Об одном подходе к решению задачи детектирования лиц на изображениях*

***Е.А. Долотов*[1], *В.Д. Кустикова*[1]**
dolotov.evgeniy@gmail.com|valentina.kustikova@itmm.unn.ru
[1]Институт информационных технологий, математики и механики,
Нижегородский государственный университет им. Н.И. Лобачевского, Нижний Новгород, Россия

Рассматривается задача детектирования лиц на изображениях. Описывается общая схема решения задачи с использованием метода, основанного на построении модели деформируемых частей на базе глубокой пирамиды признаков. Метод относится к числу передовых, поскольку демонстрирует одни из лучших результатов на широко известных данных Face Detection Data Set and Benchmark (FDDB). Разрабатывается программная реализация метода и приводится ее описание. Воспроизводятся результаты детектирования лиц с использованием указанной реализации на FDDB. Эксперименты показывают сравнимые значения показателей качества с опубликованными на официальной странице набора данных.

Ключевые слова: *детектирование лиц, глубокое обучение, сверточные нейронные сети, метод опорных векторов*

One approach to solving the problem of face detection*

***E.A. Dolotov*[1], *V.D. Kustikova*[1]**
[1]Institute of information technologies, mathematics and mechanics,
Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia

The problem of face detection is considered. We describe the general scheme of solving the problem by using a method based on constructing a deformable part-based model on the deep feature pyramid. Method refers to the number of advanced, because it demonstrates some of the best results on the well-known Face Detection Data Set and Benchmark (FDDB). We provide software implementation of the method and its description. Results of face detection are reproduced on FDDB using the developed implementation. Experiments show comparable quality values published on the official page of the dataset.

Keywords: *face detection, deep learning, convolutional neural networks, support vector machines*

Введение

Задача детектирования лиц на изображениях является одной из классических задач компьютерного зрения. Решение данной задачи находит широкое практическое применение. Функция автоматического поиска лиц на фотографиях активно используется в программном обеспечении цифровых фото- и видеокамер, в системах управления фотоальбомами, в различных видеорегистраторах и системах видеонаблюдения.

В данной работе приводится общая схема алгоритма, предложенного в [8], разрабатывается программная реализация указанного алгоритма, дается анализ качества детектирования лиц на тестовом множестве Face Detection Data Set and Benchmark (FDDB) [2]. Разработка ведется на базе широко известной библиотеки компьютерного зрения OpenCV [7] и библиотеки глубокого обучения Caffe [1].

Схема решения задачи

Алгоритм детектирования лиц [8] основан на применении метода скользящего окна и бинарной классификации отдельных областей, накрываемых окном.

Классификация включает два основных этапа:

1. Обучение модели лица на *тренировочном наборе данных*. Для этого осуществляется *извлечение признаков* из изображений с помощью сверточной нейронной сети (Convolutional Neural Network, CNN) и обучение линейной машины опорных векторов (Support Vector Machine, SVM).
2. Принятие решения для нового изображения о его принадлежности классу лиц. На данном этапе выполняется извлечение признаков из изображения согласно тому же алгоритму, что и при обучении классификатора. Далее обученный классификатор определяет, является ли входное изображение изображением лица.

Метод скользящего окна предполагает проход по изображению с перекрытиями, поэтому для одного и того же лица может быть обнаружено несколько окаймляющих прямоугольников, для которых соответствующие области проклассифици-

Работа выполнена в лаборатории «Информационные технологии» Института информационных технологий, математики и механики ННГУ им. Н.И. Лобачевского при поддержке компании Itseez с использованием ресурсов суперкомпьютерного комплекса МГУ им. М.В. Ломоносова [10].

рованы как лица. В связи с этим выполняется *объединение этих прямоугольников* и *уточнение их границ*. Рассмотрим более детально каждый шаг алгоритма детектирования.

Извлечение признаков. Вначале для исходного изображения строится пирамида изображений. Пирамида состоит из семи уровней. Изображение на каждом уровне имеет разрешение $1713 \times \times 1713$ пикселей. Размер исходного изображения изменяется так, чтобы его наибольшая сторона состояла из 1713 пикселей, после чего оно помещается в верхний левый угол изображения на седьмом уровне пирамиды, оставшаяся часть изображения окрашивается в черный цвет. Далее изображение уменьшается в $\sqrt{2}$ раз от уровня к уровню и также помещается в левый верхний угол (рис. 1).

Изображение с каждого уровня подается на вход сверточной нейронной сети, которая получена из сети AlexNet [6] посредством удаления последних полносвязных слоев. После обработки пирамиды изображений с помощью сверточной нейронной сети формируется *пирамида карт признаков*, также состоящая из семи уровней. Каждая карта признаков представляет из себя набор из 256 квадратных матриц $X \in \mathbb{R}^{108\times108}$. Каждый вектор признаков $x_{i,j,k}$, находящийся на позиции (j,k) на уровне i пирамиды признаков, нормализуется по следующей формуле:

$$\widetilde{x}_{i,j,k} = \frac{x_{i,j,k} - \mu_i}{\sigma_i}, \qquad (1)$$

где μ_i – среднее значение вектора признаков, σ_i – среднеквадратическое отклонение для уровня i пирамиды признаков.

Обучение линейной машины опорных векторов. На основе построенных признаковых описаний тренировочных изображений осуществляется обучение классификатора, в качестве которого в [8] используется линейная машина опорных векторов. При обучении для каждого изображения в тренировочном множестве извлекается одна карта признаков, соответствующая одному лицу. Для этого сначала находится оптимальный уровень пирамиды, удовлетворяющий следующему соотношению:

$$l = \arg\min_i |b_i^y - h| + |b_i^x - w|, \qquad (2)$$

где (h,w) – размеры прямоугольника, используемого при классификации, а (b_i^y, b_i^x) – размеры прямоугольника, содержащего лицо, на уровне i пирамиды изображений. Далее с уровня l в пирамиде нормализованных карт признаков извлекается одна карта признаков с размерами (b_i^y, b_i^x), соответствующая лицу. После этого все карты признаков масштабируются к размеру (h,w).

Карты признаков, которые используются в качестве негативов при обучении, извлекаются случайным образом из нормализованной карты признаков. Карта признаков считается принадлежащей к множеству негативов, если она пересекается менее, чем на 30% с картами, соответствующими лицам.

Следует отметить, что поскольку объем тренировочных данных велик, линейная машина опорных векторов вначале использует лишь небольшую часть примеров для обучения. Затем к тренировочному множеству добавляется часть неправильно проклассифицированных изображений, и обучение повторяется вновь.

Детектирование лиц. В процессе детектирования лиц с использованием построенного классификатора осуществляется построение пирамиды признаков для вновь полученного изображения. Далее к пирамиде применяется метод скользящего окна. Осуществляется проход с единичным шагом по нормализованным картам признаков, извлекаются все прямоугольные области размера (h,w). Полученные карты признаков преобразуются в вектора из $h \times w \times 256$ элементов, которые затем классифицируются с помощью обученной линейной машины опорных векторов. При этом расстояние до гиперплоскости, разделяющей два класса, принимается за достоверность принадлежности к классу лиц.

Объединение прямоугольников. После прохода скользящим окном каждому лицу на изображении может соответствовать несколько прямоугольников. В ходе объединения прямоугольников результатом может являться прямоугольник, имеющий наибольшую достоверность принадлежности к классу лиц, либо прямоугольник, все параметры которого определяются как среднее арифметическое соответствующих параметров всех прямоугольников из множества, полученного для данного лица.

Уточнение границ прямоугольников. Для уточнения границ прямоугольников используется алгоритм, описанный в [3]. Данный алгоритм приближает прямоугольник, полученный в результате детектирования, к соответствующему прямоугольнику из разметки, используя при этом параметры самого прямоугольника, а также соответствующий ему вектор признаков.

Пусть прямоугольник P описывается набором из четырех параметров $P = (P_x, P_y, P_w, P_h)$, где (P_x, P_y) – координаты центра прямоугольника, а (P_w, P_h) – его размеры. Тогда для нахождения уточненного прямоугольника $\widetilde{P}$ используются сле-

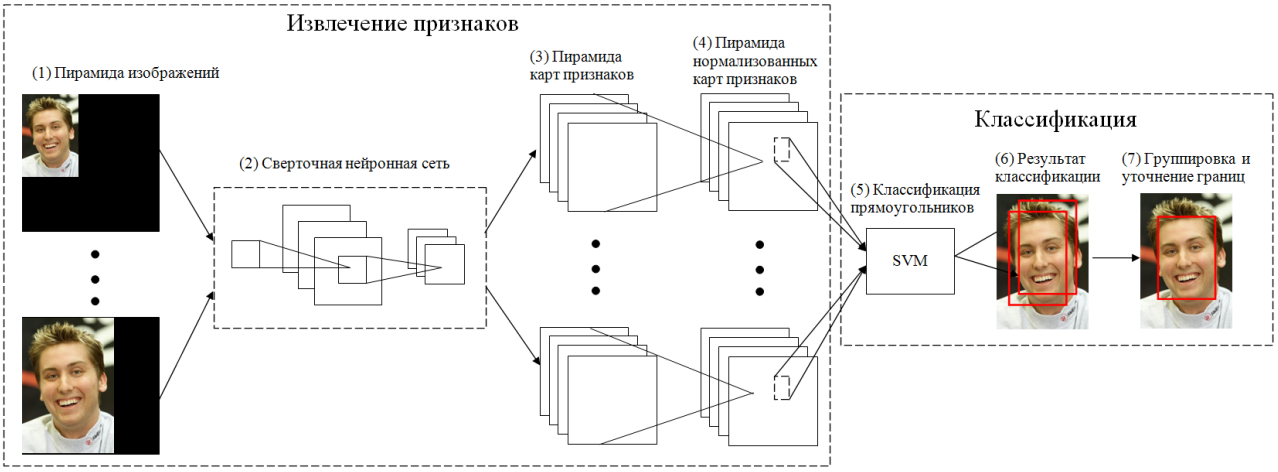

Рис. 1. Общая схема детектирования лиц.

дующие соотношения:

$$\begin{aligned} \widetilde{P}_x &= P_w d_x(P) + P_x, \\ \widetilde{P}_y &= P_h d_y(P) + P_y, \\ P_x &= P_w e^{d_w(P)}, \\ P_y &= P_h e^{d_h(P)}, \end{aligned} \tag{3}$$

где каждая функция $d_*(P)$ является линейной функцией вида $d_*(P) = w_*^T \varphi(P)$, зависящей от вектора признаков $\varphi(P)$, который соответствует прямоугольнику P, и вектора настраиваемых параметров w_*.

Для нахождения параметров используется множество пар (P^i, G^i), где P^i – прямоугольник, полученный в результате детектирования, а G^i – соответствующий прямоугольник из разметки. Тогда вектор параметров w_* представляет собой решение следующей задачи оптимизации:

$$\begin{aligned} w_* &= \arg\min_{\widetilde{w}_*} \sum_{i=1}^{N} (t_*^i - \widetilde{w}_*^T \varphi(P^i))^2 + \lambda ||\widetilde{w}_*||^2, \\ t_x &= \frac{G_x - P_x}{P_w}, t_y = \frac{G_y - P_y}{P_h}, \\ t_w &= \log \frac{G_w}{P_w}, t_h = \log \frac{G_h}{P_h}. \end{aligned} \tag{4}$$

Решение данной задачи представимо в виде:

$$w_* = (X^T X + \lambda E)^{-1} X^T Y, \tag{5}$$

где $X \in \mathbb{R}^{N \times p}$ – матрица, состоящая из векторов признаков, расположенных по строкам, $E \in \mathbb{R}^{p \times p}$ – единичная матрица, $Y \in \mathbb{R}^N$ – вектор, состоящий из элементов t_*.

Программная реализация

Программная реализация описанного алгоритма выполнена с использованием языка программирования C++ [11] и содержит следующие программные модули (рис. 2):

1. *DeepPyramid*. Обеспечивает построение пирамиды изображений и последовательно запускает все этапы работы алгоритма.
2. *NeuralNetwork*. Осуществляет построение пирамиды признаков посредством вызова функций библиотеки глубокого обучения Caffe [1]. Модуль *FeatureMap* содержит примитивы для представления пирамиды карт признаков.
3. *FeatureMapSVM*. Реализует метод скользящего окна и обеспечивает классификацию отдельных областей, накрываемых окном, с помощью линейной машины опорных векторов. Использует реализацию метода опорных векторов из библиотеки компьютерного зрения OpenCV [7].
4. *NMS*. Содержит реализацию нескольких стратегий объединения прямоугольников.
5. *BoundingBoxRegressor*. Реализует этап уточнения границ прямоугольников.

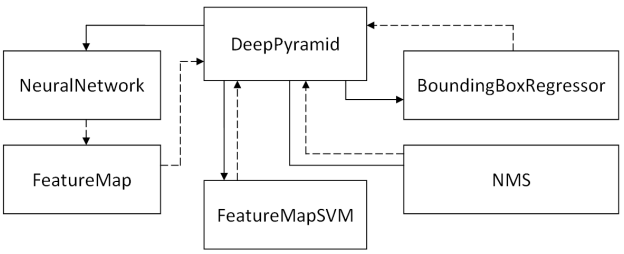

Рис. 2. Схема функционирования системы.

Критерий оценки качества поиска

Для оценки качества детектирования с помощью разработанной программной реализации используется показатель, равный отношению площади пересечения прямоугольника, полученного в результате детектирования, и прямоугольника из разметки к площади их объединения (Intersection

over Union, IoU):

$$IoU(d_i, l_j) = \frac{area(d_i \cap l_j)}{area(d_i \cup l_j)}, \quad (6)$$

где d_i – прямоугольник, полученный в результате детектирования, а l_i – прямоугольник из разметки. Таким образом, считается, что лицо обнаружено правильно, если данный показатель превышает некоторый порог. В противном случае принимается, что лицо не обнаружено. Срабатывание алгоритма детектирования на области, где лицо отсутствует, считается ложным срабатыванием.

На основании приведенного показателя осуществляется построение ROC-кривой, которая отражает зависимость количества ложных срабатываний алгоритма детектирования (false positive) от точности детектирования (true positive rate). Построение ROC-кривых обеспечивается с помощью инструментов, предоставляемых разработчиками наборов данных, используемых в ходе апробации.

Тренировочные и тестовые данные

В процессе проведения экспериментов для обучения классификатора и тестирования разработанной реализации алгоритма детектирования используется набор данных Face Detection Data Set and Benchmark (FDDB) [2], который состоит из 2845 изображений, содержащих 5171 лицо, максимальное и минимальное разрешение которых составляет 398×589 и 8×13 пикселей соответственно.

Результаты экспериментов

Полученные результаты экспериментов приведены ниже (рис. 3, 4). На графике показаны ROC-кривые (рис. 3, снизу вверх), соответствующие разработанной реализации, реализации авторов метода, а также лучшим результатам в настоящий момент.

Построенные ROC-кривые говорят о сравнимости результатов с [8]. При одинаковом количестве ложных срабатываний точность детектирования в среднем меньше на 2.7%. Различие может объясняться тем, что использованы разные предобученные модели сети AlexNet [6] (разные веса сетей приводят к отличиям в значениях признаков), а также отличиями в реализации стратегий объединения прямоугольников. Данный вопрос является одним из предметов дальнейших исследований.

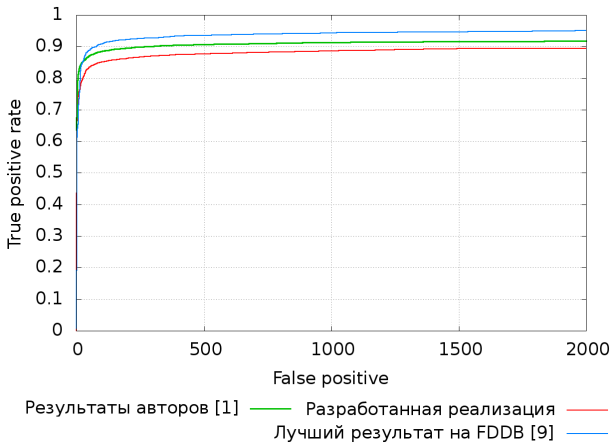

Рис. 3. Сравнение полученных результатов с результатами авторов.

Следует отметить, что при выполнении вычислений на центральном процессоре Intel(R) Core i5-2430M @ 2.4GHz и 6GB RAM детектирование лиц на одном изображении вне зависимости от разрешения занимает 45 секунд. Использование графического процессора NVIDIA Tesla X2070 на этапе извлечения признаков позволяет сократить время работы до 9 секунд.

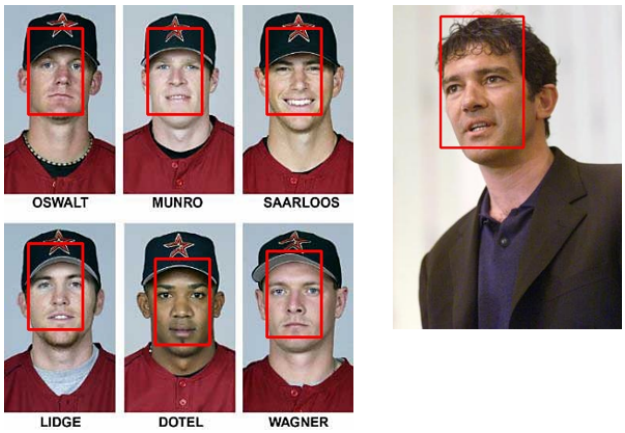

Рис. 4. Пример работы реализации на изображениях из базы FDDB.

Заключение

В данной работе рассмотрен один из передовых методов решения задачи детектирования лиц [8]. Разработана программная реализация этого метода, основанная на библиотеке компьютерного зрения OpenCV [7] и библиотеке глубокого обучения Caffe [1]. Воспроизведены результаты детектирования лиц с использованием указанной реализации на данных FDDB. Эксперименты показывают сравнимые значения показателей качества с опубликованными на официальной странице набора данных [2].

Программная реализация выложена в открытый доступ [11].

В дальнейшем планируется более детально изучить причины небольших различий в полученных результатах качества, провести апробацию алгоритма на других наборах данных [4, 5], а также разработать модификации алгоритма с целью повышения качества детектирования лиц.

Литература

[1] Caffe Framework – http://caffe.berkeleyvision.org.

[2] Face Detection Data Set and Benchmark Home – http://vis-www.cs.umass.edu/fddb.

[3] *Girshick R., Donahue J., Darrell T., Malik J.* Rich feature hierarchies for accurate object detection and semantic segmentation // In the Proceedings of the Conference on Computer Vision and Pattern Recognition. – 2014.

[4] *Huang G.B., Ramesh M., Berg T., Miller E.L.* Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments // Technical Report 07-49, University of Massachusetts, Amherst. – 2007.

[5] *Kostinger M., Wohlhart P., Roth P.M., Bischof H.* Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization // ICCV-Workshop. – 2011.

[6] *Krizhevsky A., Sutskever I., Hinton G.* ImageNet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. – 2012.

[7] OpenCV – http://opencv.org.

[8] *Ranjan R., Patel V.M., Chellappa R.* A Deep Pyramid Deformable Part Model for Face Detection. – 2015. – http://arxiv.org/abs/1508.04389.

[9] *Zhang K., Zhang Zh., Li Z.* Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. – 2016. – http://arxiv.org/abs/1604.02878.

[10] *Воеводин Вл.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.А., Никитенко Д.А., Стефанов К.С., Воеводин Вад.В.* Практика суперкомпьютера “Ломоносов” // Открытые системы. – Москва: Изд. дом "Открытые системы". – No. 7. – 2012. – С. 36-39.

[11] Разработанная программная реализация https://github.com/DolotovEvgeniy/face-detection-model.