3. Huete, A.R. Acomparison of vegetation indices over a global set of TM images forEOS-MODIS /A.R. Huete [at al.] // Remote Sensing of Environment. – № 59. – P. 440–451.
4. Sang ‐ Il, Na. Estimating Leaf Area Index of Paddy Rice from RapidEye Imagery to Assess Evapotranspiration in Korean Paddy Fields / Na, Sang-Il [at al.] // Korean Journal of Soil Science and Fertilizer. – Volume 46, Issue 4. – 2013. – P. 245–252.
5. Haralick, R.M. Textural Features for Image Classification / R.M. Haralick, K. Shanmugam, I. Dinstein // IEEE Transactions on Systems, Man and Cybernetics. – 1973. – № 6. – P. 610–621.
6. Потапов, А.А. Фракталы в радиофизике и радиолокации / А.А. Потапов. – Москва : Логос, 2002. – 664 с.
7. Нигматуллин, Р.Р. Фракталы, дробные операторы и дробная кинетика в диэлектрической спектроскопии и волновых процессах / Р.Р. Нигматуллин, А.А. Потапов // Физика волновых процессов и радиотехнические системы. – 2007. – Т. 10, № 3. – С. 30–49.
8. SenseflyDatasets [Электронныйресурс]. – Режим доступа : https://www.sensefly.com/education/datasets/. – Дата доступа : 04.01.2018.
9. Parikh, D. An ensemble-based incremental learning approach to data fusion / D. Parikh, R. Polikar // IEEE Transactions on Systems, Man and Cybernetics – Part B: Cy-bernetics. – 2007. – Vol. 37. – Iss. 2. – P. 437–450.
10. Marushko, Y. Using Ensembles of Neural Networks with Different Scales of Input Data for the Analysis of Telemetry Data / Y. Marushko // Proc. of the XV Intern. PhD Workshop OWD 2013 (Wisla, 19–22 Oct. 2013). – Gliwice: Silesian University of Technology, 2013. – P. 386–391.
11. Neural network ensemble operators for time series forecasting / NikolaosKourentzes [et al.] // Expert Systems with Applications. – July 2014. – Vol.41, Iss. 9. – ISSN: 0957-4174. – P. 4235–4244.

Материал поступил в редакцию 20.03.2018

GANCHENKO V., DOUDKIN A., MARUSHKO Y. Construction of maps of agricultural fields on aerial photographs of different spectral range for precision farming systems

An algorithm for forming fuzzy descriptors for constructing crop maps for multispectral images is determined. An algorithm for learning the neural network recognition model is determined. The algorithm for mapping the crops is defined. The method of combining informative features of multispectral images for assessing the state of agricultural vegetation is based on the joint use of visible range data and a number of vegetative indices computed from images in the visible and infrared regions of the spectrum, as well as color and texture characteristics. Experimental results of application of the proposed algorithms are presented.

The NDVI index on the basis of the near infrared spectral range shows a higher information content with respect to the presence of the vegetation cover. Neural network model and fuzzy descriptors allow to more accurately form contours of recognized objects.

УДК 004.932.72'1

Кузьмицкий Н.Н.

ДЕТЕКТИРОВАНИЕ ТЕКСТОВЫХ ОБЪЕКТОВ НА ОСНОВЕ «НЕГЛУБОКОЙ» СВЕРТОЧНОЙ НЕЙРОСЕТИ С ОПТИМИЗАЦИЕЙ ВЫЧИСЛЕНИЙ

Введение. Технологии автоматической обработки текста разрабатываются на протяжении нескольких десятков лет, что привело к появлению эффективных прикладных систем (например, фирмы ABBYY). Однако главным образом они ориентированы на анализ машинного текста и изображений документов, при этом в целом ряде типовых приложений (обработке рукописного текста, техническом зрении роботов и др.) уровень эффективности все еще не достаточен как по точности, так и ресурсоемкости. Кроме того, регулярно появляются новые практические задачи (в частности, создание систем «дополненной реальности»), связанные с использованием портативных оптических устройств (например, видеокамер смартфонов), что стимулирует активное исследование соответствующих направлений науки. Математической основой предлагаемых решений зачастую является аппарат искусственных нейросетей ввиду сложности строгой формализации задач анализа текста в их широкой постановке. Подобный выбор также сделан в представленном исследовании детектирования текстовых объектов растровых изображений с произвольной композицией, вариативными яркостными и топологическими свойствами текста.

1. Анализ предыдущих результатов и постановка задачи. Выполненный обзор показал существенные ограничения применимости по-прежнему широко используемого в обработке текста OCR-подхода при наличии оптических искажений изображения. Так, расфокусировка вызывает увеличение степени фрагментированности контуров текстовых объектов, что усложняет их последующую локализацию, а при неравномерной освещенности весьма затруднительна качественная бинаризация. Предложены варианты решения данных проблем, путем модификации классических методов (в частности, Canny и Niblack), что повышает качество кластеризации классов «текст» и «фон» при яркостной неоднородности [1]. Однако данные меры все же недостаточны для анализа изображений реальных сцен ввиду вероятного перекрытия объектов, перспективных искажений, несоответствия ракурса съемки и т. п., направляющих поиск решений в область методов машинного обучения, а именно *сверточных нейросетей* (СНС).

Одним из отличительных свойств СНС является объединение в рамках одной нейросети двух этапов анализа: выделения высокоуровневых признаков объектов (происходящего в режиме «черного ящика») и их классификации, что представляется существенным преимуществом перед использованием упрощенных зрительных моделей, интуитивно выбираемых характеристик объектов и значений параметров порогового ограничения. Среди недостатков СНС можно выделить недостаточную формализацию, зависимость от представительных баз данных и значительных вычислительных ресурсов как для обучения, так и применения. Последнее обстоятельство в особенности актуально для весьма востребованных в настоящее время *глубоких нейросетей* (ГНС) [2], которые содержат большое количество вычислительных элементов, что затрудняет использование ГНС на стандартных ЭВМ и мобильных устройствах. Использование ГНС в практических приложениях (например, для распознавания речи) основано на наличии вычислительных ресурсов серверного оборудования с существенным ограничением функциональности в отсутствии соединения. При этом научные исследования проводятся средствами высокопроизводительного оборудования (минимально с помощью ЭВМ, оснащенной мощным процессором и расчетными графическими картами), требующего дополнительных финансовых затрат.

Перспективным направлением расширения применимости ГНС является их редукция, например, удаление «избыточных» по некоторым критериям нейронов или целых слоев обученной нейросети. При этом возникают вопросы как к формализации «избыточности»,

Кузьмицкий Николай Николаевич, *к.т.н., доцент кафедры «ЭВМ и системы» Брестского государственного технического университета. Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.*

так и к пониманию принципов внутренней организации ГНС, их обобщающей способности, наряду с целесообразностью применения масштабных вычислений и громоздких обучающих выборок. Альтернативной ГНС являются «компактные» СНС архитектуры подобной классическим (например, LeNet-5 [3]), широко изученные с теоретической точки зрения и апробированные во многих практических приложениях. Так, известно, что возможно создание классификаторов разнотипных текстовых образов в виде комитета компактных СНС [4]. Также предложен метод детектирования лиц в видеопотоке сверхвысокого разрешения с помощью каскада СНС, обеспечивающий высокую производительность обработки в режиме реального времени [5]. Однако ограниченная обобщающая способность компактных СНС зачастую приводит к ложным срабатываниям, как следствие, к необходимости создания эффективных алгоритмов постобработки решений СНС, учитывающих априорные данные о свойствах объектов рассматриваемых классов.

Таким образом, целью представленного исследования является создание текстового детектора в виде нейросетевой модели компактной архитектуры и алгоритмов локализации текстовых объектов на изображениях произвольных сцен. Дополнительное требование заключалось в оптимизации вычислительной сложности как самой модели, с учетом структуры и априорных данных, так и ее программной реализации для выбранной платформы.

2. Модель нейросетевого детектора текстовых объектов. СНС – это многослойные иерархические модели, функционирующие по принципам подобным биологическим зрительным системам, в основе которых лежат три архитектурные идеи:

- *локальные рецептивные поля*: нейроны текущего слоя получают входной сигнал от окрестностей нейронов предыдущего слоя, благодаря чему нейросеть обучается двумерной структуре входного образа;
- *разделяемые веса*: нейроны слоя объединены картами, в рамках которых они обладают общими весами, что сокращает количество настраиваемых параметров, при этом карты выделяют различные признаки образа;
- *пространственные подвыборки*: уменьшение размерности карт позволяет формировать высокоуровневые признаки и повышает устойчивость к искажениям.

Существуют различные типы СНС, отличающиеся топологией слоев и способом организации процесса обучения. Один из классических способов основан на применении метода Левенберга-Марквардта, увеличивающего сходимость процесса за счет индивидуальной настройки «скорости коррекции» весов нейронов, путем замедления обучение в «крутых» областях весового пространства и ускорения на плоских [3]:

$$w_{new}^k = w_{old}^k - \eta/(h_k + \mu)\cdot \partial E/\partial w^k , \qquad (1)$$

где w^k – k-й настраиваемый вес; η – глобальный параметр обучения;

h_k – оценка второй производной функции ошибки E по w^k;

μ – константа.

В качестве функции ошибки может применяться среднеквадратичное отклонение. При этом используемая в расчетах аппроксимация Гаусса-Ньютона приводит к неотрицательности гессиана (рассматриваются только диагональные элементы), а его вычисление аналогично обратному распространению первой производной функции ошибки. Оценивается гессиан перед каждой эпохой обучения и лишь на части тренировочного множества (в [3] предлагается применять 500 из 60 000 образов, выбранных случайно).

Для обнаружения текстовых объектов на изображении создана *модель текстового детектора* в форме сверточной нейросети, представленной на рис. 1:

1) на вход подаются нормированные к диапазону [-1, 1] и преобразованные в линейный вектор значения яркости пикселей полутонового изображения *I*;

2) выполняется распространение сигнала в сети путем чередования этапов свертки и подвыборки, и его прямой передачи в последних двух слоях;

3) значения откликов двух нейронов выходного слоя интерпретируются в качестве решения о наличии (отсутствии) на входном изображении текстового образа.

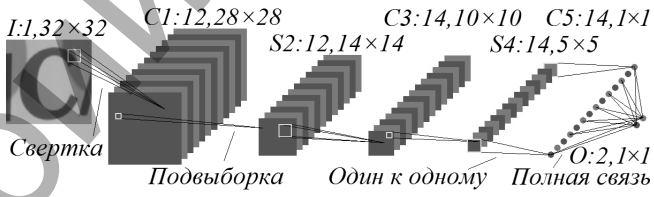


Рисунок 1 – Архитектура нейросетевого детектора текстовых объектов (для каждого слоя указано <метка: количество карт, размер карт>)

Первые четыре слоя являются экстрактором высокоуровневых признаков, а последние два – стандартным классификатором в виде многослойного персептрона. Всего модель содержит 409 912 связей и 2 272 настраиваемых параметра. Выбор размера входного изображения и количества карт в слоях объясняется стремлением к достижению баланса между высокой обобщающей способностью нейросети и эффективностью ее применения. Размер фильтров и окрестностей

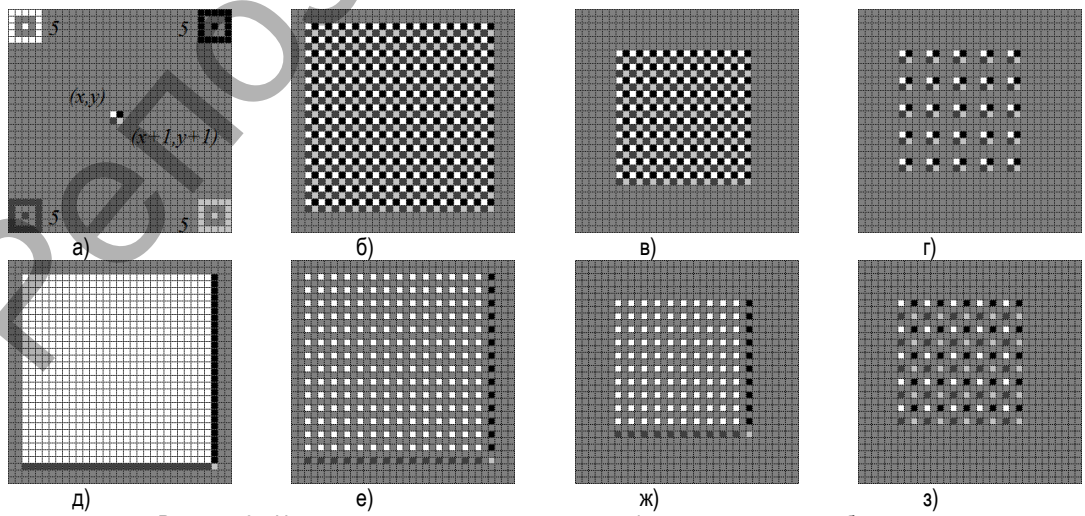


Рисунок 2 – Центры соседних локальных окон при фрагментировании изображения с шагом h = 1 (а) и структура вычисления откликов детектора для данных окон по слоям *C1*, *S2*, *C3*, *S4* с шагом h = 1 (д, б, в, г) и h = 2 (д, е, ж, з)

усреднения (подвыборок) определялся так, чтобы в результате послойного уменьшения масштаба карт количество нейронов в первом классифицирующем слое (C5) совпадало с числом карт этого слоя. Функцией активации нейронов является гиперболический тангенс ввиду наличия асимптот, симметричности и простого вычисления производной. Обучающее множество состояло из образов букв английского алфавита и цифр (ввиду их широкого использования), полученных из изображений реальных сцен пяти маркированных баз (ICDAR 2003, 74K, KAIST, SVHN, CVL OCR DB), в том числе: заглавных букв – 29 172 образа, цифр – 13 500, фоновых – 42 672 примера.

Перед запуском обучения весам нейросети присваиваются случайные значения, равновероятно распределенные на интервале $\left[-2{,}4/p_i, 2{,}4/p_i\right]$, где p_i – число связей *i*-го нейрона. Обучение проводится за 34 эпохи с начальным значением $\eta = 0{,}00085$, которое изменяется каждую эпоху путем умножения на коэффициент 0,85 (по аналогии с методикой из [3]). Коррекция весов проводится после обработки каждого образа (в online-режиме), а для ускорения обучения пропускается этап обратного распространения ошибки, если ее величина не превышает заданного значения ε. Точность классификации образов тренировочной выборки обученной нейросетью составила составила 96,21 %, тестовой – 94,95 %.

Качественная оценка модели показала, что она:

- выполняет посимвольное детектирование (описанная в [6] применима только для поиска регистрационных номеров автотранспорта) – возможность использования в различных задачах анализа текста, например в сегментации строк и слов;
- имеет компактную архитектуру (в 40 раз меньшую, чем у предложенной в [7]) – относительно низкая ресурсоемкость обучения и применения;
- способна обнаруживать наклоненные текстовые объекты (в отличие от представленной в [8]) – адаптивность к искажениям, например перспективным;
- инициирует ложные срабатывания на фоновых фрагментах, схожих по начертанию с символами, – необходимость постобработки решений.

3. Алгоритм мультимасштабного фрагментирования изображения. Нейросетевая модель классифицирует текстовые и фоновые образы, ограниченные изображением 32×32 пикселя, однако текстовые объекты изображений реальных сцен могут иметь произвольные размеры и пространственное положение. В связи с этим применение модели основано на *алгоритме «мультимасштабного фрагментирования изображения»*:

1) приведем изображение к фиксированному размеру ($H \times W$);
2) зададим диапазон изменения масштаба для локализации текстовых образов предполагаемых размеров, например масштабы от 1:1,5 до 1,3:1;
3) выберем общую величину h шага фрагментирования вдоль координатных осей;
4) дополняем изображение в цикле по изменению масштаба рамкой толщиной в 16 пикселей для выделения текста вблизи границ;
5) перемещаем локальное окно по изображению в порядке слева направо и сверху вниз, чтобы центр (x, y) текущего окна не попадал в рамку;
6) выделяем фрагмент изображения с координатами $(x-15, y-15)$ и $(x+16, y+16)$ левого верхнего и правого нижнего углов, нормируем его и подаем на вход детектора;
7) сохраняем отклик в каждых позиции и масштабе.

Недостаток алгоритма – значительные вычислительные затраты ввиду многочисленных вызовов нейросети для фрагментов. Возможны следующие варианты сокращения их числа:

- учет вероятных размеров текстовых образов, уменьшающий количество рассматриваемых масштабов (например, при фиксированном расстоянии до объектов);
- учет вероятного положения текста и, как следствие, сокращение области поиска (для изображений со схожей композицией, например дорожных сцен);
- увеличение шагов фрагментирования вдоль осей, что приводит к уменьшению количества вызовов нейросетевого детектора, но возможному пропуску искомых объектов;
- учет дополнительных характеристик текстовых объектов (например, исключение из обработки фрагментов с низкой плотностью контуров).

Указанные действия сокращают ресурсоемкость алгоритма на основе априорной информации о структуре и свойствах сцены, которая, однако, не всегда доступна, поэтому поиск путей оптимизации алгоритма основывался на анализе особенностей архитектуры нейросети. В результате установлено, что при высокой плотности пересечения фрагментов изображения (при $h < 32$), целесообразно вычислять отклики детектора одновременно во всех позициях локального окна, т. е. можно рассчитать отклики текущего слоя нейросети, сгруппировать их с учетом его структуры и передать на вход следующего слоя.

Преимущество данного подхода можно оценить на примере вычисления откликов нейронов первого слоя *C1*, содержащего 12 карт по 28×28 нейрона каждая, использующих по одному фильтру размера 5×5. Рассмотрим четыре локальных окна размером 32×32 пикселя с центром в точках (x, y), $(x, y+1)$, $(x+1, y)$ и $(x+1, y+1)$ (рис. 2а), для которых необходимо выполнить свертку путем вычисления сумм поэлементных произведений фильтров и каждого подокна размера 5×5 в 28·28 позициях. Свертки для окон отличаются лишь в крайних позициях (рис. 2д), поэтому более 90 % расчетов для первого окна соответствует и остальным. Следовательно, чтобы вычислить отклики *C1* во всех локальных окнах нужно один раз выполнить свертку изображения с 12 фильтрами и объединить результаты с учетом структуры слоя (аналогично рассчитываются отклики остальных). С учетом этого разработан описанный ниже *алгоритм модифицированного фрагментирования изображения*.

Вход: изображение *I* размера $H \times W$; нейросетевой детектор, задаваемый множеством слоев $L = \{L_1, \ldots, L_6 \mid L_i = (W_i, s_i, V_i, f)\}$, где $W_i{}^j$ – фильтры размера $m_i \times n_i$ ($m_i = 2a_i + 1$, $n_i = 2b_i + 1$, (a_i, b_i) – центр фильтра) *j*-й карты *i*-го слоя, а s_i^j – смещение нейронов; V_i^j – номера карт $(i-1)$-го слоя, связанных с *j*-й картой *i*-го; f_i – функция активации.

Выход: отклики $M_i^j(x, y)$ *j*-й карты *i*-го слоя, вычисляемые по формуле:

$$M_i^j(x+d_i, y+d_i) = f_i\left(s_i^j + \sum_{l \in V_i^j} \sum_{p=-a_i, h_i}^{a_i} \sum_{t=-b_i, h_i}^{b_i} w_i^l(p+a_i, t+b_i) \times M_{i-1}^l(x+p+d_i, y+t+d_i)\right), \quad (2)$$

где h_i, d_i – шаги разреженной свертки, зависящие от структуры вычислений в *i*-м слое; w_i^l – фильтр *j*-й карты *i*-го слоя, связывающий ее с *l*-й картой $(i-1)$-го слоя.

Этапы реализации алгоритма при фрагментировании с $h = 1$:

1) вычисление M_1^j ($j = \overline{1,12}$), связанных с *I* фильтрами размера 5×5, $h_1 = 1$;

2) вычисление M_2^j ($j=\overline{1,12}$), связанных с M_1^j фильтрами размера 2×2, $h_2=1$;

3) вычисление M_3^j ($j=\overline{1,14}$), связанных с группами M_2^j фильтрами размера 9×9, $h_3=2$;

4) вычисление M_4^j ($j=\overline{1,14}$), связанных с M_3^j фильтрами размера 3×3, $h_4=2$;

5) вычисление M_5^j ($j=\overline{1,14}$), связанных с M_4^j фильтрами размера 17×17, $h_5=4$;

6) вычисление M_6^j ($j=\overline{1,2}$), связанных с каждой M_5^j фильтрами размера 1×1, $h_6=1$.

Так как свертка в слоях $S2$ и $S4$ проводится по непересекающимся окрестностям предыдущих, при расчете M_3^j, M_4^j и M_5^j ее шаг отличен от единицы (рис. 2б, в, г), а $d_{1-6}=1$.

Анализ структуры вычислений при фрагментировании с h = 2 показал, что при кратности h размеру фильтров подвыборочных слоев (т. е. при четности h для данной модели) можно добиться значительного сокращения расчетов из-за общности результатов сверток для соседних локальных окон (рис. 2е, ж, з). Обобщив данное наблюдение для произвольного четного шага фрагментирования, можно определить величины соответствующих шагов сверток по слоям детектора (аналогично для моделей СНС с такими же типами слоев). Эксперименты показали, что модифицированное фрагментирование более чем на два порядка сокращает ресурсоемкость применения детектора для стандартного ЭВМ: Intel Core 2 Duo i3-530 2,93 GHz, DDR3 2048 Mb. Проход в одном масштабе с h = 2 выполняется в среднем в 2,8 раза быстрее, чем с h = 1 (в 2,2 раза, чем с h = 3) ввиду уменьшения числа позиций локального окна и сокращения избыточных расчетов. Наиболее эффективным по балансу ресурсоемкости и точности является фрагментирование с h = 4 (см. рис. 3).

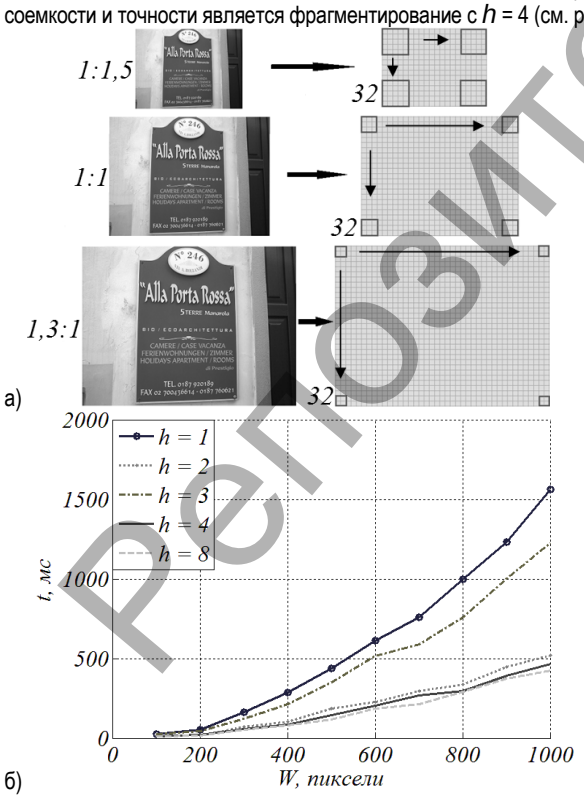


Рисунок 3 – Ресурсоемкость алгоритма модифицированного фрагментирования (а) в одном масштабе изображения $H\times W$ ($H/W=1,2$) (б) с разными значениями шага h

Следует отметить, что в литературе описаны способы оптимизации СНС, аналогичные созданной модификации фрагментирования [9, 10]. Однако их общим недостатком является недостаточная формализация расчетных процедур и отсутствие учета кратности шага фрагментирования размерам фильтров СНС.

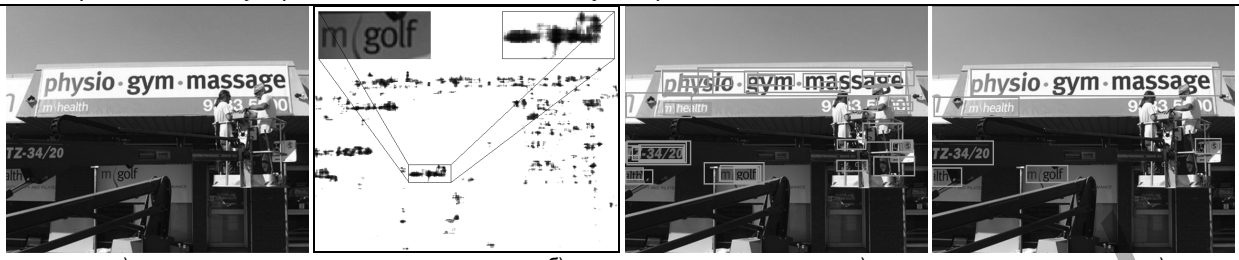
Оптимизация на основе априорной информации. Один из вариантов сокращения объема вычислений при фрагментировании основан на учете дополнительных признаков текста. В частности, известно, что для улучшения читабельности текстовые объекты зачастую размещаются на контрастном фоне, следовательно, их контурные признаки должны быть хорошо различимыми. Кроме того, фиксированная структура некоторых типов документов (например, паспортов) предусматривает неравномерное распределение текста (наличие широких пробельных интервалов между полями). Учет подобных свойств уменьшает анализируемую площадь изображения – сокращается объем требуемых расчетов, для чего предлагается выполнить следующие действия:

- вычислить для входного изображения карту значений некоторого простого и характерного признака текста (например, модуля градиента);
- построить для признаковой карты «интегральное изображение», которое позволяет быстро определить сумму значений признака в любой окрестности;
- если среднее значение признака в окрестности текущего пикселя меньше установленного порога – исключить пиксель из дальнейшего рассмотрения;
- объединить оставшиеся пиксели в кластеры по близости их пространственного положения (например, маркировкой связных бинарных сегментов);
- выполнить модифицированное фрагментирование отдельно для частей изображения, ограниченных минимальными прямоугольниками кластеров, и объединить отклики.

Эффективность данной оптимизации зависит от низкой вычислительной сложности создания кластеров и их достаточной изолированности (высокой разреженность покрытия изображения прямоугольниками). Так, для выборки изображений 31-й страницы паспорта белорусского образца с использованием бинарного представления в качестве признака среднее снижение ресурсоемкости составляет более 25%.

4. Алгоритм локализации текстовых блоков по откликам детектора. Отклики последнего слоя $M(x,y)\in\{(t_1,t_2)\}$ являются результатом прохода детектора по изображению в одном масштабе. Если для некоторого отклика верно неравенство $t_2>t_1$, то справедливо утверждение: окрестность изображения с координатами $(x-15,y-15)$ левого верхнего и $(x+16,y+16)$ правого нижнего угла содержит текстовый образ. Функцией активации нейронов является гиперболический тангенс, а обучение проводится так, что $t_1\to 1(-1)$ и $t_2\to -1(1)$ при подаче на вход фонового (текстового) образа. Однако эксперименты показали, что выходные значения t_i распределяются в диапазоне $[-1,1]$ произвольно, что обусловило рассмотрение матрицы уверенности детектора $U(x,y)=\max\{1-(t_1+1)/(t_2+1),\,0\}$ с нормированными к $[0,1]$ элементами (см. рис. 4а, б).

На рисунке тоновой визуализацией показано, что текстовым образам соответствуют высокие значения уверенности (чем темнее, тем выше), однако детектор может иметь уверенный отклик и для фрагмента с нецентрированными символами (их частями), а также для фоновых с текстурой близкой текстовым. Поэтому матрицу уверенности в первую очередь можно использовать для восстановления «поблочного» пространственного распределения текстовой информации с учетом ее структурных характеристик.



а) б) в) г)

Рисунок 4 – Локализация текстовых блоков по откликам детектора: изображение реальной сцены (а), матрица уверенности детектора для него в одном масштабе (б), первичные текстовые блоки различных масштабов изображения (в) и итоговые блоки (г)

Основные этапы алгоритма локализации:

- вычисление матриц уверенности детектора для выбранных масштабов изображения;
- образование множества «первичных» текстовых блоков отдельных масштабов;
- формирование итоговых блоков объединением первичных со схожими свойствами.

Первый этап реализуется с помощью описанного выше алгоритма мультимасштабного фрагментирования изображения для дискретного набора масштабов.

Образование первичных текстовых блоков. Учитывая, что детектор обучается на изображениях размером 32×32 пикселя, содержащих текстовый образ (его центр будем называть истинным прообразом), его высота не может превышать 32 пикселя. Это позволяет ввести параметры $max\_dy = 32$ и $min\_dy = max\_dy/3 \approx 10$ – максимальное и минимальное расстояния в пикселях по горизонтали между прообразами (символами одного слова). Также зададим минимальный порог уверенности для прообраза, например $min\_U = 0,8$.

Поиск истинных прообразов проведем с помощью немаксимального подавления:

- для текущей строки сформируем список прообразов с уверенностью выше min_U;
- обойдем список в порядке убывания уверенности элементов и исключим из рассмотрения текущий прообраз, если в списке есть более уверенный и близкий к нему (расстояние по горизонтали не превышает min_dy), или же такой элемент находился в списке, и его уверенность превышала текущую не менее чем на $dU\_one\_max = min\_U/10 = 0,08$.

В результате в списке останутся элементы, являющиеся локальными максимумами текущей строки матрицы уверенности. Параметр $dU\_one\_\max$ позволяет корректно обработать ситуацию, когда один прообраз мог удалить остальные (относящиеся к одному символу), а сам затем стать исключенным некоторым прообразом соседнего символа. Используя список прообразов, можно сформировать множество первичных текстовых блоков:

- обойдем список в порядке убывания уверенности и первый из свободных элементов (не отнесенный на текущий момент ни к одному текстовому блоку) образует новый блок;
- просмотрим список с начала, пока не встретим прообраз, уверенность которого больше средней уверенности текущего блока или меньше ее не более чем на величину $dU\_two\_max = 2dU\_one\_max = 0,16$, и при этом отдаленный не более чем на $\max\_dy$ относительно любого элемента блока;
- найденный элемент добавим к блоку, пересчитаем его среднюю уверенность и повторим предыдущее действие, в противном случае сохраним характеристики блока: минимальный прямоугольник (вычисленный с помощью координат фрагментов изображения с центрами в прообразах блока), текущий масштаб изображения, среднюю уверенность, а также данные о расположении и уверенности элементов.

Поиск первичных блоков проведем для каждой строки изображения в каждом из его масштабов, выбранных на первом этапе. В результате получим начальную информацию о распределении текстовых данных (см. рис. 4в).

Объединение первичных блоков в итоговые. Построенные блоки могут относиться к одному текстовому объекту, т. к. их формирование проходило в близких строках изображения. Поэтому необходимо выполнить объединение первичных блоков одного масштаба:

1) обрабатываем блоки в порядке убывания средней уверенности элементов – до тех пор, пока существует пара блоков с достаточной площадью пересечения минимальных прямоугольников (более 50 % от площади любого из них);

2) выполним посимвольное слияние пары на базе более уверенного (первого) блока, обходя элементы второго в порядке убывания уверенности:

2.1) сравним координаты и уверенность текущего прообраза $\left(x_i^2, y_i^2\right)$ второго блока со средней координатой x_{aver}^1 и уверенностью U_{aver}^1 прообразов первого, если: $\left|x_{aver}^1 - x_i^2\right| > max\_dx$ (где $max\_dx = 10$ – максимальное расстояние по вертикали между прообразами) и/или $U_{aver}^1 - U\left(x_i^2, y_i^2\right) > dU\_two\_max$ – перейдем к следующему прообразу;

2.2) если для некоторого прообраза $\left(x_k^1, y_k^1\right)$ первого блока справедливо $\left|y_k^1 - y_i^2\right| \le \min\_dy$, то будим рассматривать эти прообразы как прообразы одного символа, при этом, если $\left|U\left(x_k^1, y_k^1\right) - U\left(x_i^2, y_i^2\right)\right| < dU\_one\_max$ – усредним их координаты, иначе выберем характеристики прообраза с большей уверенностью и перейдем к пп. 2.4;

2.3) иначе добавим к первому блоку текущий прообраз второго блока;

2.4) если изменились характеристики любого элемента первого блока – выполним проверку аналогичную пп. 2.1, 2.2 для всех его прообразов;

3) пересчитаем характеристики первого блока и изменим очередность его обработки в соответствии с новым значением средней уверенности.

Символы слова могут иметь различную высоту (например, из-за искажений), поэтому блоки близких масштабов также могут относиться к одному и тому же текстовому объекту. Справедливость данного замечания подтверждается представленными на рис. 4в блоками, сформированными в различных масштабах изображения. Исходя из этого, на заключительном этапе алгоритма выполним следующие шаги:

- приведем координаты блоков и их прообразов к фиксированному масштабу;

- объединим блоки с помощью описанной выше процедуры, потребовав выполнения условия: масштаб каждого прообраза блока должен отличаться от его среднего значения не более чем на $\max\_dS = 0,3$;
- отбросим блоки, уверенность которых меньше максимальной более чем на величину $dU\_one\_\max$ (итоговые блоки представлены на рис. 4г).

Качественное сравнение алгоритма с аналогичными [7, 8] показывает, что он обладает большей универсальностью, поскольку:

- формирование блоков проводится в ходе анализа символов-кандидатов с учетом их индивидуальных характеристик, а не простым выбором наиболее уверенного прообраза, что позволяет не предъявлять завышенных требований к точности детектора и использовать в его основе значительно менее громоздкую нейросетевую архитектуру;
- допускается локализация наклоненных текстовых объектов, при этом допустимый уровень наклона регулируется параметром max_dx;
- объединение результатов обработки в разных строках и масштабах позволяет формировать блоки с контролируемым отличием размера символов, что необходимо учитывать ввиду возможного различия регистра символов и их стилистического оформления.

5. Модуль детектирования текста. Представленные модель и алгоритмы являются основой модуля детектирования текста, разработанного на языке программирования C#:

1) на вход подается изображение, а также предполагаемые диапазон высоты символов и координаты содержащих их областей изображения (опциональные параметры);

2) выполняется его предобработка (удаление локального шума и контрастирование);

3) проводится мультимасштабное фрагментирование изображения: при указании предельных высот символов начальный и конечный масштаб выбирается так, чтобы их высота находилась в диапазоне 16…24 пикселя, иначе используются значения по умолчанию;

4) локализуются текстовые блоки по вычисленным откликам детектора;

5) возвращаются выходные характеристики текстовых блоков: масштаб, координаты объемлющих прямоугольников и средняя уверенность детектора.

При реализации модуля задействованы только возможности C#, зачастую не имеющие реализаций в виде готовых классов. В частности, основным средством работы с матрицами являлись указатели, функционирующие при ограниченном контроле CLR, и распараллеленные циклы, существенно повысившие производительность. Кроме того, обнаружена низкая скорость расчета гиперболического тангенса (метода Math.Tanh), что подтверждает ранее полученный вывод: "экспонента или гиперболический тангенс в их стандартных для сред программирования реализациях не адаптированы для использования в программах моделирования нейронных сетей … реализации этих функций утяжелены проверками" [9].

Рассмотрены альтернативные способы вычисления гиперболического тангенса:

1) использование стандартной формулы $\tanh(x) = \left(e^{2x} - 1\right) / \left(e^{2x} + 1\right)$ с вызовом Math.Exp;

2) применение формулы из п. 1 и близкой к точной аппроксимации экспоненты с помощью ее разложения в ряд Тейлора (до девятого члена включительно);

3) аппроксимирование функции [5]:

$$\tanh(x) = \mathrm{sgn}(x)\left(1 - 1/\left(1 + |x| + x^2 + 1{,}41645 \cdot x^4\right)\right).$$

Первый способ в среднем в 2,5 раза ускоряет расчет с погрешностью $\varepsilon \approx 0$, второй – в 4 раза с $\varepsilon \approx 0{,}0004$, третий – в 7,5 раз с $\varepsilon \approx 0{,}0143$. Отметим, что наличие погрешности требует применения соответствующих расчетов как при обучении, так и применении нейросети.

Тестирование модуля. Тестирование выполнялось на выборке из 100 изображений, отражающих момент въезда автотранспорта на охраняемую территорию. Ракурс съемки обеспечивал горизонтальную (с максимальным отклонением ±10º) ориентацию пластины регистрационного номера, который требовалось локализовать с помощью модуля. Учитывая размер изображений в 704×576 пикселей и расстояние от видеокамеры до объекта, обработка выполнялась в диапазоне масштабов увеличения $[1, 2]$ с шагом 1/3, при этом шаг фрагментирования равнялся четырем (средняя продолжительность обработки – 1,5 с на ЭВМ стандартной конфигурации, указанной выше).

Качество локализации оценивалось по характеристикам соответствия двух множеств прямоугольников (G – определенных вручную истинных прямоугольников, объемлющих текстовые образы, D – сформированных модулем), рассчитываемых по формулам:

$$recall = \sum_{i=1}^{|G|} \max_j \left(S_{G_i \cap D_j} / S_{G_i}\right) \Big/ |G|,$$

$$precision = \sum_{j=1}^{|D|} \max_i \left(S_{D_j \cap G_i} / S_{D_j}\right) \Big/ |D|. \qquad (3)$$

Значение $recall$ отражает относительную величину покрытия истинных прямоугольников сформированными, а $precision$ – точность покрытия. Полученное $recall$ = 0,96 показало, что локализованы практически все символы (см. рис. 5а, б), лишь для нескольких изображений из-за высокого размытия номера выполнена частичная локализация (см. рис. 5в). Снижение точности ($precision$ = 0,65) в основном вызвано увеличением размера сформированных прямоугольников в сравнении с истинными (см. рис. 5г). Причиной является недостаточное количество масштабов изображения, не позволившее рассмотреть все текстовые образы в оптимальном размере (20 ± 4 пикселя) для их детектирования. Однако данный недостаток лишь частично усложняет дальнейшие этапы анализа (сегментацию, распознавание и т. п.), поэтому в целом поставленная задача была успешно решена.

При количественном сравнении с модулем, основанным на каскаде Хаара [12], обученным на изображениях номерных пластин, получена следующая оценка качества локализации (при оптимальных зна-

а) б) в) г)

Рисунок 5 – Примеры результатов работы модуля детектирования текста: точной (а) (*recall* = 0,92, *precision* = 0,96), частичной (б) (*recall* = 0,36, *precision* = 0,86), избыточной (в) (*recall* = 1,00, *precision* = 0,55) локализации номеров, а также слов базы ICDAR 2013 (г)

чениях параметров): *recall* = 0,84 и *precision* = 0,46. Это показало бóльшую эффективность разработанного модуля, при этом в отличие от аналогичных нейросетевых [6], он является более универсальным и может использоваться для локализации произвольных текстовых объектов (строк, слов и др.). Также получена оценка в рамках решения задачи локализации слов 233 тестовых изображений базы ICDAR 2013 (см. рис. 5г), которая выражалась тремя параметрами: $rec = N/K$, $prec = N/P$ (где N – число верно локализованных слов, K – общее число слов, P – общее число локализованных слов) и $F - score = 2rec \cdot prec/(rec + prec)$, при расчете которых применяется специализированная система штрафов в ситуациях соответствия «один ко многим» и «многие ко многим». Оценка модуля составила: *rec* = 78,71, *prec* = 71,03, *F–score* = 74,67, коммерческого аналога ABBY OCR SDK v10 – 35,07, 60,95 и 44,52, а нейросети со значительно более громоздкой архитектурой – 89,53, 94,26 и 91,84 соответственно [13]. Данная оценка показывает работоспособность предложенной модели детектора в условиях сложной композиции изображений и высокой стилистической вариативности текста. Возможности повышения точности локализации связаны с увеличением количества рассматриваемых масштабов изображения и совершенствованием процедуры сегментации блоков на слова.

Заключение. В исследовании показана перспективность применения компактных СНС для решения практических задач обработки текста на изображениях. Преимуществом компактных СНС, по сравнению с ГНС, является достаточно высокая обобщающая способность в сочетании с возможностью их реализации на неспециализированных языках программирования, обучения и применения на стандартном оборудовании. Предложенные алгоритмы применения СНС универсальны относительно архитектур со сверточными и подвыборочными слоями и могут использоваться для поиска нетекстовых объектов. Актуальными задачами являются совершенствование предложенной модели детектора с целью снижения количества ложных срабатываний (принятий фонового фрагмента за текстовый) и дальнейшая вычислительная оптимизация модели на базе векторизации.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Кузьмицкий, Н.Н. Построение целостных контуров объектов на полутоновых изображениях / Н.Н. Кузьмицкий, С.С. Дереченник // Информационные технологии и системы: материалы Международной научной конференции. – 2011. – С. 175–176.
2. Krizhevsky, A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. Hinton // Proceedings of the International Conference on Neural Information Processing Systems. – 2012. –. Vol. 1. – P. 1097–1105.
3. LeCun, Y. Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86. – P. 2278–2324.
4. Кузьмицкий, Н.Н. Построение универсальных классификаторов текстовых образов русского языка на базе сверточных нейросетей / Н.Н. Кузьмицкий // Доклады БГУИР. – 2015. – № 4. – С. 33–39.
5. Калиновский, И.А. Обзор и тестирование детекторов фронтальных лиц / И.А. Калиновский, В.Г. Спицын // Компьютерная оптика. – 2016. – Т. 40, № 1. – С. 99–111.
6. Druki, A.A. Application of Convolutional Neural Networks for Automatic Number Plate Recognition on Complex Background Images / A.A. Druki, J.A. Bolotova, V.G. Spitsyn // Applied Mechanics and Materials. – 2015. – Vol. 756. – P. 695–703.
7. Wang, K. End-to-end scene text recognition / K. Wang, B. Babenko, S. Belongie // Proceedings of the IEEE International Conference on Computer Vision. – 2011. – Vol. 6. – P. 1457–1464.
8. Delakis, M. Text detection with convolutional neural networks / M. Delakis, C. Garcia // Proceedings of the International Conference on Computer Vision Theory and Applications. – 2008. – Vol. 2. – P. 290–294.
9. Sermanet, P. OverFeat: Integrated recognition, localization and detection using convolutional networks / P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun [Электронный ресурс]. – Режим доступа : http://arxiv.org/abs/1312.6229.pdf. – Дата доступа: 01.08.2017.
10. Garcia, C. Convolutional face finder: A neural architecture for fast and robust face detection / C. Garcia, M. Delakis // Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26. – P. 1408–1423.
11. NeuroPro нейронные сети, методы обработки и анализа данных: от исследований до разработок и внедрений [Электронный ресурс]. – Режим доступа : http://neuropro.ru/memo312.shtml. – Дата доступа : 06.08.2017.
12. Open Source Computer Vision Library [Электронный ресурс]. – Режим доступа : https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_russian_plate_number.xml. – Дата доступа : 07.08.2017.
13. ICDAR 2013 Robust Reading Competition [Электронный ресурс]. – Режим доступа : http://rrc.cvc.uab.es/?ch=2&com=evaluation. – Дата доступа : 08.08.2017.

Материал поступил в редакцию 24.10.2017

KUZMITSKY N.N. Detection of text objects based on the «not-deep» convolutional neural network with optimization of calculations

Paper presents the model of text detector in form of «not-deep» convolutional neural network and the method of its application based on modified multiscale fragmentation of image, which reduces resource intensity of processing by more than two orders in comparison with standard fragmentation. The algorithm for text localization based on responses of detector is developed, which adaptability exceeds similar ones due to joint analysis of responses in adjacent lines and close scales of image, which allows localizing distorted text blocks of different sizes and orientations.

Based on the neural network model, the module for text detection is created, applicable for processing images with an arbitrary composition. Taking into account the priori information and features of the chosen software platform, ways of reducing resource intensity of the module are determined. Testing the module on sample of images reflecting moment of vehicles entry to protected area demonstrated high quality of registration numbers text localization, which exceeds level of the specialized module based on Haar cascade.

УДК 004.81

Крапивин Ю.Б.

ЛИНГВИСТИЧЕСКИЙ АНАЛИЗ ТЕКСТА В ЗАДАЧЕ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ЗАИМСТВОВАННЫХ ФРАГМЕНТОВ ТЕКСТОВЫХ ДОКУМЕНТОВ

Введение. Постоянно увеличивающийся объем информации, представленной на различных языках как в полнотекстовых базах данных, так и в сети Интернет, обостряет проблему ее оперативной и качественной обработки с целью удовлетворения информационной потребности пользователей. Под информационным поиском (ИП) обычно понимают непосредственно процесс поиска и пред-

Крапивин Юрий Борисович, *старший преподаватель кафедры интеллектуальных информационных технологий Брестского государственного технического университета.*
Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.