# Упорядочивание данных в системах видеонаблюдения на основе технологий глубокого обучения

**А.Д. Соколова[1], А.В. Савченко[1]**

[1]Национальный исследовательский университет Высшая школа экономики, Большая Печерская 25/12, Нижний Новгород, Россия, 603155

**Аннотация.** Рассматривается задача организации информации в системах видеонаблюдения с помощью автоматического выделения групп треков, так, что каждая группа содержит изображения лица только одного человека. Исследованы методы агрегации векторов признаков каждого кадра, извлекаемых с помощью глубокой сверточной нейронной сети. Треки, содержащие одинаковые лица, группируются с использованием методов верификации лиц и алгоритмов последовательной кластеризации. В экспериментальном исследовании с набором данных YouTubeFaces рассматриваются несколько способов объединения отдельных кадров для получения дескриптора видеодорожки. Показано, что наиболее высокую точность показывает сравнение нормализованных признаков, полученных с помощью усреднения векторов признаков всех кадров каждого трека.

## 1. Введение

В последнее время в связи с возникшим ростом объемов мультимедийных данных все большее внимание привлекает задача создания автоматической системы для организации информации. Некоторые приложения, такие как Google Photos, предоставляют возможность поиска, упорядочивания и демонстрации изображения конкретного человека. Среди таких технологий особенно востребованным являются системы распознавания лиц по видео в сфере обеспечения общественной безопасности [1, 2]. Увеличение объема накопленных видеоданных привело к необходимости решения задачи их упорядочивания [3]. Например, системы видеонаблюдения за несколько секунд поступают на вход сотни кадров [4, 5, 6]. Поэтому все большее внимание привлекает задача группировки изображений посетителей, чьи лица были замечены видеокамерой [7].

В связи с этим в настоящей работе рассматривается задача автоматической группировки видеоданных, на которых присутствует один человек, на основе методов кластерного анализа. Главная часть кластеризации это правило, по которому определяется, что несколько видео треков содержат изображения одного человека. Данная подзадача может быть выполнена с использованием глубоких сверточных нейронных сетей, которые показали высокую точность при решении многих сложных задач распознавания изображений [8]. На данный момент наблюдается тенденция в создании новых, более глубоких и широких, видов архитектур сверточных нейронных сетей [9, 10]. Таким образом, главная цель работы – произвести сравнительный анализ алгоритмов по верификации лиц, в частности, различных способов агрегирования признаков, полученных из каждого кадра видеопотока.

## 2. Методы агрегации векторов признаков видеокадров

Исследуемая в настоящей работе задача состоит в том, чтобы разбить этот набор на $M<T$ последовательных треков $\{X(m)\}$, $m$ = 1,2, ..., $M$, содержащих изображения лиц одного человека, а затем объединить похожие треки в кластеры. Каждый $m$-th трек характеризуется индексами начала $t_1(m)$ и конца $t_2(m)$.

В настоящей работе используется схема решения задачи, представленная на Рисунке 1.

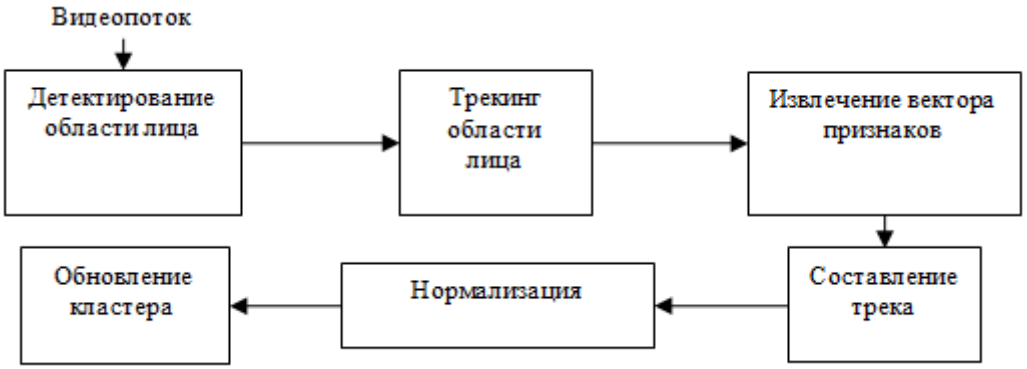


**Рисунок 1.** Схема автоматического упорядочивания видеоданных.

Вначале на каждом кадре необходимо обнаружить лица. Для этого были использованы моделей из TensorFlow Models [11]. Этот ресурс содержит разнообразные предварительно обученных нейросетевые модели, а также предоставляет интерфейс для детектирования объектов TensorFlow Object Detection API. Обнаружение лиц происходит на основе MobileNet SSD с предварительно подготовленной моделью, обученной на наборе данных WiderFace [12]. Далее осуществляется трекинг выделенных лиц, но детектирование лиц периодически повторяется, чтобы: 1) уточнить результаты отслеживания; 2) искать новые лица и 3) отметить исчезнувшие лица.

Для решения задачи группировать треки, содержащие изображения одного и того же человека использовались методы кластеризации [13, 14], для применения которых необходимо извлечь признаки области лица из каждого кадра, агрегировать признаки отдельного кадра в дескриптор для всего трека и затем сопоставлять эти дескрипторы. В настоящее время для извлечения признаков изображения используют сверточные нейронные сети. Для задач распознавания лиц существует множество уже обученных сетей на огромных базах данных Casia WebFaces, MS-Celeb-1M и т.п.[15, 16, 17]. На выходе предпоследнего (обычно полносвязного) слоя сети образуется вектор признаков $\mathbf{x}(t)$ размерности $N$. Для их сравнения зачастую используют метрику Евклида ($L_2$) $\rho(\mathbf{x}(t_1), \mathbf{x}(t_2))$ [15]. Однако для кластеризации видеотреков необходимо определять расстояние $\rho(X(m_1), X(m_2))$ между последовательностями кадров $X(m_1)$ и $X(m_2)$ в общем случае различной длины. Например, можно воспользоваться усреднением попарных расстояний между всеми кадрами:

$$\rho\left(X(m_1), X(m_2)\right) = \frac{1}{\Delta t(m_1)\Delta t(m_2)} \sum_{t=t_1(m_1)}^{t_2(m_1)} \sum_{t'=t_1(m_2)}^{t_2(m_2)} \rho\left(\mathbf{x}(t), \mathbf{x}(t')\right) \quad (1)$$

К сожалению, вычислительная сложность такого подхода оказывается достаточно велика в связи с необходимостью сопоставления $\Delta t(m_1)\Delta t(m_2)$ расстояний между векторами признаков высокой размерности. Поэтому в настоящей работе были реализованы следующие методы:

1. Вычисление расстояния между медоидами каждого трека:

$$\rho\left(X(m_1), X(m_2)\right) = \rho\left(\mathbf{x}^*(m_1), \mathbf{x}^*(m_2)\right), \mathbf{x}^*(m_i) = \underset{\mathbf{x}(t), t\in[t_1(m_i), t_2(m_i)]}{\operatorname{argmin}} \sum_{t'=t_1(m_i)}^{t_2(m_i)} \rho\left(\mathbf{x}(t), \mathbf{x}(t')\right) \quad (2)$$

2. Сравнение средних векторов признаков каждого трека:

$$\rho\big(X(m_1), X(m_2)\big) = \rho\big(\bar{\mathbf{x}}(m_1), \bar{\mathbf{x}}(m_2)\big), \qquad \bar{\mathbf{x}}(m_i) = \frac{1}{\Delta t(m_i)} \sum_{t'=t_1(m_i)}^{t_2(m_i)} \mathbf{x}(t) \tag{3}$$

3. Сравнение медиан $\mathbf{x}'(m_i)$ каждого трека:

$$\rho\big(X(m_1), X(m_2)\big) = \rho\big(\mathbf{x}'(m_1), \mathbf{x}'(m_2)\big). \tag{4}$$

Для того, чтобы сделать признаки более устойчивыми к условиям наблюдения (разрешение камеры, освещение и т.п.) обычно применяют их нормировку в метрике Евклида [10]. Обычно выполняется предварительная нормировка признаков каждого кадра. Однако в настоящей работе исследуется также нормировка агрегированных признаков (3) [18].

На заключительном этапе (**Рисунок 1**) происходит последовательная кластеризация: вектор признаков последнего трека сопоставляется с признаками ранее обнаруженных кластеров. Если расстояние до ближайшего кластера не превышает определенного порога, этот трек добавляется в кластер, а информация о нем обновляется.

## 3. Экспериментальные результаты

В настоящем разделе продемонстрированы результаты работы предлагаемого алгоритма (Рис. 1). Разработка и тестирование системы проводилось на языке C++ с помощью MS Visual Studio 2015 с использованием библиотеки OpenCV [17]. Среднее время детектирования лиц на одном кадре на ПК Lenovo ideapad 310, 64-разрядной операционной системе Windows 10 составляет 60 мс. Для извлечения признаков изображения использовалась библиотека Caffe [9] и две свободно доступных для распознавания лиц сверточные нейронные сети: VGGNet [16] и Lightened CNN (версия C) [10]. Сеть VGGNet извлекает $D = 4096$ вектор признаков со слоя “fc7” из 224x224 RGB изображений. С помощью Lightened CNN извлекаются $D = 256$ вектора признаков со слоя “eltwise_fc2” из 128x128 grayscale изображения. Преимуществами их использования является быстрая скорость обработки одного изображения и высокая точность распознавания.

В работе исследовались два типа расстояний между кадрами: традиционная метрика $L_2$ (Евклида) и критерий Стьюдента (t-test):

$$t = \frac{\rho(X(m_1), X(m_2))}{\sqrt{\frac{D(m_1)}{\Delta t(m_1)} + \frac{D(m_2)}{\Delta t(m_2)}}} \tag{5}$$

Эксперименты проводились на наборе данных YTF (YouTubeFaces) [19], который содержит 3425 видео 1595 различных людей. Самая короткая продолжительность трека составляет 48 кадров, самая длинная – 6070 кадров, средняя продолжительность видеоклипа – 181.3 кадра. Были найдены следующие показатели: AUC (Area under curve), FRR (False Reject Rate) для фиксированного FAR (False Accept Rate) = 1%. Результаты для моделей Lightened CNN и VGGNet приведены в Таблице 1 и Таблице 2, соответственно.

Результаты демонстрируют, что следует придать особое значение нормировки векторов признаков. Наиболее эффективным алгоритмом является вычисление средних признаков трека с последующей нормировкой (AvePool (3)->$L_2$-norm).

Для набора YTF извлечение признаков с помощью Lightened CNN оказалось предпочтительнее по сравнению с VGGNet. Сверточная нейронная сеть Lightened CNN позволяет принять решение намного быстрее (Таблица 3).

Для кластеризации была реализована агломеративная иерархическая кластеризация, в которой порог для определения результирующих кластеров определялся по фиксированному значению FAR. Кроме того, использовался алгоритм кластеризации [20] из библиотеки DominantSet [21]. Результаты сведены в Таблице 4.

Здесь общее число кластеров больше, чем количество различных людей из набора данных YTF в связи с тем, что различные видео одного человека могли попасть в разные кластеры.

**Таблица 1.** Результаты верификации лиц по видео, Lightened CNN.

| Мера близости треков | Расстояние между кадрами | AUC (%) | FRR@FAR=1% |
|---|---|---|---|
| Расстояние (1) | $L_2$ | 90.7±0.6 | 77.0±8.4 |
| $L_2$-нормировка->Расстояние (1) | $L_2$ | 98.2±0.4 | 14.1±3.6 |
| Медоиды (2) | $L_2$ | 89.7±0.6 | 80.6±6.4 |
| | t-test | 84.7±0.7 | 72.9±7.8 |
| $L_2$-нормировка (2) медоидов | $L_2$ | 97.2±0.6 | 19.1±4.3 |
| | t-test | 88.8±0.6 | 54.1±5.9 |
| Усреднение признаков(3) | $L_2$ | 91.3±1.3 | 71.8±10.0 |
| | t-test | 91.8±1.4 | 72.3±11.5 |
| Усреднение $L_2$-нормированных признаков (3) | $L_2$ | 97.7±0.5 | 21.4±6.4 |
| | t-test | 96.8±0.5 | 37.2±7.6 |
| $L_2$-нормировка среднего вектора признаков (3) | $L_2$ | 98.3±0.7 | 12.4±3.1 |
| | t-test | 97.6±0.5 | 12.5±3.1 |
| $L_2$-нормировка медианы (4) | $L_2$ | 96.7±0.6 | 22.3±7.2 |
| | t-test | 94.4±0.5 | 37.0±7.5 |

**Таблица 2.** Результаты верификации лиц по видео, VGGNet.

| Мера близости треков | Расстояние между кадрами | AUC (%) | FRR@FAR=1% |
|---|---|---|---|
| Расстояние (1) | $L_2$ | 83.3±0.8 | 85.8±9.0 |
| $L_2$-нормировка->Расстояние (1) | $L_2$ | 97.9±0.6 | 23.2±6.3 |
| Медоиды (2) | $L_2$ | 85.7±1.8 | 86.0±8.4 |
| | t-test | 80.8±1.2 | 83.9±7.7 |
| $L_2$-нормировка (2) медоидов | $L_2$ | 93.5±1.1 | 25.4±6.2 |
| | t-test | 85.2±0.7 | 69.9±7.9 |
| Усреднение признаков(3) | $L_2$ | 89.1±0.9 | 79.7±7.8 |
| | t-test | 87.4±1.2 | 81.2±5.8 |
| Усреднение $L_2$-нормированных признаков (3) | $L_2$ | 97.2±0.6 | 54.4±6.3 |
| | t-test | 96.3±0.7 | 76.9±6.8 |
| $L_2$-нормировка среднего вектора признаков (3) | $L_2$ | 98.1±1.0 | 19.4±5.9 |
| | t-test | 97.7±0.6 | 25.3±7.8 |
| $L_2$-нормировка медианы (4) | $L_2$ | 96.2±0.7 | 32.4±6.5 |
| | t-test | 94.8±0.7 | 41.1±7.3 |

**Таблица 3.** Среднее время (сек.) верификации видео.

| Мера близости треков | Расстояние между кадрами | Lightened CNN | VGGNet |
|---|---|---|---|
| Расстояние (1) | $L_2$ | 0.017 | 0.28 |
| $L_2$-нормировка->Расстояние (1) | $L_2$ | 0.016 | 0.26 |
| Медоиды (2) | $L_2$ | 0.029 | 0.54 |
| | t-test | 0.41 | 5.3 |
| $L_2$-нормировка (2) медоидов | $L_2$ | 0.03 | 0.57 |
| | t-test | 0.043 | 0.71 |
| Усреднение признаков(3) | $L_2$ | 0.2 | 3.0 |
| | t-test | 0.017 | 0.25 |
| Усреднение $L_2$-нормированных признаков (3) | $L_2$ | 0.019 | 0.34 |
| | t-test | 0.017 | 0.25 |
| $L_2$-нормировка среднего вектора признаков (3) | $L_2$ | 0.014 | 0.19 |
| | t-test | 0.015 | 0.15 |
| $L_2$-нормировка медианы (4) | $L_2$ | 0.011 | 0.10 |
| | t-test | 0.013 | 0.12 |

**Таблица 4.** Результаты кластеризации.

| FAR(%) | Lightened CNN | | VGGNet | |
|---|---|---|---|---|
| | **Общее число кластеров** | **Число неверных кластеров** | **Общее число кластеров** | **Число неверных кластеров** |
| Иерархическая кластеризация, FAR=1% | 2492 | 35 | 2634 | 44 |
| Иерархическая кластеризация, FAR=10% | 2147 | 162 | 2195 | 171 |
| DominantSet | 2372 | 31 | 2403 | 38 |

Кроме того, заметим, что средняя продолжительность работы алгоритма для обработки всех треков YTF для иерархической кластеризации составляет 8 минут, в то время как алгоритм DominantSet для группировки всех видео занял более 3 часов.

## 4. Заключение

В работе исследована задача кластеризации видеопоследовательностей в системах видеонаблюдения. В частности, основной акцент был сделан на вычислении степени близости видеотреков с использованием агрегации векторов признаков, извлеченных с помощью глубоких сверточных нейронных сетей. Эксперименты продемонстрировали, что наибольшей точностью и вычислительной эффективностью для задачи верификации пользователя по видеоизображению лица характеризуется усреднение векторов признаков всех кадров трека с последующей нормировкой. В дальнейшим планируется провести более подробное исследование различных алгоритмов кластеризации для достижения низкой вычислительной сложности и высокой точности обработки данных.

## 5. Благодарности

Статья подготовлена в результате проведения исследования (№ 17-05-0007) в рамках Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2017 г. и в рамках государственной поддержки ведущих университетов Российской Федерации "5-100".

## 6. Литература

[1] Chellappa, R. Face Tracking and Recognition in Video / R. Chellappa, M. Du, P. Turaga, S.K. Zhou // Handbook of Face Recognition. – 2011. – P. 323-351.

[2] Shan, C. Face Recognition and Retrieval in Video // Video Search and Mining, Studies in Computational Intelligence. – 2010. – Vol. 287. – P. 235-260.

[3] Savchenko, A.V. Search Techniques in Intelligent Classification Systems / A.V. Savchenko // Springer International Publishing. – 2016.

[4] Chen, J.C. An end-to-end system for unconstrained face verification with deep convolutional neural networks / J.C. Chen, R. Ranjan, A. Kumar, C.H. Chen, V.M. Patel, R. Chellappa // IEEE International Conference on Computer Vision Workshops. – 2015. –P. 118-126.

[5] Li, H. Eigen-PEP for video face recognition / H. Li, G. Hua, X. Shen, Z. Lin, J. Brandt // Asian Conference on Computer Vision (ACCV 2014). – LNCS. – Vol. 9005. – P. 17-33.

[6] Savchenko, A.V. Deep neural networks and maximum likelihood search for approximate nearest neighbor in video-based image recognition / A.V. Savchenko // Optical Memory and Neural Networks (Information Optics). – 2017. – Vol. 26(2). – P. 129-136.

[7] Savchenko, A.V. Organizing Multimedia Data in Video Surveillance Systems Based on Face Verification with Convolutional Neural Networks / A.V. Savchenko, A.D. Sokolova, A.S. Kharchevnikova // AIST 2017. LNCS. – Vol. 10716. – P. 213-220. (in print)

[8] Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин. – Издательский дом Вильямс, 2008.

[9] Y. Jia. Caffe: Convolutional architecture for fast feature embedding // Proceedings of the 22nd ACM international conference on Multimedia. – 2014. – P. 675-678.
[10] Wu, X. A Lightened CNN for Deep Face Representation / X. Wu, R. He, Z. Sun // arXiv preprint arXiv: 1511.02683. – 2015.
[11] TensorFlow Object Detection API [Электронный ресурс]. – Режим доступа: http://github.com/tensorflow/models/tree/master/research/object_detection (1.11.2017).
[12] Wider Face: A face detection benchmark. – Режим доступа: http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace (7.11.2017).
[13] Kaufman, L. Finding groups in data: an introduction to cluster analysis / L. Kaufman, P.J. Rousseeuw // John Wiley & Sons, 2009.
[14] Savchenko, A.V.: Clustering and maximum likelihood search for efficient statistical classification with medium-sized databases // Optimization Letters. – 2017. – Vol. 11(2). – P. 329-341.
[15] Goodfellow, I. Deep learning / I. Goodfellow, Y. Bengio, A. Courville // MIT press, 2016.
[16] Parkhi, O.M. Deep face recognition / O.M. Parkhi, A. Vedaldi, A. Zisserman // Proceedings of the British Machine Vision. – 2015. – P. 6-17.
[17] OpenCV: библиотека алгоритмов компьютерного зрения [Электронный ресурс]. – Режим доступа: http://opencv.org (2.08.2016).
[18] Savchenko, A.V. Statistical testing of segment homogeneity in classification of piecewise-regular objects / A.V. Savchenko, N.S. Belova // International Journal of Applied Mathematics and Computer Science. – 2015. – Vol. 5(4). – P. 915-925.
[19] Wolf, L. Face recognition in unconstrained videos with matched background similarity / L. Wolf, T. Hassner, I. Maoz // IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). – 2011. – P. 529-534.
[20] Pelilo, M. Dominant sets and pairwise clustering / M. Pelilo, M. Pavan // PAMI. – 2007.
[21] DominantSetLibrary. – Режим доступа: https://github.com/xwasco/DominantSetLibrary (7.11.2017).

# Data organization in video surveillance systems using deep learning technologies

**A.D. Sokolova[1], A.V. Savchenko[1]**

[1]National Research University Higher School of Economics, Bolshaya Pecherskaya street, 25/12, Nizhny Novgorod, Russia, 603155

**Abstract.** The task of organizing information in video surveillance systems is implemented by grouping the video tracks, which contain identical faces. We examine aggregation methods for the features of individual frames extracted using deep convolutional neural networks. The tracks with identical faces are grouped based on known face verification algorithms and clustering methods. Experimental study on the YouTubeFaces dataset demonstrates results of combining frame features in order to obtain a descriptor of video track. It is shown that the most accurate method is $L_2$-normalization of average unnormalized features of individual frames of each video track.

**Keywords:** convolutional neural networks, deep learning, face recognition, clustering, detection, verification methods.