. Определим ранжированное состояние $D(a,b)$ между лицами a и b:

$$D(a,b) = \sum_{i=0}^{O_a(b)} O_b\left(f_a(i)\right), \tag{6}$$

где $f_a(i)$ – i-ое лицо в списке O_a, $O_b(f_a(i)$ – ранг лица $f_a(i)$ в списке O_b.

Данное расстояние асимметрично. Нормализуем и приведем его к симметричной форме:

$$D^R(a,b) = \frac{D(a,b) + D(b,a)}{\min\left(O_a(b), O_b(a)\right)} \tag{7}$$

Далее исследователи определяют кластерное ранжированное расстояние (cluster-level rank-order distance) и кластерное нормализованное расстояние (cluster-level normalized distance). Сначала определяется абсолютное расстояние между кластерами:

$$d\left(C_i, C_j\right) = \min_{\forall a \in C_i, b \in C_j} d(a,b), \tag{8}$$

где a и b – лица из кластеров C_i и C_j соответственно. Тогда кластерное ранжированное расстояние определяется, как:

$$D^R\left(C_i, C_j\right) = \frac{D\left(C_i, C_j\right) + D\left(C_j, C_i\right)}{\min\left(O_{C_i}\left(C_j\right), O_{C_j}\left(C_i\right)\right)}, \tag{9}$$

где O_{C_i} и O_{C_j} – ранги кластеров, $D(C_i, C_j)$ и $D(C_j, C_i)$ – вычисляются, как (1), только на уровне кластеров.

Наконец, кластерное нормализованное расстояние определяется так:

$$\begin{aligned} D^N\left(C_i, C_j\right) &= \tfrac{1}{\phi(C_i, C_j)} \cdot d\left(C_i, C_j\right) \\ \phi\left(C_i, C_j\right) &= \tfrac{1}{|C_i|+|C_j|} \textstyle\sum_{a \in C_i \cup C_j} \tfrac{1}{K} \sum_{k=1}^{K} d\left(a, f_a(k)\right), \end{aligned} \tag{10}$$

где $\phi(C_i, C_j)$ – среднее расстояние между лицами в двух кластерах по

их первым K соседям, K – константа.

На основе этих понятий предлагается сам алгоритм кластеризации:

1. Пусть каждое лицо является отдельным кластером.
2. Объединить два кластера, если ранжированное и нормализованное расстояние между ними меньше некоторого порога.
3. Остановиться, если никакие кластеры больше не могут быть объединены. Иначе обновить кластеры и расстояния и вернуться к пункту 2.

Используя данный алгоритм, исследователям удалось достичь следующих результатов:

- precision – 0.98, recall – 0.87 на альбоме Easyalbum [7];
- precision – 0.97, recall – 0.85 на альбоме Gallagher [9].

2.3.7. Approximate Rank-Order Clustering

В 2018 году группа исследователей из университета штата Мичиган опубликовали модификацию описанного в предыдущем разделе алгоритма – приближенную ранжированную кластеризацию (Approximate Rank-Order Clustering) [16].

Создатели данного алгоритма попытались решить проблему масштабирования Rank-Order Clustering, связанную с необходимостью рассчитывать для каждого лица список его ранжированных соседей, что при реализации напрямую имеет временную сложность $O(n^2)$. Было предложено вместо этого брать лишь k его ближайших соседей, рассчитанных с помощью алгоритма на основе рандомизированных K-d деревьев (Randomized k-d Tree)[15].

Далее исследователи предлагают в выражении (6) брать сумму значений характеристической функции множества, состоящего из k ближайших соседей данного лица. Таким образом, выражение приобретает вид:

$$d_m(a,b) = \sum_{i=1}^{\min(O_a(b),k)} I_b\left(O_b\left(f_a(i)\right),k\right), \tag{11}$$

где $I_b(x,k)$ – функция-индикатор, принимающая значение 0, если лицо x попадает в k ближайших соседей лица b, и значение 1 в противном случае.

Нормализованное расстояние (7) выглядит следующим образом:

$$D_m(a,b) = \frac{d_m(a,b) + d_m(b,a)}{\min\left(O_a(b), O_b(a)\right)}. \tag{12}$$

В остальном алгоритм следует идеям, описанным в [23].

3. Эксперимент

3.0.1. Labeled Faces in the Wild

В данной работе для оценки работы алгоритмов кластеризации используется набор данных Labeled Faces in the Wild (LFW) [12], собранный и опубликованный исследователями из университета Массачусетса в 2007 году. С тех пор он успел стать стандартом в области экспериментов по идентификации и верификации лиц людей. И, хотя соответствующей процедуры для него не было формализовано, его часто используют в тех немногих работах по кластеризации изображений лиц, что вообще публикуются.

"In the Wild" в названии набора данных означает, что подобранные в нем фотографии стремятся как можно точнее отражать данные, встречающиеся в повседневных задачах распознавания лиц. Он содержит более 13000 фотографий 5749 человек. При этом их изображения распределены неравномерно – для 1680 личностей имеется 2 и более фотографии, для остальных – только одна. В то же время в наборе данных варьируется положение лица в кадре: от людей, смотрящих прямо в кадр, до фотографий в профиль, под углом, в головных уборах, очках и т.д.

Все это делает LFW подходящим набором данных для проведения экспериментов в области кластеризации изображений лиц.

3.0.2. Pairwise precision, recall, f1-score

В качестве метрики для оценки результатов работы алгоритмов кластеризации лиц были выбраны парные (pairwise) precision, recall и f-мера. Для того, чтобы описать их устройство, необходимо ввести несколько определений:

1. Правильным положительным предсказанием (True Positive, TP) будем называть пары лиц, принадлежащие одному и тому же человеку, и при этом определенные в один кластер.

2. Правильным негативным предсказанием (True Negative, TN) называются пары лиц, принадлежащие разным людям, и определенные в разные кластеры.

3. Неправильным положительным предсказанием (False Positive, FP) будем называть пары лиц, принадлежащие разным людям, но определенные в один кластер.

4. Неправильным отрицательным предсказанием (False Negative, FN) называются пары лиц, принадлежащие одному и тому же человеку, но определенные в разные кластеры.

Таким образом мы можем задать парные precision, recall и f-меру:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F\text{-}measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{15}$$

3.0.3. Реализация

В рамках данной работы был реализован конвейер кластеризации лиц, состоящий из следующих модулей:

1. **Модуль извлечения лиц (Face Extraction)**. Отвечает за обнаружение лиц на фотографиях и их нормализацию путем обрезания и выравнивания изображения. В качестве детекторов лиц были использованы реализации MTCNN [11] от Дэвида Сэндберга[2] и HOG-детектор [22] библиотеки dlib. Для нормализации изображений использованы библиотеки OpenCV[3] и dlib.

[2] `https://github.com/davidsandberg/facenet` – репозиторий Дэвида Сэндберга с реализациями MTCNN и FaceNet (дата обращения – 18.05.2020)

[3] `https://opencv.org/` – домашняя страница библиотеки OpenCV (дата обращения – 18.05.2020)

2. **Модуль создания векторных представлений (Face Embeddings)**. Отвечает за извлечение характеристик лиц и уменьшение размерности данных. Были использованы библиотеки Keras[4] и TensorFlow[5] вместе с реализациями моделей FaceNet, OpenFace[6], VGGFace-16[7], VGGFace-resnet50, VGGFace-senet50.

3. **Модуль кластеризации векторных представлений**. Содержит реализации DBSCAN, Mean Shift, K-means из библиотеки Scikit-Learn[8], Chinese Whispers из библиотеки dlib. Алгоритмы Threshold Clustering, Rank-Order Clustering и Approximate Rank-Order Clustering были реализованы с использованием библиотек Numpy[9] и Pyflann[10].

Система реализована на языке программирования Python. Ее диаграмму классов UML можно увидеть на рис. 4.

3.0.4. Опыты

Все опыты, описанные в данном разделе проводились на компьютере со следующими спецификациями: Intel Core i5-6200U CPU @ 2.30GHz, 8 ГБ ОЗУ, ОС Windows 10 Pro.

Среднее время создания векторных представлений лиц. Замеры времени работы реализаций моделей для создания векторных представлений проходили следующим образом: были взяты все 13233 изображений людей размером 250x250 пикселей из набора данных LFW, затем каждый выбранный алгоритм отработал на этих фотографиях,

[4] `https://keras.io/` – домашняя страница библиотеки Keras (дата обращения – 20.05.2020)

[5] `https://www.tensorflow.org/` – домашняя страница библиотеки TensorFlow (дата обращения – 20.05.2020)

[6] `https://cmusatyalab.github.io/openface/` домашняя страница проекта OpenFace (дата обращения – 18.05.2020)

[7] `https://github.com/rcmalli/keras-vggface` – репозиторий, содержащий реализации VGGFace-16, VGGFace-resnet50, VGGFace-senet50 (дата обращения – 18.05.2020)

[8] `https://scikit-learn.org/` – домашняя страница библиотеки Scikit-Learn (дата обращения – 18.05.2020)

[9] `https://numpy.org/` – домашняя страница библиотеки Numpy (дата обращения – 18.05.2020)

[10] `https://github.com/primetang/pyflann` – репозиторий проекта Pyflann (дата обращения – 18.05.2020)

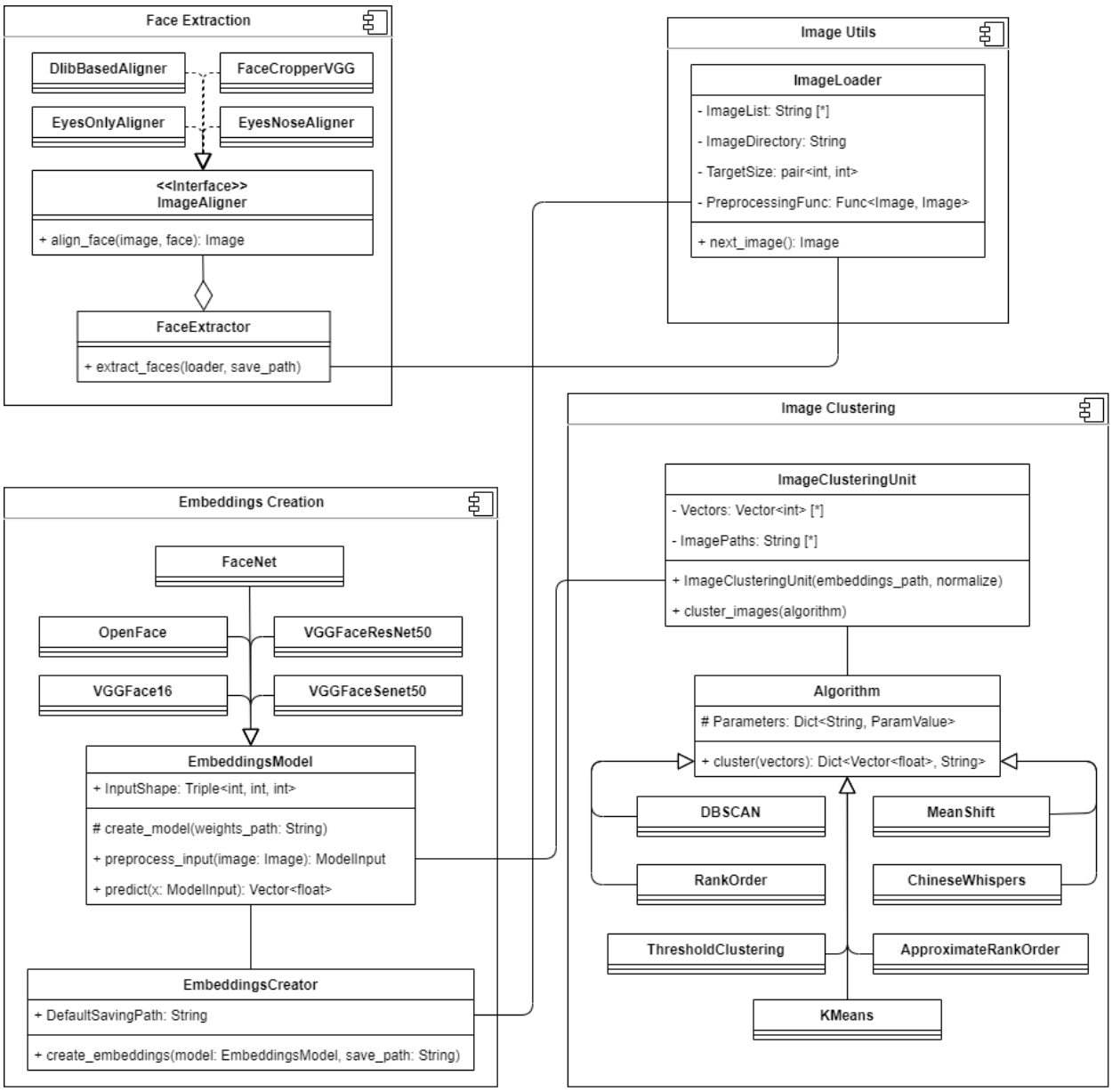

Рис. 4: Система кластеризации лиц

время работы на каждой итерации измерялось. После этого было посчитано среднее время работы (в секундах) для каждой модели и среднее квадратичное отклонение. Результаты можно увидеть в таблице 2.

Результаты на наборе данных LFW. Все опыты проводились на всем наборе данных LFW. Лица на изображениях были предварительно извлечены и выравнены с помощью детектора MTCNN. Для модели OpenFace дополнительно проводилось выравнивание на основе 68 landmarks detector библиотеки dlib. Затем создавались векторные пред-

Модель	Среднее время работы (в секундах)	Среднее квадратичное отклонение времени работы (в секундах)
FaceNet	0.078	0.060
OpenFace	0.019	0.017
VGGFace-16	0.312	0.015
VGGFace-ResNet50	0.223	0.027
VGGFace-SeNet50	0.246	0.034

Таблица 2: Результаты замеров среднего времени работы алгоритмов создания векторных представлений

ставления лиц и проводилась их кластеризация с измерением ключевых метрик: парные precision, recall, f-мера, общее время работы алгоритма. Параметры подбирались «на глаз» так, чтобы получить максимальное значение F-меры.

FaceNet. В данном опыте для каждого изображения, предварительно обработанного, из набора данных LFW было создано 128-мерное векторное представление с помощью реализации FaceNet. Для данного опыта были выбраны следующие параметры алгоритмов кластеризации:

- порог 0.72 для кластеризации по пороговому значению;
- порог 0.74 и число итераций 100 для Chinese Whispers;
- ϵ 0.66 и $MinPts$ 1 для DBSCAN;
- ширина пропускания 0.7 для Mean Shift;
- число кластеров 5320 для K-means;
- значение K 820 и порог 136.7 для Rank-Order Clustering;
- количество соседей 855 и порог 0.17 для Approximate Rank-Order.

Результаты представлены в таблице 3.

Алгоритм	Precision	Recall	F-мера	Общее время работы
Threshold Clustering	0.95	0.87	0.90	00:14:40
DBSCAN	0.94	0.88	0.91	00:01:22
K-means	0.85	0.11	0.19	00:17:26
Chinese Whispers	0.96	0.93	0.94	00:00:40
Mean Shift	0.93	0.94	0.93	00:03:54
Rank-Order	0.82	0.79	0.81	05:43:11
App. Rank-Order	0.83	0.84	0.84	02:42:51

Таблица 3: Результаты кластеризации векторных представлений, полученных с помощью FaceNet

OpenFace. В данном опыте для каждого изображения, предварительно обработанного, из набора данных LFW было создано 128-мерное векторное представление с помощью реализации OpenFace. Для данного опыта были выбраны следующие параметры алгоритмов кластеризации:

- порог 0.48 для кластеризации по пороговому значению;
- порог 0.52 и число итераций 100 для Chinese Whispers;
- ϵ 0.48 и $MinPts$ 1 для DBSCAN;
- ширина пропускания 0.54 для Mean Shift;
- число кластеров 5300 для K-means;
- значение K 810 и порог 136.7 для Rank-Order Clustering;
- количество соседей 860 и порог 0.18 для Approximate Rank-Order.

Результаты представлены в таблице 4.

Алгоритм	Precision	Recall	F-мера	Общее время работы
Threshold Clustering	0.76	0.30	0.43	00:24:17
DBSCAN	0.53	0.49	0.51	00:00:59
K-means	0.43	0.04	0.07	00:12:41
Chinese Whispers	0.72	0.53	0.61	00:00:38
Mean Shift	0.66	0.45	0.53	00:06:27
Rank-Order	0.61	0.48	0.54	05:07:43
App. Rank-Order	0.70	0.48	0.58	02:14:26

Таблица 4: Результаты кластеризации векторных представлений, полученных с помощью OpenFace

VGGFace-16. В данном опыте для каждого изображения, предварительно обработанного, из набора данных LFW было создано 512-мерное векторное представление с помощью реализации VGGFace-16. Для данного опыта были выбраны следующие параметры алгоритмов кластеризации:

- порог 0.72 для кластеризации по пороговому значению;
- порог 0.62 и число итераций 100 для Chinese Whispers;
- ϵ 0.58 и $MinPts$ 1 для DBSCAN;
- ширина пропускания 0.54 для Mean Shift;
- число кластеров 5320 для K-means;
- значение 770 K и порог 134.2 для Rank-Order Clustering;
- количество 880 соседей 0.19 и порог для Approximate Rank-Order.

Результаты представлены в таблице 5.

VGGFace-SeNet50. В данном опыте для каждого изображения, предварительно обработанного, из набора данных LFW было создано

Алгоритм	Precision	Recall	F-мера	Общее время работы
Threshold Clustering	0.97	0.72	0.83	00:14:09
DBSCAN	0.92	0.84	0.88	00:08:56
K-means	0.72	0.07	0.13	01:09:20
Chinese Whispers	0.97	0.88	0.92	00:07:40
Mean Shift	0.96	0.77	0.86	00:25:17
Rank-Order	0.81	0.71	0.76	07:21:05
App. Rank-Order	0.85	0.74	0.79	02:57:52

Таблица 5: Результаты кластеризации векторных представлений, полученных с помощью VGGFace-16

2048-мерное векторное представление с помощью реализации VGGFace-SeNet50. Для данного опыта были выбраны следующие параметры алгоритмов кластеризации:

- порог 0.64 для кластеризации по пороговому значению;
- порог 0.7 и число итераций 100 для Chinese Whispers;
- ϵ 0.66 и $MinPts$ 1 для DBSCAN;
- ширина пропускания 0.61 для Mean Shift;
- число кластеров 5320 для K-means;
- значение K 800 и порог 142.3 для Rank-Order Clustering;
- количество соседей 850 и порог 0.2 для Approximate Rank-Order.

Результаты представлены в таблице 6.

VGGFace-ResNet50. В данном опыте для каждого изображения, предварительно обработанного, из набора данных LFW было создано 2048-мерное векторное представление с помощью реализации VGGFace-ResNet50. Для данного опыта были выбраны следующие параметры алгоритмов кластеризации:

Алгоритм	Precision	Recall	F-мера	Общее время работы
Threshold Clustering	0.97	0.89	0.94	00:32:27
DBSCAN	0.99	0.93	0.96	00:34:48
K-means	0.86	0.10	0.19	04:08:02
Chinese Whispers	0.98	0.97	0.97	00:25:53
Mean Shift	0.99	0.94	0.97	01:37:25
Rank-Order	0.88	0.71	0.79	10:04:27
App. Rank-Order	0.94	0.74	0.82	03:28:10

Таблица 6: Результаты кластеризации векторных представлений, полученных с помощью VGGFace-SeNet50

- порог 0.72 для кластеризации по пороговому значению;
- порог 0.77 и число итераций 100 для Chinese Whispers;
- ϵ 0.72 и $MinPts$ 1 для DBSCAN;
- ширина пропускания 0.68 для Mean Shift;
- число кластеров 5320 для K-means;
- значение K 780 и порог 153.7 для Rank-Order Clustering;
- количество соседей 880 и порог 0.18 для Approximate Rank-Order.

Результаты представлены в таблице 7.

Выводы. На основе описанных выше результатов можно сделать ряд выводов. Во-первых, модель FaceNet показала себя лучше всего с точки зрения соотношения качества результата и скорости работы. OpenFace показала ощутимый прирост быстродействия по сравнению с FaceNet, но ценой этому является существенное понижение других ключевых метрик. VGGFace-16 в целом показала себя немного хуже FaceNet, как в плане метрик, так и с точки зрения быстродействия. VGGFace-ResNet50 и VGGFace-SeNet50, как и ожидалось, смогли показать лучшее качество

Алгоритм	Precision	Recall	F-мера	Общее время работы
Threshold Clustering	0.98	0.89	0.93	00:30:36
DBSCAN	0.99	0.93	0.96	00:35:25
K-means	0.85	0.09	0.18	04:13:41
Chinese Whispers	0.99	0.96	0.98	00:25:11
Mean Shift	0.99	0.92	0.95	01:39:01
Rank-Order	0.90	0.72	0.81	10:21:17
App. Rank-Order	0.98	0.72	0.83	03:36:43

Таблица 7: Результаты кластеризации векторных представлений, полученных с помощью VGGFace-ResNet50

работы по сравнению с FaceNet, OpenFace и VGGFace-16, но они также показали и увеличившееся в разы время работы, что может сделать их неприменимыми для обработки большего количества данных.

Из алгоритмов кластеризации хуже всего показал себя широко распространенный алгоритм K-means. Мало того, что для работы он требует указания итогового количества кластеров (а эта информация редко известна заранее), алгоритм не смог показать удовлетворительные результаты на наборе данных LFW и потребовал при этом слишком много времени для завершения. DBSCAN, Chinese Whispers, Mean Shift и Threshold Clustering показали сравнимые результаты, лучшие из имеющихся, но алгоритм Chinese Whispers оказался ощутимо быстрее остальных. Отдельно стоит отметить, что Threshold Clustering, простой и прямолинейный подход к кластеризации лиц, оказался практически на одном уровне с другими лидерами. Rank-Order Clustering и Approximate Rank-Order Clustering показали результаты хуже лидеров как с точки зрения ключевых метрик, так и в плане времени работы.

Границы применимости. Нахождение лучших результатов, которые алгоритмы кластеризации способны показать на наборе данных LFW, требует исчерпывающего перебора параметров. Однако для этого необходимы огромные вычислительные мощности и большое количе-

ство времени на проведение экспериментов, что не соответствует масштабу данной работы. Поэтому представленные здесь результаты могут оказаться не лучшими возможными, но в то же время они способны проиллюстрировать потенциал алгоритмов, некоторые их слабые и сильные стороны. Отдельно стоит упомянуть про алгоритмы Rank-Order Clustering и Approximate Rank-Order Clustering, реализации которых представляют собой прямое переложение подходов, описанных в соответствующих статьях. Это означает, что никакие дополнительные оптимизации не были использованы, а это может обуславливать их существенно большее время работы по сравнению с библиотечными реализациями других алгоритмов.

Заключение

Таким образом, в рамках данной работы:

- был выполнен обзор алгоритмов для создания векторных представлений лиц на основе моделей FaceNet, OpenFace, VGGFace-16, VGGFace-ResNet50, VGGFace-SeNet50;
- был выполнен обзор алгоритмов Chinese Whispers, Mean Shift, K-means, DBSCAN, Threshold Clustering, Rank-Order Clustering и Approximate Rank-Order Clustering для кластеризации векторных представлений;
- реализован конвейер кластеризации лиц[11];
- проведены замеры среднего времени работы алгоритмов создания векторных представлений;
- проведено сравнение результатов работы выбранных алгоритмов кластеризации на наборе данных LFW.

Отдельно хотелось бы выразить благодарность специалистам компании Belkasoft, в частности **Никите Тимофееву** и **Михаилу Виноградову**, за неоценимую помощь в установлении контактов с представителями индустрии и работе над исследованием.

[11] `https://github.com/OlegChern/Face-Clustering-Test-System` – репозиторий с проектом (дата обращения – 18.05.2020)

Список литературы

[1] Amos Brandon, Ludwiczuk Bartosz, Satyanarayanan Mahadev. OpenFace: A general-purpose face recognition library with mobile applications. — 2016.

[2] Biemann Chris. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems // Proceedings of TextGraphs. — 2006. — 07. — P. 73–80.

[3] Bijl Erik Jouke. A comparison of clustering algorithms for face clustering. — University of Groningen, технический отчёт, 2018. — `http://fse.studenttheses.ub.rug.nl/18064/1/Report_research_internship.pdf`, дата обращения – 12.12.2019.

[4] Comaniciu D., Meer P. Mean shift: a robust approach toward feature space analysis // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2002. — May. — Vol. 24, no. 5. — P. 603–619.

[5] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep Residual Learning for Image Recognition. — 2015. — 1512.03385.

[6] A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise / Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. — KDD'96. — AAAI Press, 1996. — P. 226–231.

[7] EasyAlbum: An Interactive Photo Annotation System Based on Face Clustering and Re-ranking / Jingyu Cui, Fang Wen, Rong Xiao et al. // Proceedings of the SIGCHI conference on Human factors in computing systems. — Association for Computing Machinery, Inc., 2007. — April.

[8] Face recognition with learning-based descriptor / Z. Cao, Q. Yin, X. Tang, J. Sun // 2010 IEEE Computer Society Conference on

Computer Vision and Pattern Recognition. — 2010. — June. — P. 2707–2714.

[9] Gallagher A., Chen T. Clothing Cosegmentation for Recognizing People // Proc. CVPR. — 2008.

[10] Hu Jie, Shen Li, Sun Gang. Squeeze-and-Excitation Networks // CoRR. — 2017. — Vol. abs/1709.01507. — 1709.01507.

[11] Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks / K. Zhang, Z. Zhang, Z. Li, Y. Qiao // IEEE Signal Processing Letters. — 2016. — Oct. — Vol. 23, no. 10. — P. 1499–1503.

[12] Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments / Gary Huang, Marwan Mattar, Tamara Berg, Eric Learned-Miller // Tech. rep. — 2008. — 10.

[13] Lloyd S. Least squares quantization in PCM // IEEE Transactions on Information Theory. — 1982. — March. — Vol. 28, no. 2. — P. 129–137.

[14] Meisel William S. Computer-oriented approaches to pattern recognition. — 1972.

[15] Muja M., Lowe D. G. Scalable Nearest Neighbor Algorithms for High Dimensional Data // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2014. — Vol. 36, no. 11. — P. 2227–2240.

[16] Otto Charles, Wang Dayong, Jain Anil. Clustering Millions of Faces by Identity // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 2016. — 04. — Vol. PP.

[17] Parkhi Omkar, Vedaldi Andrea, Zisserman Andrew. Deep Face Recognition. — Vol. 1. — 2015. — 01. — P. 41.1–41.12.

[18] A Review of Methods for Face Verification under Illumination Variation / Mehran Emadi, Farhad Navabifar, Marzuki Khalid, Rubiyah Yusof // Proceedings of the International Conference

on Image Processing, Computer Vision, and Pattern Recognition (IPCV) / Citeseer. — 2011. — P. 1.

[19] Roomi Mansoor, Beham Dr.M.Parisa. A Review Of Face Recognition Methods // International Journal of Pattern Recognition and Artificial Intelligence. — 2013. — 04. — Vol. 27.

[20] Schroff F., Kalenichenko D., Philbin J. FaceNet: A unified embedding for face recognition and clustering // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2015. — June. — P. 815–823.

[21] VGGFace2: A dataset for recognising faces across pose and age / Qiong Cao, Li Shen, Weidi Xie et al. // CoRR. — 2017. — Vol. abs/1710.08092. — 1710.08092.

[22] Yang H., Wang X. A. Cascade Face Detection Based on Histograms of Oriented Gradients and Support Vector Machine // 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). — 2015. — P. 766–770.

[23] Zhu C., Wen F., Sun J. A rank-order distance based clustering algorithm for face tagging // CVPR 2011. — 2011. — June. — P. 481–488.