

ИДЕНТИФИКАЦИЯ ЛИЦ В РЕАЛЬНОМ ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ И ХЭШИРУЮЩЕГО ЛЕСА

Ю.В. Визильтер¹, В.С. Горбацевич¹, А.В. Воротников¹, Н.А. Костромов¹

¹Государственный научно-исследовательский институт авиационных систем (ГосНИИАС), Москва, Россия

Аннотация

В работе предлагается новый подход к построению биометрического шаблона с использованием свёрточной нейронной сети и хэширующего леса. Метод состоит из двух этапов: на первом происходит обучение свёрточной нейронной сети, далее к полученным описаниям применяется хэширующее преобразование с использованием нового предложенного метода хэширующего леса. Такой способ формирования шаблона является обобщением метода Boosted SSC (*Similarity Sensitive Coding*) для решения задачи построения оптимального хеша, учитывающего специфику задач верификации и идентификации лиц. Обучение производилось на базе лиц CASIA-WebFace, а тестирование – на базе лиц LFW. Применительно к задачам бинарного хэширования с метрикой Хемминга описанный подход позволяет получать 200-битный (25 байт) биометрический шаблон с качеством верификации 96,3 % и 2000-битный шаблон с 98,14 % на базе изображений лиц LFW. При использовании хэширующего леса с 7-битными деревьями 2000×7 достигается уровень идентификации в 93 % относительно базовых показателей свёрточной нейронной сети в 89,9%. В результате достигается скорость формирования биометрических шаблонов (описаний лиц) с частотой более 40 изображений лиц в секунду на CPU Core i7 и более 120 изображений лиц в секунду с использованием GPU GeForce GTX 650.

Ключевые слова: свёрточные нейронные сети, хэширование, бинарные деревья, метрика Хемминга, биометрия.

Цитирование: Визильтер, Ю.В. Идентификация лиц в реальном времени с использованием свёрточной нейронной сети и хэширующего леса / Ю.В. Визильтер, В.С. Горбацевич, А.В. Воротников, Н.А. Костромов // Компьютерная оптика. – 2017. – Т. 41, № 2. – С. 254-265. – DOI: 10.18287/2412-6179-2017-41-2-254-265.

Введение

На сегодняшний день существует множество методов распознавания лиц, причём при их разработке учитывались различные требования к размеру формируемых шаблонов для сравнения, скорости их формирования и сравнения, к показателям качества распознавания. В данной работе рассматривается задача распознавания лиц в реальном времени в широком смысле – создание семейства методов построения биометрических шаблонов, которое позволит достичь гибкого баланса между всеми перечисленными основными характеристиками. Хорошо известно, что метрика Хемминга для сравнения бинарных шаблонов обеспечивает самый быстрый поиск в базе ([1–12]). С другой стороны, лучшие показатели качества в распознавании лиц демонстрируют глубокие свёрточные нейронные сети, использующие небинарные биометрические шаблоны ([13–21]). Эти подходы можно объединить в особую архитектуру свёрточной нейронной сети (СНС) с бинарным выходным слоем – свёрточную нейронную сеть с хэширующим слоем (СНСХС). Подобная архитектура описана в работе [22], где СНС и слой хэширования обучаются вместе с использованием метода обратного распространения ошибки. Однако хотелось бы иметь такое семейство методов построения биометрических шаблонов, которое позволяло бы непрерывно изменять размер используемого шаблона от крайне компактных (десятки байтов) до шаблонов огромного размера, обеспечивающих наилучшее качество распознавания. В данной работе предлагается объединить СНС и дополнительное преобразование хэшированием, основанное на хэширующем лесу (ХЛ).

ХЛ формирует векторы признаков, закодированные бинарными деревьями. При таком подходе бинарные биометрические шаблоны формируются семейством (лесом) бинарных деревьев. Изменяя глубину деревьев и критерий кодирования, можно получить бинарный биометрический шаблон с требуемыми характеристиками. При этом всё семейство полученных таким образом бинарных шаблонов базируется на одной и той же свёрточной сети.

Мы назвали подобную архитектуру свёрточной нейронной сетью с хэширующим лесом (СНСХЛ). В случае кодирования деревьев глубиной в 1 бит СНСХЛ превращается в СНСХС.

Архитектура используемой базовой сети базируется на Max-Feature-Map (MFM) архитектуре, предложенной Xiang Wu [20].

В данной работе предлагается новый метод хэширования с использованием леса классификаторов – форсированный хэширующий лес (ФХЛ), который объединяет идеи метода Boosted SSC [9–10] для кодирования различий данных и прямую оптимизацию целевой функции в заданном пространстве признаков. Однако в отличие от классического Boosted SSC кодирование происходит лесом бинарных деревьев. Схема работы хэширующего леса и его связь с СНС представлена на рис. 1. В работе также предлагается новая биометрическая целевая функция кодирования для одновременной оптимизации решения задач идентификации и верификации лиц.

Предложенный метод построения биометрических шаблонов СНСХЛ был обучен на публичной базе данных лиц CASIA-WebFace и тестировался на базе

лиц LFW. Тестирование показало, что предлагаемый подход позволяет достичь как снижения размерности биометрического шаблона, так и повышения качества распознавания в задачах верификации и идентификации. В задаче построения бинарных шаблонов с метрикой Хемминга ФХЛ существенно превосходит оригинальный Boosted SSC. Бинарный шаблон, построенный предлагаемым методом размером 200 бит (25 байт), достигает качества распознавания лиц в задаче верификации в 96,3% на базе LFW при 70-кратном выигрыше в скорости сравнения относительно оригинальной сети. Бинарный шаблон размером в 2000 бит показывает результат в 98,14% при аналогичных условиях. Бинарный шаблон с 2000×7-битными хэширующими деревьями достигает 93% в идентификации (rank-1) на LFW относительно результатов базовой сети в 89,9%.

Статья организована следующим образом. Параграф 1 кратко описывает схожие работы. Параграф 2 содержит схему предложенного метода и его реализацию применительно к задаче кодирования лиц. Экспериментальные результаты продемонстрированы в Параграфе 3, после которой излагаются заключение и выводы.

1. Обзор схожих работ

Ранее был предложен целый ряд подходов к формированию описаний (биометрических шаблонов) изображений лиц, которые могут использоваться для биометрической верификации или идентификации персон ([23–25]), однако практически все последние результаты в распознавании лиц, лучшие на сегодняшний день, были получены при помощи глубоких свёрточных нейронных сетей. Существует множество подходов к обучению сетей на распознавание лиц – сети могут быть обучены как на классификацию, когда каждой персоне соответствует свой класс ([17, 19]), так и по принципу сиамских сетей на основе попарного сравнения для наилучшей верификации ([14, 19]). Также есть объединяющие оба варианта подходы ([15, 16]). Лучшие современные результаты на базе LFW получены композицией глубоких сетей, обученных на разных фрагментах (патчах) лиц ([3, 15, 16]). Тем не менее, использование ансамбля сетей требует существенно больших вычислительных ресурсов, в то время как некоторые одноэтажные сети могут быть достаточно эффективными [10, 13]. В качестве биометрического шаблона обычно используется выходной сигнал одного из слоёв, чаще всего последний скрытый слой [14, 15, 18–21]. Для снижения размерности полученных шаблонов обычно используется метод главных компонент [15, 17], а сравнение шаблонов осуществляется при помощи L2-расстояния [19, 23] или косинусного расстояния [15, 18, 20].

Бинарный хеш представляет собой битовый вектор, который используется в качестве шаблона при поиске объектов в базах. Классические методы хэширования описаны в [4]. Соответственно **бинарное хэширование** – это задача построения короткого бинарного хэша по исходному длинному вектору признаков. Основное отличие от классической задачи сни-

жения размерности заключается в том, что исходное пространство с евклидовой метрикой отображается в пространство хэшей с метрикой Хэмминга. В [2] представлена схема хэширования, основанная на наборе пороговых линейных хэш-функций с коэффициентами направляющего вектора, которые формируются в результате случайных независимых вращений. В [3] проблема сформулирована с точки зрения поиска подходящего вращения центрированных данных – как задача минимизации ошибки квантования относительно единичного гиперкуба – и для этой задачи предложен простой и эффективный алгоритм минимизации на основе итеративного квантования. Нелинейное обучение хэшированием с учителем использует нелинейное ядро для оптимизации целевой хэш-функции [7]. Алгоритм, сохраняющий сходство [5], использует кластеризацию методом k-средних для обучения бинарных индексов квантованных областей. Спектральное хэширование [11] использует идеи и методы нелинейного снижения размерности [1] для построения хэшей. Хэширование с сохранением топологии [12] реализует вложение Хэмминга с сохранением степени близости соседей. Локально-линейное хэширование [6] полагается на сохранение расстояния и одновременно реконструкцию локально-линейных структур. Семантическое хэширование [8] решает проблему, используя ограниченную машину Больцмана. Метод Boosted SSC, предложенный Шахнаровичем, Виолой и Дарделлом [9, 10], «выращивает» последовательно бит за битом хэш-код с перевешиванием примеров по аналогии с методом адаптивного бустинга и формирует взвешенное пространство Хэмминга.

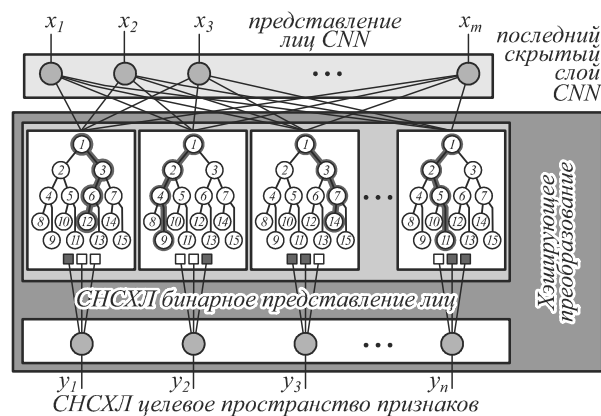


Рис. 1. Архитектура СНСХЛ: СНС + преобразование хэшированием, основанное на ХЛ

Идея построения бинарных шаблонов с помощью глубокого обучения реализована в [22]. СНС и хэширующий слой обучаются вместе, используя метод обратного распространения ошибки. В результате при использовании 32-битного биометрического шаблона достигается качество 91% в решении задачи верификации на базе LFW. Однако подобный подход неприемлем при использовании более сложных критериев кодирования. В частности, не получится оптимизировать кумулятивную кривую сравнений (ККС), харак-

теризующую качество решения задачи идентификации. Поэтому в данной работе предлагается использовать двухшаговую модель обучения: на первом шаге происходит обучение свёрточной сети, а затем преобразование хэшированием.

Предлагаемая методика хэширования основывается на использовании леса бинарных деревьев. Рассмотрим теперь работы, посвящённые задаче хэширования с использованием хэширующего леса. Куи, Шапиро и Бронштейн [27] предлагают схему хэширования случайным лесом, кодируя и сжимая информацию, для осуществления поиска в огромных массивах данных. Венс и Коста предлагают метод, основанный на случайном лесе для обучения регрессии и многоклассового разделения [28]. Ю и Юан [29] реализовали хэширование лесом со спектральным расстоянием Хэмминга, чувствительным к порядку. Хэширование лесом Спрингера и др. [30] объединяет kd-деревья с техникой хэширования. Форсированный алгоритм случайного леса, предложенный Мишиной и др. [31], не использует бинарное хэширование.

Предлагаемый подход заключается в кодировании пространства признаков, используя форсированный хэширующий лес в стиле Boosted SSC с оптимизацией специальной целевой функции. Таким образом, предлагаемый метод ФХЛ является обобщением метода Boosted SSC.

Архитектура предложенного подхода, начиная с обучения свёрточной сети и заканчивая получением хэшей, проиллюстрирована на рис. 2.

2. Построение биометрических шаблонов с помощью хэширующего леса

2.1. Boosted SSC, хэширующий лес и ФХЛ

Предлагаемый метод ФХЛ сочетает в себе алгоритмическую структуру Boosted SSC [9, 10] и структуру бинарного кодирования хэширующего леса [27–31].

Алгоритм Boosted SSC формирует пространство признаков с L1-метрикой как замену функции парного сходства, которая определена на множестве примеров позитивных (одноклассовых) и негативных (межклассовых) пар. Алгоритм SSC рассматривает пары, размеченные в соответствии со сходством, и формирует бинарное вложенное пространство. Вложение формируется независимым подбором порогов проекций исходных данных. Порог выбирается так, чтобы оптимально разделить проекции негативных пар и не разделить проекции позитивных пар. Boosted SSC жадно формирует вложение с адаптивным взвешиванием примеров в стиле адаптивного бустинга. Алгоритм *BoostPro* использует «смягчённые» пороговые функции для градиентного обучения на проекциях.

Отличие предложенного ФХЛ от Boosted SSC в следующем:

1) ФХЛ осуществляет бинарное кодирование и формирует тем самым пространство признаков, которое, вообще говоря, не бинарное, но может быть пространством бинарных векторов признаков с метрикой Хемминга, если необходимо;

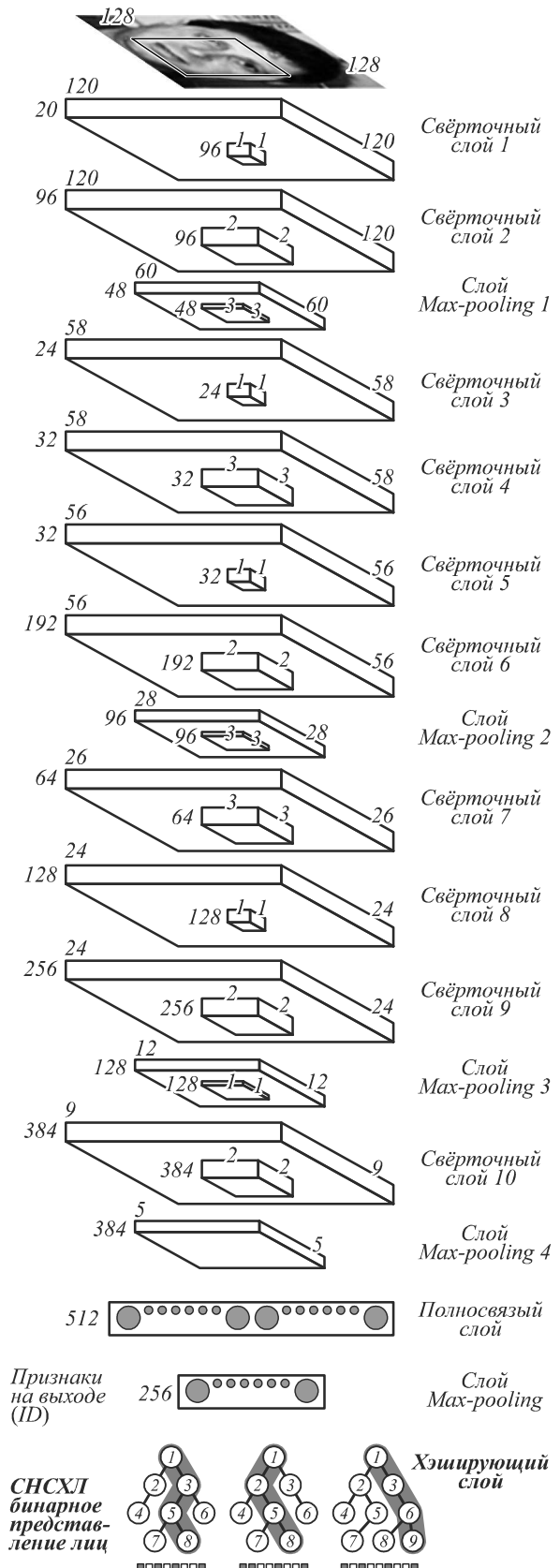


Рис. 2. Архитектура СНСХЛ, основанная на MFM-сети с мультисвёрточной структурой

2) ФХЛ реализует прямую оптимизацию любой заданной целевой функции;

- 3) ФХЛ «обучает» пороговые классификаторы с помощью алгоритма RANSAC без использования градиентной оптимизации;
- 4) ФХЛ реализует рекурсивное кодирование бинарными деревьями и формирует хэширующий лес, тогда как Boosted SSC реализует итеративное кодирование признаков и формирует хэш-вектор;
- 5) ФХЛ реализует адаптивное перевзвешивание пар обучающей выборки, опираясь на степень удовлетворения условиям целевой функции. В свою очередь, в Boosted SSC используется перевзвешивание, аналогичное алгоритму адаптивного бустинга;
- 6) Boosted SSC формирует взвешенное пространство Хэмминга. ФХЛ формирует пространство с любой заданной метрикой, включая невзвешенное Хэммингово.

Главные отличия предлагаемого метода ФХЛ по сравнению с другими методами хэширующего леса: для формирования хэширующего леса в данной работе предлагается использовать пороговые классификаторы, отбираемые с использованием техники RANSAC; используется специализированная, учитывающая специфику задач биометрии функция кодирования.

Также в данной работе предлагается ряд оригинальных приёмов, повышающих эффективность предлагаемого метода: биометрически-ориентированная целевая функция с одновременной оптимизацией идентификации и верификации лиц; выбор и обработка подвекторов входных векторов признаков; создание ансамбля независимых хэш-кодов для преодоления ограничений жадного обучения.

2.2. ФХЛ: целевое рекуррентное кодирование

Пусть обучающая выборка $X = \{\mathbf{x}_i \in R^m\}_{i=1, \dots, N}$ содержит N объектов, описываемых m -мерными векторами признаков. Представим X в виде n -мерного бинарного пространства:

$$X = \{\mathbf{x}_i \in R^m\}_{i=1, \dots, N} \rightarrow B = \{\mathbf{b}_i \in \{0, 1\}^n\}_{i=1, \dots, N}.$$

Такое представление – n -битный кодер:

$$\mathbf{h}(\mathbf{x}) : \mathbf{x} \in R^m \rightarrow \mathbf{b} \in \{0, 1\}^n. \quad (1)$$

Элементарным кодером назовём 1 -битовую хэш-функцию:

$$h(\mathbf{x}) : \mathbf{x} \in R^m \rightarrow b \in \{0, 1\}. \quad (2)$$

Пусть дана некоторая целевая функция (критерий кодирования), которую необходимо минимизировать:

$$J(X, \mathbf{h}) \rightarrow \min(\mathbf{h}). \quad (3)$$

Обозначим $\mathbf{h}^{(k)}(\mathbf{x}) = (h^{(1)}(\mathbf{x}), \dots, h^{(k)}(\mathbf{x}))$. Кроме того, операцию добавления новой однобитной хэш-функции в конец текущей сформированной последовательности хэш-функций (конкатенация) будем обозначать следующим образом: $\mathbf{h}^{(k)}(\mathbf{x}) := (\mathbf{h}^{(k-1)}(\mathbf{x}), h^{(k)}(\mathbf{x}))$. Жадный рекуррентный алгоритм целевого кодирования (Алгоритм 1) последовательно формирует биты кодера: $h^{(k)}(\mathbf{x}) = h^{(k)}(\mathbf{x}, \mathbf{h}^{(k-1)})$. Подробно процедура обучения каждого k -го бита описана в следующих подсекциях.

2.3. ФХЛ: обучение элементарной проекции с помощью алгоритма RANSAC

На k -м шаге «роста» кодера:

$$J(X, \mathbf{h}^{(k)}) = J(X, \mathbf{h}^{(k-1)}, h^{(k)}) \rightarrow \min\{h^{(k)} \in \mathbf{H}\}, \quad (4)$$

где \mathbf{H} – класс кодеров. Рассмотрим класс элементарных кодеров, в основе которых лежит выбор порога для линейных проекций

$$h(\mathbf{w}, t, \mathbf{x}) = \text{sgn}\left(\sum_{k=1, m} w_k x_k + t\right), \quad (5)$$

где \mathbf{w} – вектор весов, t – порог хэширующей функции, $\text{sgn}(u) = \{1, \text{если } u > 0; 0 - \text{иначе}\}$. В случае (5) функция (4) принимает вид

$$J(X, \mathbf{h}^{(k-1)}, h^{(k)}) = J(X, \mathbf{h}^{(k-1)}, \mathbf{w}, t) \rightarrow \min\{\mathbf{w} \in \mathbf{R}^m, t \in R\}. \quad (6)$$

Для поиска приближённого решения задачи (6) используется алгоритм RANSAC. Гипотезы о \mathbf{w} -параметрах генерируются, опираясь на случайный выбор пар элементов из различных классов обучающей выборки (Алгоритм 2). Выбор порога на 3-м шаге является одной из наиболее ресурсоёмких операций, это связано прежде всего с большим количеством пар.

2.4. ФХЛ: форсированный хэширующий лес

Описанная ранее процедура Learn1BitHash (Алгоритм 1) рекурсивно вызывает процедуру Learn1ProjectionHash (Алгоритм 2). Рассмотрим разбиение пространства X n -битным кодером:

$$\begin{aligned} X_B = \{X_b, \mathbf{b} \in \{0, 1\}^n\}, X_b = \{\mathbf{x} \in X : \mathbf{h}(\mathbf{x}) = \mathbf{b}\}, \\ X = \bigcup_{\mathbf{b} \in \{0, 1\}^n} X_b. \end{aligned} \quad (7)$$

Процесс рекурсивного кодирования – это дихотомическое разделение обучающей выборки путём нахождения оптимального элементарного кодера для каждой подвыборки каждого уровня разбиения. Таким образом, рекурсивный кодер для k -го бита

$$h^{(k)}(\mathbf{x}, \mathbf{h}^{(k-1)}) = h(\mathbf{w}(\mathbf{h}^{(k-1)}(\mathbf{x})), t(\mathbf{h}^{(k-1)}(\mathbf{x})), \mathbf{x}) \quad (8)$$

это комбинация $2^{(k-1)}$ бинарных линейных классификаторов:

$$\begin{aligned} h^{(k)}(\mathbf{x}, \mathbf{h}^{(k-1)}) = \text{Learn1BitHash}(J, X, \mathbf{h}^{(k-1)}) = \\ = \{\text{Learn1ProjectionHash}(J, X(\mathbf{h}^{(k-1)}), \mathbf{b}), \mathbf{h}^{(k-1)}\}, \\ \mathbf{b} \in \{0, 1\}^{(k-1)}. \end{aligned} \quad (9)$$

Такой рекурсивный n -битный кодер $\mathbf{h}(\mathbf{x})$ – это *де-рево* бинарных классификаторов, которое обладает гораздо большей обобщающей способностью по сравнению с n -битной последовательностью классификаторов.

Как известно, одно кодирующее дерево не может обеспечить достаточного качества решения задачи распознавания. Кроме того, количество проекций в дереве растёт экспоненциально с глубиной дерева. Таким образом, обучающая выборка фиксированного размера позволяет обучать деревья с ограниченной глубиной. Поэтому в данной работе предлагается формировать хэширующий лес, жадно подбирая хэширующие дере-

вья с целью оптимизации целевых функций для всех деревьев леса. Данный подход к формированию бинарного хеширующего леса (ФХЛ) описан в (Алгоритм 3). При этом используются обозначения: $n_{ORC} = p$ – глубина кодирующего дерева; $n_{ФХЛ} = n/p$ – количество деревьев;

$$\begin{aligned} \mathbf{h}^{[1,l]} &= (h^{(1)}(\mathbf{x}), \dots, h^{(p)}(\mathbf{x})), \\ \mathbf{h}^{[1,l-1]} &= (h^{(1)}(\mathbf{x}), \dots, h^{(p-p)}(\mathbf{x})), \\ \mathbf{h}^{[l,l]} &= (h^{(p-p+1)}(\mathbf{x}), \dots, h^{(p)}(\mathbf{x})). \end{aligned}$$

2.5. ФХЛ: хэширующий лес как метрическое пространство

Метрическое пространство (Y, d_Y) с $d_Y: Y \times Y \rightarrow R^+$ называется n -бит кодированным, если каждый $y \in Y$ соответствует уникальному $\mathbf{b} \in \{0, 1\}^n$ и даны две декодирующие функции: декодер признаков $f_y(\mathbf{b}): \{0, 1\}^n \rightarrow Y$ и декодер расстояния $f_d(\mathbf{b}_1, \mathbf{b}_2): \{0, 1\}^n \times \{0, 1\}^n \rightarrow R^+$, $f_d(\mathbf{b}_1, \mathbf{b}_2) = d_Y(f_y(\mathbf{b}_1), f_y(\mathbf{b}_2))$. Это позволяет определить целевую функцию кодирования, зависящую от расстояния, для кодера $\mathbf{h}(\mathbf{x})$ в виде:

$$\begin{aligned} \mathcal{J}(X, \mathbf{h}) &\rightarrow \min(\mathbf{h}) \Leftrightarrow \mathcal{J}(D_Y) \rightarrow \min(D_Y), \\ D_Y &= \{d_{ij} = f_d(\mathbf{h}(\mathbf{x}_i), \mathbf{h}(\mathbf{x}_j)), \mathbf{x}_i, \mathbf{x}_j \in X, \\ \mathbf{h}(\mathbf{x}) &\in \mathbf{H}\}_{i,j=1,N}, \end{aligned} \quad (10)$$

Алгоритм 1: Greedy ORC

Вход: X, \mathcal{J}, n_{ORC} .

Выход: $\mathbf{h}(\mathbf{x}): \mathbf{x} \in R^m \rightarrow \mathbf{y} \in \{0, 1\}^{n_{ORC}}, \mathbf{h}(\mathbf{x}) \in \mathbf{H}$.

Инициализация:

Шаг 0. $k := 0; \mathbf{h}^{(k)} := ()$.

Повторить:

$k := k + 1;$

Обучить k -й элементарный:

$$h^{(k)}(\mathbf{x}, \mathbf{h}^{(k-1)}) := \text{Learn1BitHash}(\mathcal{J}, X, \mathbf{h}^{(k-1)});$$

Добавить k -й элементарный кодер хэширующей функции:

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

пока $k < n_{ORC}$ // остановиться, если достигнут заданный размер кодера.

Алгоритм 2: RANSAC Learn1ProjectionHash

Вход: $\mathcal{J}, X, \mathbf{h}^{(k-1)}, k_{RANSAC}$.

Выход: $h(\mathbf{w}, t, \mathbf{x})$.

Инициализация:

Шаг 0. $k := 0; \mathcal{J}_{max} := -\infty$.

Повторить:

$k := k + 1;$

Шаг 1. Выбрать случайную пару элементов из разных классов $(\mathbf{x}_i, \mathbf{x}_j)$ в X .

Шаг 2. Получить $(\overline{\mathbf{x}_i}, \overline{\mathbf{x}_j})$ в качестве вектора-нормали для гиперплоскости: $\mathbf{w}_k := \mathbf{x}_j - \mathbf{x}_i$.

Шаг 3. Вычислить с использованием Алгоритма 3 порог t_k , минимизирующий \mathcal{J} (6) по t с $\mathbf{w} = \mathbf{w}_k$: $t_k := \text{argmin}_t \mathcal{J}(X, \mathbf{h}^{(k-1)}, \mathbf{w}_k, t)$.

Шаг 4. Если $\mathcal{J}(X, \mathbf{h}^{(k-1)}, \mathbf{w}_k, t_k) > \mathcal{J}_{max}$, тогда

$$\mathcal{J}_{max} := \mathcal{J}(X, \mathbf{h}^{(k-1)}, \mathbf{w}_k, t_k); \mathbf{w} := \mathbf{w}_k; t := t_k.$$

пока $k < k_{RANSAC}$. // остановиться, если достигнуто заданное число итераций RANSAC

Алгоритм 3: Boosted Hashing Forest

Вход: $X, \mathcal{J}, n_{ORC}, n_{ФХЛ}$.

Выход: $\mathbf{h}(\mathbf{x}): \mathbf{x} \in R^m \rightarrow \mathbf{y} \in \{0, 1\}^n$.

Инициализация:

$l := 0; \mathbf{h}^{[1,0]} := ()$.

Повторить:

$l := l + 1;$

Поставить цель в виде функции, зависящей от l -го кодирующего дерева:

$$g^{[l]}(X, \mathbf{h}^{[l,l]}) = \mathcal{J}(X, \mathbf{h}^{[1,l-1]}, \mathbf{h}^{[l,l]});$$

Обучить l -е кодирующее дерево:

$$\mathbf{h}^{[l,l]} := \text{GreedyORC}(g^{[l]}, X, n_{ORC});$$

Добавить l -е кодирующее дерево в хэширующий лес:

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

пока $l < n_{ORC}$. // остановиться, если достигнут заданный размер кодера.

Такая целевая функция зависит только от набора кодируемых расстояний d_{ij} . В данном конкретном случае сравниваются p -битные деревья, кодирующие расстояние в виде индексов. Это геодезические расстояния между кодами, как между соответствующими листьями кодирующих деревьев:

$$\begin{aligned} d_T(y_1, y_2) &= f_{dT}(\mathbf{b}_1, \mathbf{b}_2) = \\ &= 2 \sum_{k=1,p} \overline{(1 - \prod_{l=1,k} (1 - |b_1^{(l)} - b_2^{(l)}|))}. \end{aligned} \quad (11)$$

В итоге формируется расстояние для всего n -мерного леса, содержащего $q = n/p$ деревьев, как сумму расстояний между соответствующими p -битными деревьями:

$$d_{ij} = \sum_{l=1,q} f_{dT}(\mathbf{h}^{[l,l]}(\mathbf{x}_i), \mathbf{h}^{[l,l]}(\mathbf{x}_j)). \quad (12)$$

2.6. ФХЛ: целевая функция для верификации и идентификации

Пусть функция сходства s описывает позитивные (аутентичные) и негативные (ложные) пары:

$$s_{ij} = \begin{cases} 1, & \text{если класс}(\mathbf{x}_i) = \text{класс}(\mathbf{x}_j), \\ 0, & \text{иначе.} \end{cases} \quad (13)$$

«Идеальное» расстояние для k -битного бинарного кода выглядит следующим образом:

$$g_{ij}^{(k)} = \begin{cases} 0, & \text{если } s_{ij} = 1, \\ d_{\max}(k), & \text{иначе,} \end{cases} \quad (14)$$

где $d_{\max}(k)$ – максимально возможное расстояние. Тогда целевая функция может быть записана в виде

$$\begin{aligned} \mathcal{J}_{Dist}(D_Y) &= \sum_{i=1,N} \sum_{j=1,N} v_{ij} (d_{ij} - g_{ij})^2 \rightarrow \\ &\rightarrow \min(D_Y = \{d_{ij}\}_{i,j=1,N}), \end{aligned} \quad (15)$$

где v_{ij} – разные веса аутентичных и ложных пар. Эта целевая функция (15) отвечает за качество решения задачи верификации.

В ориентированных на идентификацию биометрических задачах требуется контролировать и расстоя-

ние, и упорядоченность расстояний. Пусть $d_k^1 = \max_l \{d_{kl} : s_{kl} = 1\}$. Это расстояние до самой удалённой аутентичной пары, а $d_k^0 = \min_l \{d_{kl} : s_{kl} = 0\}$ – расстояние до самой ближайшей ложной пары для запроса $\mathbf{h}(\mathbf{x}_k)$. Тогда ошибка упорядочивания e_{ij} для пары $(\mathbf{x}_i, \mathbf{x}_j)$ может быть выражена следующим образом:

$$e_{ij} = \begin{cases} 1, & \text{если } (s_{ij} = 0 \text{ и } h_{ij} < \max(d_i^1, d_j^1)) \\ & \text{или } (s_{ij} = 1 \text{ и } h_{ij} > \min(d_i^0, d_j^0)), \\ 0, & \text{иначе.} \end{cases} \quad (16)$$

Ошибка упорядочивания возникает, если ложная пара ближе, чем аутентичная, или если аутентичная дальше, чем ложная. В таком случае целевая функция примет вид:

$$J_{Ord}(D_Y) = \sum_{i=1, \dots, N} \sum_{j=1, \dots, N} v_{ij} (d_{ij} - g_{ij})^2 e_{ij} \rightarrow \min(D_Y = \{d_{ij}\}_{i,j=1, \dots, N}). \quad (17)$$

В этом случае штрафуются разница между d_{ij} и целевым расстоянием g_{ij} , как в (15), но только в том случае, если существует ошибка упорядочивания (16) для этой пары. Таким образом, критерий (17) напрямую отражает параметры идентификации лиц (ККС).

Для одновременного контроля верификации и идентификации критерии (15) и (17) совмещаются:

$$J(D_Y) = \alpha J_{Dist}(D_Y) + (1 - \alpha) J_{Ord}(D_Y) = \sum_{i=1, \dots, N} \sum_{j=1, \dots, N} v_{ij} (d_{ij} - g_{ij})^2 (e_{ij} + \alpha(1 - e_{ij})) \rightarrow \min(D_Y = \{d_{ij}\}_{i,j=1, \dots, N}), \quad (18)$$

где $\alpha \in [0, 1]$ – настраиваемый параметр, задающий приоритет одной или другой задаче.

2.7. ФХЛ-реализация

Для улучшения результата используется ряд дополнительных полуэвристических модификаций описанной ранее схемы. Вместо (14) используется следующее выражение:

$$g_{ij}^{(k)} = \begin{cases} 0, & \text{если } s_{ij} = 1, \\ m^{(k-1)}_1 + 3\sigma^{(k-1)}_1, & \text{иначе,} \end{cases} \quad (19)$$

где $m^{(k-1)}_1$ и $\sigma^{(k-1)}_1$ – математическое ожидание и среднеквадратичное отклонение закодированных расстояний между аутентичными парами. Подобное целевое расстояние (19) исключает штраф для ложных пар, которые не могут быть приняты за аутентичные. В выражении (18) используется адаптивное взвешивание пар на каждом k -м шаге:

$$v_{ij}^{(k)} = \begin{cases} \gamma/a^{(k)}, & \text{если } s_{ij} = 1, \\ 1/b^{(k)}, & \text{иначе,} \end{cases} \quad (20)$$

$a^{(k)} = \sum_{i=1, \dots, N} \sum_{j=1, \dots, N} s_{ij} (d_{ij} - g_{ij})^2 (e_{ij} + \alpha(1 - e_{ij}))$,
 $b^{(k)} = \sum_{i=1, \dots, N} \sum_{j=1, \dots, N} (1 - s_{ij}) (d_{ij} - g_{ij})^2 (e_{ij} + \alpha(1 - e_{ij}))$,
 где $a^{(k)}$ и $b^{(k)}$ определяют выравнивание весов между всеми аутентичными и ложными парами, и настраиваемый параметр $\gamma > 1$ даёт небольшой перевес в пользу аутентичных пар.

Входной m -мерный вектор признаков на набор независимых кодируемых подвекторов с фиксированными размерами из множества $\mathbf{m} = \{m_{\min}, \dots, m_{\max}\}$. В процессе работы алгоритма данные подвектора кодируются независимо при помощи Алгоритма 2, а в Алгоритме 1 используется лучший из кодеров по всем использованным подвекторам. Выходной бинарный вектор размера n состоит из нескольких независимо «выращиваемых» частей размера $n_{\text{ФХЛ}} < n$. Эта стратегия обучения предотвращает преждевременное насыщение целевой функции.

Таким образом, реализация бинарного хэширования включает следующие параметры: \mathbf{m} , n_{ORC} , $n_{\text{ФХЛ}}$, k_{RANSAC} , α и γ . Тип кодируемой метрики также является свободным параметром.

3. Эксперименты

В данном параграфе описывается методология обучения и тестирования СНСХЛ, результаты в задаче вложения пространства Хэмминга, сравнивается предложенный метод ФХЛ с оригинальным Boosted SSC, исследуется зависимость показателей СНСХЛ от глубины кодирующих деревьев и сравниваются результаты СНСХС и СНСХЛ с результатами лучших методов на базе изображений лиц LFW. В данной работе точность верификации по базе LFW измеряется по протоколу *unrestricted*. Для оценки качества идентификации (ККС и rank-1) используется протокол тестирования, описанный в [32].

3.1. Методология: обучение и тестирование СНСХЛ

Базовая свёрточная сеть была обучена на базе лиц CASIA-WebFace. Изображения лиц нормализованы путём поворота особых точек, соответствующих центрам глаз, до горизонтального положения, далее масштабированы до фиксированного расстояния между этими точками, далее изображения были обрезаны до размера 128×128 . Для глубокого обучения исходной сети на многоклассовую идентификацию использовалась среда обучения Caffe (<http://caffe.berkeleyvision.org/>) подобно [17, 20]. Хэширующий лес обучался на базе изображений лиц, состоящей из 1000 аутентичных пар и, соответственно, 999000 ложных пар базы изображений «Faces in the Wild» (не содержащихся в тестовой выборке).

С использованием предлагаемого подхода были получены две группы биометрических шаблонов:

1. Компактные бинарные шаблоны с максимальной скоростью сравнения, используя метрику Хемминга – кодирующие деревья глубины 1.
2. Биометрические шаблоны, обеспечивающие максимальное качество решения задачи идентификации – с использованием кодирующих деревьев глубины 2–7.

В работе использовались следующие параметры алгоритма: $\mathbf{m} = \{8, 16, 32\}$, $k_{\text{RANSAC}} = 100$, $\alpha = 0,25$, $\gamma = 1,1$. Но установили $n_{\text{ФХЛ}} = 200$ для СНСХЛ-200×1, $n_{\text{ФХЛ}} = 500$ для СНСХЛ-2000×1 и $n_{\text{ФХЛ}} = 100$ для СНСХЛ-2000×7.

Эти значения определены экспериментально, путём анализа скорости роста показателей идентификации в зависимости от количества закодированных бит в процессе хэширования. Для нормализации изображений базы LFW использовалась техника, предложенная в [33].

При сравнении с базовой сетью биометрические шаблоны для базовой СНС формируются подобно [12] – как активационный вектор размерности 256 для последнего скрытого слоя сети. В качестве метрик сравнения биометрических шаблонов, сгенерированных базовой сетью, использовалось косинусное сходство (СНС+КС) и L2-расстояние (СНС+L2).

3.2. Вложение Хэмминга – деревья глубины 1

В данном подпараграфе описываются результаты тестирования бинарных биометрических шаблонов, сравниваемых метрикой Хемминга, построенных с использованием предлагаемого подхода. При такой постановке задачи СНСХЛ переходит обратно в СНСХС.

При размере биометрического шаблона в 2000 бит СНСХЛ-2000×1 достигает точности верификации в 98,14% на LFW, которая превосходит базовую сеть в качестве решения как задачи верификации (рис. 3а), так и задачи идентификации (рис. 3б). Более того, при размере биометрического шаблона в 25 байт СНСХЛ-200×1 достигает качества верификации 0,96.

По сравнению с базовым методом Boosted SSC, предлагаемый метод позволяет получить существенный выигрыш при решении задач идентификации (рис. 3в) и верификации (рис. 3а).

При этом предлагаемый подход превосходит базовый BoostSSC практически для всех размеров биометрических шаблонов (рис. 3в).

3.3. СНСХЛ: зависимость результатов от глубины деревьев

В предыдущем параграфе были рассмотрены результаты, полученные при использовании деревьев глубины 1. В данном подпараграфе рассмотрена зависимость качества решения задач верификации и идентификации от глубины используемых деревьев.

СНСХЛ с 2000 7-битными деревьями (СНСХЛ-2000×7) достигает качества верификации в 98,59%. Результат в идентификации СНСХЛ-2000×7 – 93% rank-1 на LFW (по сравнению с 89,9% rank-1 у базовой сети). Рис. 4в содержит графики ROC-для СНСХЛ с разными глубинами кодирующих деревьев. Лес с 7-битными деревьями лучший на ROC графиках, но 6-битные и 5-битные решения дают очень близкие результаты. Мы предполагаем, что причина этого в недостаточном размере обучающей выборки для хэширующего леса. Рис. 4а, б показывают, что СНСХЛ-2000×7 опережает как базовую сеть, так и СНСХЛ-2000×1 в верификации (ROC) и идентификации (СМС).

В целом, можно сделать вывод, что добавление хэширующего слоя на верхний уровень СНС позволяет сгенерировать более компактные бинарные биометрические шаблоны и улучшить показатели верификации и особенно идентификации.

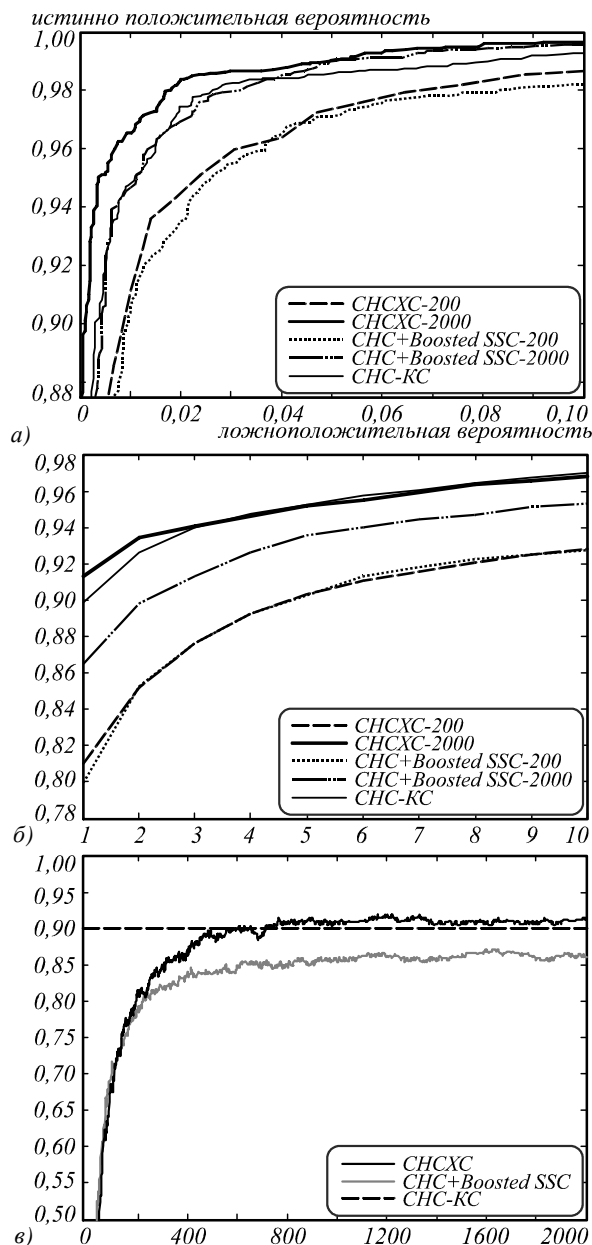


Рис. 3. Графики ROC (а), СМС (б), показатели идентификации (rank 1) (с) на LFW в зависимости от размера биометрического шаблона в битах для предложенных ФХЛ (СНС+ХЛ), и оригинального Boosted SSC (СНС+Boosted SSC), и лучшего исходного СНС решения без хэширования: СНС + последний скрытый слой + косинусное сходство (СНС +КС)

3.4. Сравнение СНСХС и СНСХЛ

с результатами лучших методов по базе LFW

Сравним предлагаемое СНСХЛ-решение с признанными лучшими современными методами (лучшими на базе LFW) в точности верификации (табл. 2). СНСХЛ-2000×1 опережает DeepFace-ансамбль [11], DeepID [18], WebFace [21] и MFM Net [12]. DeepID2 [17], DeepID3 [25] и Baidu [7], многопатчевые СНС превосходят СНСХЛ-2000×1, базирующийся на однопатчевой сети.

Следует заметить, что СНСХЛ-200×1 (25 байт) хэш показывает результат 96,3% на LFW.

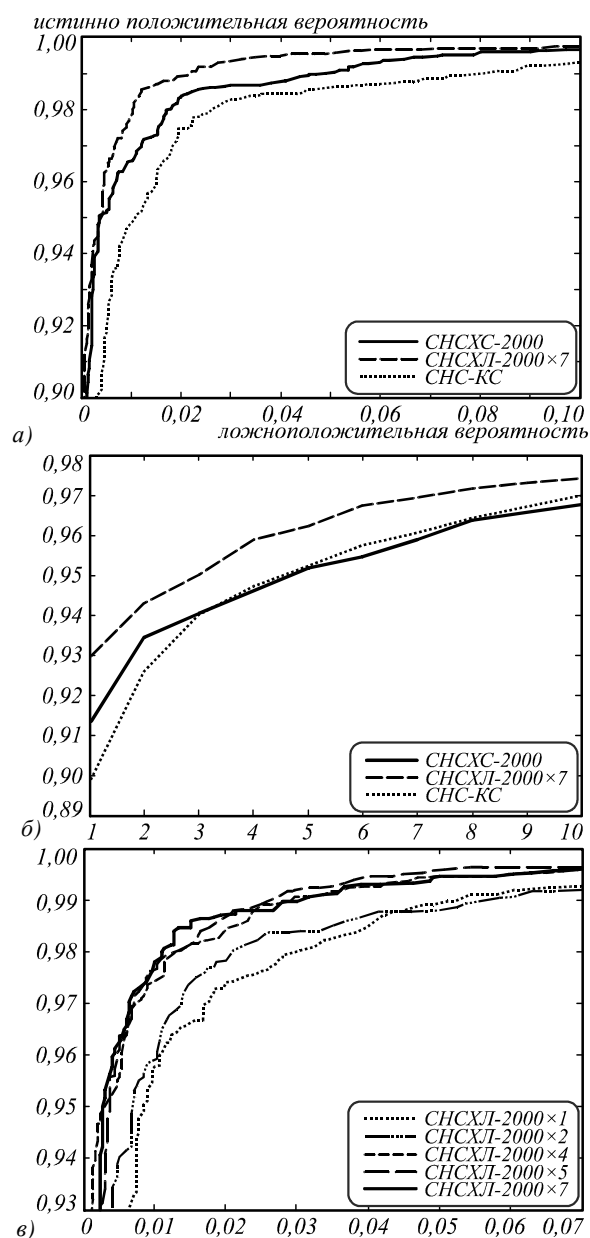


Рис. 4. Графики ROC (а) и CMC (б) для CHC+KC, CHCXL-2000×1 и CHCXL-2000×7; ROC графики (в) для CHCXL-1000х-битных деревьев

Сравним его с предыдущим лучшим результатом CHCXC [22]. С одной стороны, экстремально короткое 32-битное бинарное представление лиц [22] позволяет достичь 91% верификации на LFW. Обученный предлагаемым методом CHCXC-32×1 достигает лишь 90%.

С другой стороны, как отмечают авторы [22], предлагаемому ими методу требуются более 1000 бит для достижения 96% верификации на базе LFW. Таким образом, предлагаемый CHCXL-200×1 сжимает информацию о лицах в 5 раз эффективнее.

Результат идентификации (rank-1), работающего в реальном времени, CHCXL-2000×7 – 0,93 на LFW. По данному показателю предлагаемый подход достаточно сильно приблизился к лучшим известным показателям идентификации, полученным на существенно

более глубокой и медленной многослойной сети DeepID3 CHC [16] (0,96 rank-1 на LFW). Baidu [26] декларирует и более высокие результаты (0,98 rank-1 на LFW), но использует обучающую выборку размером в 1,2 миллиона изображений против 400 тысяч в данной работе.

Табл. 1. Вероятность правильной верификации по базе LFW, размер кода и скорость сравнения CHC и CHCXC

Метод	Вероятность верификации	Размер шаблона	Сравнений в секунду
CHC+L2	0,947	8192 bit	2713222
CHC+ФХЛ-200×1	0,963	200 bit	194986071
CHC+KC	0,975	8192 bit	2787632
CHC+ФХЛ-2000×1	0,9814	2000 bit	27855153

Табл. 2. Верификация на LFW

Метод	Вероятность верификации
WebFace [17]	0,9613
CHCXC-200×1	0,963 ± 0,00494
DeepFace-ensemble [10]	0,9730 ± 0,0025
DeepID [34]	0,9745 ± 0,0026
MFM Net [16]	0,9777
CHCXC-2000×1	0,9814
CHCXL-2000×7	0,9859
DeepID2 [27]	0,9915 ± 0,0013
DeepID3 [8]	0,9953 ± 0,0010
Baidu [33]	0,9977 ± 0,0006

Заключение

В данной работе предлагается метод построения биометрических шаблонов на основе CHC для задачи идентификации лиц в реальном времени. Метод позволяет строить компактные биометрические шаблоны (250 байт) с одновременно возросшими (по сравнению с базовой сетью) показателями верификации (98,14%) и идентификации (91% rank-1) на базе лиц LFW, а также ультракомпактные (25 байт) шаблоны (в 40 раз меньше шаблона, построенного с использованием базовой сети, и в 70 раз быстрее скорость сравнения при потере точности верификации в 1% по сравнению с базовой сетью). С другой стороны, описанный подход позволяет строить большие биометрические шаблоны, позволяющие достигать высокого качества распознавания – CHCXL с 2000 кодирующими 7-битными деревьями достигает 98,59% точности верификации и 93% rank-1 на LFW (прибавка более чем в 3% в rank-1 относительно исходной CHC).

Также в данной работе предлагается мультисвёрточная глубокая сеть на основе архитектуры сети, описанной в [20]. Использование мультисвёрточной архитектуры позволяет создавать шаблоны лиц с частотой 40 изображений в секунду на CPU Core i7 и свыше 120 кадров в секунду на видеоадаптере GeForce GTX 650.

В дальнейших работах планируется достичь повышения результатов распознавания лиц путём использования базовой сети более сложной топологии. Предложенный метод в дальнейшем будет приме-

няться для задач кодирования данных и снижения размерности (обучение с учителем, частичное обучение с учителем и обучение без учителя). Также предполагается исследовать влияние свойств кодируемой метрики на процесс обучения хэширующего леса.

Благодарности

Работа была поддержана грантом РФФ (Проект № 16-11-00082).

Литература

1. **Belkin, M.** Laplacian eigenmaps and spectral techniques for embedding and clustering / M. Belkin, P. Niyogi // Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. – 2001. – P. 585-591.
2. **Gionis, A.** Similarity search in high dimensions via hashing / A. Gionis, P. Indyk, R. Motwani // Proceedings of the 25th International Conference on Very Large Data Bases. – 1999. – P. 518-529.
3. **Gong, Y.** Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval / Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2012. – Vol. 35, Issue 12. – P. 2916-2929. – DOI: 10.1109/TPAMI.2012.193.
4. **Grauman, K.** Learning binary hash codes for large-scale image search / K. Grauman, R. Fergus. – In Book: Machine Learning for Computer Vision / ed. by R. Cipolla, S. Battiato, G.M. Farinella. – Berlin, Heidelberg: Springer, 2013. – P. 49-87. – ISBN: 978-3-642-28660-5. – DOI: 10.1007/978-3-642-28661-2_3.
5. **He, K.** K-means Hashing: An affinity-preserving quantization method for learning binary compact codes / K. He, F. Wen, J. Sun // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2013. – P. 2938-2945. – DOI: 10.1109/CVPR.2013.378.
6. **Irie, G.** Locally linear hashing for extracting non-linear manifolds / G. Irie, L. Zhenguo, W. Xiao-Ming, C. Shih-Fu // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2014. – P. 2115-2122. – DOI: 10.1109/CVPR.2014.272.
7. **Liu, W.** Supervised hashing with kernels / W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2012. – P. 2074-2081. – DOI: 10.1109/CVPR.2012.6247912.
8. **Salakhutdinov, R.** Semantic hashing / R. Salakhutdinov, G. Hinton // International Journal of Approximate Reasoning. – 2009. – Vol. 50, Issue 7. – P. 969-978. – DOI: 10.1016/j.ijar.2008.11.006.
9. **Shakhnarovich, G.** Learning task-specific similarity: PhD thesis / G. Shakhnarovich. – Cambridge, MA: Massachusetts Institute of Technology, 2005.
10. **Shakhnarovich, G.** Fast pose estimation with parameter sensitive hashing / G. Shakhnarovich, P. Viola, T. Darrell // Proceedings of the Ninth IEEE International Conference on Computer Vision. – 2003. – Vol. 2. – P. 750-757. – DOI: 10.1109/ICCV.2003.1238424.
11. **Weiss, Y.** Spectral Hashing / Y. Weiss, A. Torralba, R. Fergus. – In Book: Advances in Neural Information Processing Systems 21 – Proceedings of the 2008 Conference. – 2008. – P. 1753-1760.
12. **Zhang, L.** Topology preserving hashing for similarity search / L. Zhang, Y. Zhang, X. Gu, J. Tang, Q. Tian // Proceedings of the 21st ACM international conference on Multimedia. – 2013. – P. 123-132. – DOI: 10.1145/2502081.2502091.
13. **Cao, Z.** Face Recognition with Learning-based Descriptor / Z. Cao, Q. Yin, X. Tang, J. Sun // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2010. – P. 2707-2714. – DOI: 10.1109/CVPR.2010.5539992.
14. **Fan, H.** Learning deep face representation / H. Fan, Z. Cao, Y. Jiang, Q. Yin, C. Doudou // arXiv preprint arXiv:1403.2802. – 2014.
15. **Sun, Y.** Deep learning face representation by joint identification-verification / Y. Sun, X. Wang, X. Tang // Proceedings of the 27th International Conference on Neural Information Processing Systems. – 2014. – P. 1988-1996.
16. **Sun, Y.** DeepID3: Face recognition with very deep neural networks / Y. Sun, X. Wang, X. Tang // arXiv preprint arXiv:1502.00873. – 2015.
17. **Sun, Y.** Deep learning face representation from predicting 10,000 classes / Y. Sun, X. Wang, X. Tang // Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 1891-1898. – DOI: 10.1109/CVPR.2014.244.
18. **Taigman, Y.** DeepFace: closing the gap to human-level performance in face verification / Y. Taigman, M. Yang, M. Ranzato, L. Wolf // Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – P. 1701-1708. – DOI: 10.1109/CVPR.2014.220.
19. **Wang, W.** Face recognition based on deep learning / W. Wang, J. Yang, J. Xiao, S. Li, D. Zhou. – In Book: Human Centered Computing / ed. by Q. Zu, B. Hu, N. Gu, S. Seng. – Vol. 8944. – Springer, 2015. – P. 812-820. – ISBN: 978-3-319-15553-1. – DOI: 10.1007/978-3-319-15554-8_73.
20. **Wu, X.** Learning robust deep face representation / X. Wu // arXiv preprint arXiv:1507.04844. – 2015.
21. **Zhou, E.** Naive-deep face recognition: Touching the limit of LFW benchmark or not? / E. Zhou, Z. Cao, Q. Yin // arXiv preprint arXiv:1501.04690. – 2015.
22. **Fan, H.** Learning compact face representation: Packing a face into an int32 / H. Fan, M. Yang, Z. Cao, Y. Jiang, Q. Yin // Proceedings of the 22nd ACM international conference on Multimedia. – 2014. – P. 933-936. – DOI: 10.1145/2647868.2654960.
23. **Chen, D.** Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification / D. Chen, X. Cao, F. Wen, J. Sun // Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 2013. – P. 3025-3032. – DOI: 10.1109/CVPR.2013.389.
24. **Nguyen, H.-V.** Cosine similarity metric learning for face verification / H.-V. Nguyen, L. Bai // Proceedings of 10th Asian Conference on Computer Vision. – 2010. – P. 709-720. – DOI: 10.1007/978-3-642-19309-5_55.
25. **Taigman, Y.** Multiple one-shots for utilizing class label information / Y. Taigman, L. Wolf, T. Hassner // Proceedings of British Machine Vision Conference. – 2009. – DOI: 10.5244/C.23.77.
26. **Liu, J.** Targeting ultimate accuracy: face recognition via deep embedding / J. Liu, Y. Deng, T. Bai, Z. Wei, C. Huang // arXiv preprint arXiv:1506.07310. – 2015.
27. **Qiu, Q.** Random forests can hash / Q. Qiu, G. Sapiro, A. Bronstein // arXiv preprint arXiv:1412.5083. – 2014.
28. **Vens, C.** Random forest based feature induction / C. Vens, F. Costa // Proceedings of IEEE 11th International Conference on Data Mining (ICDM). – 2011. – P. 744-753. – DOI: 10.1109/ICDM.2011.121.

29. **Yu, G.** Scalable forest hashing for fast similarity search / G. Yu, J. Yuan // Proceedings of IEEE International Conference on Multimedia and Expo (ICME). – 2014. – P. 1-6. – DOI: 10.1109/ICME.2014.6890219.
30. **Springer, J.** Forest hashing: Expediting large scale image retrieval / J. Springer, X. Xin, Z. Li, J. Watt, A. Katsaggelos // Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – 2013. – P. 1681-1684. – DOI: 10.1109/ICASSP.2013.6637938.
31. **Mishina, Y.** Boosted random forest / Y. Mishina, R. Murata, M. Tsuchiya, H. Fujiyoshi // IEICE Transactions on Information and Systems. – 2015. – Vol. E98-D, Issue 9. – P. 1630-1636. – DOI: 10.1587/transinf.2014OPP0004.
32. **Best-Rowden, L.** Unconstrained face recognition: Identifying a person of interest from a media collection / L. Best-Rowden, H. Han, C. Otto, B. Klare, A.K. Jain // IEEE Transactions on Information Forensics and Security. – 2014. – Vol. 9, Issue 12. – P. 2144-2157. – DOI: 10.1109/TIFS.2014.2359577.
33. **Huang, G.-B.** Learning to align from scratch / G.-B. Huang, M. Mattar, H. Lee, E. Learned-Miller // Proceedings of the 25th International Conference on Neural Information Processing Systems. – 2012. – P. 764-772.
34. **Schroff, F.** FaceNet: A unified embedding for face recognition and clustering / F. Schroff, D. Kalenichenko, J. Philbin // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 815-823. – DOI: 10.1109/CVPR.2015.7298682.

Сведения об авторах

Сведения об авторе **Визильтер Юрий Валентинович** см. стр. 235 этого выпуска.

Горбачевич Владимир Сергеевич, 1985 года рождения, в 2009 году окончил ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)». В настоящее время является начальником лаборатории ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Опубликовал более 20 научных работ (из них более 6 статей в рецензируемых журналах из списка ВАК). Область научных интересов: обработка и анализ изображений, цифровая фотограмметрия, машинное зрение, математическая морфология, распознавание образов, машинное обучение, биометрия. E-mail: gvs@gosniias.ru.

Воротников Андрей Валерьевич, 1991 года рождения, в 2013 году окончил МГУ имени М.В. Ломоносова. В настоящее время является сотрудником ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Область научных интересов: обработка и анализ изображений, машинное зрение, математическая морфология, распознавание образов, машинное обучение, биометрия. E-mail: vorotnikov@gosniias.ru.

Костромов Никита Алексеевич, 1986 года рождения, в 2009 году окончил ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)». В настоящее время является начальником сектора ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Является автором и соавтором более 10 научных трудов (из них 8 статей в рецензируемых журналах из списка ВАК). Круг научных интересов: обработка и анализ изображений, машинное обучение, глубокое обучение, свёрточные сети, распознавание образов, биометрия. E-mail: nikita-kostromov@yandex.ru.

ГРТИ: 28.23.29

Поступила в редакцию 23 ноября 2016. Окончательный вариант – 16 марта 2017.

REAL-TIME FACE IDENTIFICATION VIA CNN AND BOOSTED HASHING FOREST

Y.V. Vizilter¹, V.S. Gorbachevich¹, A.V. Vorotnikov¹, N.A. Kostromov¹
¹State Research Institute of Aviation Systems (GosNIAS), Moscow, Russia

Abstract

This paper presents a new approach to constructing a biometric template using a Convolutional Neural Network (CNN) with Hashing Forest. The approach consists of several steps: training a convolutional neural network, transforming it to a multiple convolution architecture, and finally learning the output hashing transform via a new Boosted Hashing Forest technique. This technique generalizes the Boosted SSC (Similarity Sensitive Coding) approach for hashing learning with joint optimization of face verification and identification. The proposed network via hashing forest is trained on the CASIA-WebFace dataset and evaluated on the LFW dataset. The result of coding the output of a single CNN is 97% on LFW. For Hamming embedding, the proposed approach enables a 200 bit (25 byte) code to be constructed with a 96.3% verification accuracy and a 2000-bit code with a 98.14% verification accuracy on LFW. The convolutional network with hashing forest with 2000×7-bit hashing trees achieves 93% rank-1 on LFW relative to the basic convolutional network's 89.9% rank-1. The proposed approach generates templates at the rate of 40+ fps with a GPU Core i7 and 120+ fps with a GPU GeForce GTX 650.

Keywords: convolutional neural networks, hashing, binary trees, Hamming distance, biometrics.

Citation: Vizilter YuV, Gorbachevich VS, Vorotnikov AV, Kostromov NA. Real-time face identification via CNN and boosted hashing forest. *Computer Optics* 2017; 41(2): 254-265. DOI: 10.18287/2412-6179-2017-41-2-254-265.

Acknowledgements: The work was supported by a grant from the Russian Science Foundation (Project No. 16-11-00082).

References

- [1] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Proc NIPS* 2001; 14: 585-591.
- [2] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing. *Proc VLDB* 1999: 518-529.
- [3] Gong Y, Lazebnik S, Gordo A, Perronnin F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans Pattern Anal Mach Intell* 2012; 35(12): 2916-2929. DOI: 10.1109/TPAMI.2012.193.
- [4] Grauman K, Fergus R. Learning binary hash codes for large-scale image search. In: Cipolla R, Battiato S, Fari-nella GM, eds. *Machine Learning for Computer Vision*. Berlin, Heidelberg: Springer; 2013: 49-87. ISBN: 978-3-642-28660-5. DOI: 10.1007/978-3-642-28661-2_3.
- [5] He K, Wen F, Sun J. K-means Hashing: An affinity-preserving quantization method for learning binary compact codes. *Proc CVPR* 2013: 2938-2945. DOI: 10.1109/CVPR.2013.378.
- [6] Irie G, Zhenguo L, Xiao-Ming W, Shih-Fu C. Locally linear hashing for extracting non-linear manifolds. *Proc CVPR* 2014: 2115-2122. DOI: 10.1109/CVPR.2014.272.
- [7] Liu W, Wang J, Ji R, Jiang Y-G, Chang S-F. Supervised hashing with kernels. *Proc CVPR* 2012: 2074-2081. DOI: 10.1109/CVPR.2012.6247912.
- [8] Salakhutdinov R, Hinton G. Semantic hashing. *International Journal of Approximate Reasoning* 2009; 50(7): 969-978. DOI: 10.1016/j.ijar.2008.11.006.
- [9] Shakhnarovich G. Learning task-specific similarity. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology; 2005.
- [10] Shakhnarovich G, Viola P, Darrell T. Fast pose estimation with parameter sensitive hashing. *Proc ICCV '03* 2003; 2: 750-757. DOI: 10.1109/ICCV.2003.1238424.
- [11] Weiss Y, Torralba A, Fergus R. Spectral Hashing. In *Book: Advances in Neural Information Processing Systems 21 – Proceedings of the 2008 Conference* 2008: 1753-1760.
- [12] Zhang L, Zhang Y, Gu X, Tang J, Tian Q. Topology preserving hashing for similarity search. *Proc ACM Int Conf Multimedia* 2013: 123-132. DOI: 10.1145/2502081.2502091.
- [13] Cao Z, Yin Q, Tang X, Sun J. Face Recognition with Learning-based Descriptor. *Proc CVPR* 2010: 2707-2714. DOI: 10.1109/CVPR.2010.5539992.
- [14] Fan H, Cao Z, Jiang Y, Yin Q, Doudou C. Learning deep face representation. *arXiv preprint arXiv:1403.2802* 2014.
- [15] Sun Y, Wang X, Tang X. Deep learning face representation by joint identification-verification. *Proc NIPS* 2014: 1988-1996.
- [16] Sun Y, Wang X, Tang X. DeepID3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873* 2015.
- [17] Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10,000 classes. *Proc CVPR* 2014: 1891-1898. DOI: 10.1109/CVPR.2014.244.
- [18] Taigman Y, Yang M, Ranzato M, Wolf L. DeepFace: closing the gap to human-level performance in face verification. *Proc CVPR* 2014: 1701-1708. DOI: 10.1109/CVPR.2014.220.
- [19] Wang W, Yang J, Xiao J, Li S, Zhou D. Face recognition based on deep learning. In *Book: Zu Q, Hu B, Gu N, Seng S, eds. Human Centered Computing*. Springer; 2015: 812-820. ISBN: 978-3-319-15553-1. DOI: 10.1007/978-3-319-15554-8_73.
- [20] Wu X. Learning robust deep face representation. *arXiv preprint arXiv:1507.04844* 2015.
- [21] Zhou E, Cao Z, Yin Q. Naive-deep face recognition: Touching the limit of LFW benchmark or not? *arXiv preprint arXiv:1501.04690* 2015.
- [22] Fan H, Yang M, Cao Z, Jiang Y, Yin Q. Learning Compact Face Representation: Packing a Face into an int32. *Proc ACM Int Conf Multimedia* 2014: 933-936. DOI: 10.1145/2647868.2654960.
- [23] Chen D, Cao X, Wen F, Sun J. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. *Proc CVPR* 2013: 3025-3032. DOI: 10.1109/CVPR.2013.389.
- [24] Nguyen H-V, Bai L. Cosine similarity metric learning for face verification. *Proc ACCV* 2010: 709-720. DOI: 10.1007/978-3-642-19309-5_55.
- [25] Taigman Y, Wolf L, Hassner T. Multiple one-shots for utilizing class label information. *Proc BMVC* 2009. DOI: 10.5244/C.23.77.
- [26] Liu J, Deng Y, Bai T, Wei Z, Huang C. Targeting ultimate accuracy: face recognition via deep embedding. *arXiv preprint arXiv:1506.07310* 2015.
- [27] Qiu Q, Sapiro G, Bronstein A. Random forests can hash. *arXiv preprint arXiv:1412.5083* 2014.
- [28] Vens C, Costa F. Random Forest Based Feature Induction. *Proc ICDM* 2011: 744-753. DOI: 10.1109/ICDM.2011.121.
- [29] Yu G, Yuan J. Scalable forest hashing for fast similarity search. *Proc ICME* 2014: 1-6. DOI: 10.1109/ICME.2014.6890219.
- [30] Springer J, Xin X, Li Z, Watt J, Katsaggelos A. Forest hashing: Expediting large scale image retrieval. *Proc ICASSP* 2013: 1681-1684. DOI: 10.1109/ICASSP.2013.6637938.
- [31] Mishina Y, Murata R, Tsuchiya M, Fujiyoshi H. Boosted Random Forest. *IEICE Transactions on Information and Systems* 2015; E98-D(9): 1630-1636. DOI: 10.1587/transinf.2014OPP0004.
- [32] Best-Rowden L, Han H, Otto C, Klare B, Jain AK. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Trans Inf Forens Security* 2014; 9(12): 2144-2157. DOI: 10.1109/TIFS.2014.2359577.
- [33] Huang G-B, Mattar M, Lee H, Learned-Miller E. Learning to align from scratch. *Proc NIPS '12* 2012: 764-772.
- [34] Schroff, F, Kalenichenko D. and Philbin J. FaceNet: A unified embedding for face recognition and clustering. *Proc CVPR* 2015: 815-823. DOI: 10.1109/CVPR.2015.7298682.

Authors' information

The information about author **Yuri Valentinovich Vizilter** you can find on page 236 of this issue.

Vladimir Sergeevich Gorbachevich (b.1985) graduated from Moscow Aviation Institute (National Research University) in 2009. Currently he works as the head of laboratory at the FGUP "GosNIIAS". Author of 20 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: gvs@gosniias.ru.

Andrey Valerievich Vorotnikov (b.1991) graduated from Lomonosov Moscow State University in 2013. Currently he works as the engineer at the FGUP "GosNIIAS". Research interests are processing and image analysis, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: vorotnikov@gosniias.ru.

Nikita Alexeevich Kostromov (b.1986) graduated from Moscow Aviation Institute (National Research University) in 2009. He currently resides at the FGUP "GosNIIAS". Author of more than 10 scientific papers. Research interests are processing image and video analysis, pattern recognition, machine learning, deep learning, convolution network, biometrics. E-mail: nikita-kostromov@yandex.ru.

Received November 23, 2017. The final version – March 16, 2017.
