УДК 004.932.1

Н.Н. Кузьмицкий

ОБНАРУЖЕНИЕ ФРАГМЕНТОВ ТЕКСТА НА ИЗОБРАЖЕНИЯХ РЕАЛЬНЫХ СЦЕН НА БАЗЕ СВЕРТОЧНОЙ НЕЙРОСЕТЕВОЙ МОДЕЛИ

Рассматривается модель детектора текстовых образов на базе сверточной нейронной сети, способной синтезировать высокоуровневые признаки образов в режиме «черного ящика». Описывается методика применения детектора, основанная на алгоритмах мультимасштабного сканирования и локальной интерпретации откликов, позволяющая обнаруживать текстовые объекты на изображениях реальных сцен. Показываются преимущества разработок в сравнении с аналогами, выполняется оценка эффективности на примере известной базы данных.

Введение

Обнаружение текстовых объектов на изображениях реальных сцен является весьма распространенной задачей в практических приложениях: при поиске изображений по содержанию в больших коллекциях и глобальной сети, идентификации объектов в промышленности, помощи людям с ограниченными возможностями по зрению (просмотре цены товара), навигации в городе по информационным меткам на иностранном языке и др. [1]. Многие из предположений традиционных систем оптического распознавания (optical character recognition, OCR) относительно характеристик изображений в случае реальных сцен являются несостоятельными (например, черный текст на ярком фоне), при этом в ходе анализа приходится сталкиваться со следующими проблемами (рис. 1):

– текстовые образы могут иметь различную яркость и размер (в пределах одного слова), располагаться в произвольном месте сцены, содержащей текстуры и фоновые объекты;

– в то время как в документах используются известные машинописные шрифты, текст реальных сцен может содержать существенно стилизованные и искаженные образы, необходимые, например, для узнаваемости фирменного бренда;

– традиционные OCR-системы предназначены для обработки высококачественных изображений, полученных с помощью сканера, однако изображения реальных сцен могут быть синтезированы в условиях недостаточного освещения, различного ракурса, переносным устройством с низкими оптическими характеристиками камеры и т. п.

Рис. 1. Примеры изображений текстовых образов реальных сцен

Указанные особенности приводят к необходимости разработки новых подходов, лишенных таких недостатков OCR, как «ручной» выбор признаков, сложность оптимального подбора параметров, низкая универсальность и др. Перспективным направлением исследований является использование для решения поисковых задач методов машинного обучения.

1. Модель сверточного нейросетевого детектора текстовых образов

Сверточные нейронные сети (convolutional neural networks, CNN) – это многослойные иерархические модели, поводом к созданию которых послужили исследования зрительного аппарата кошек, проведенные в 1960-х гг. Их результатом стало открытие двух типов клеток, влияющих на зрительную восприимчивость: первые обладают свойством локальной чувстви-

тельности и предназначены для выделения элементарных признаков образов (например, ориентированных краев), вторые путем их комбинирования формируют высокоуровневые признаки.

Известны различные реализации сверточных нейронных сетей, отличающиеся топологией слоев, организацией процесса обучения и др. Исходя из результатов их применения при решении задач анализа изображений, а также возможности обучения без использования специализированного аппаратного обеспечения, в качестве базового для разработки детектора текстовых образов был выбран тип нейросетей, который разработал Y. LeCun в конце 1990-х гг. [2]. В их основе лежат три архитектурные идеи:

1) *локальные рецептивные поля* (нейроны получают сигнал от окрестностей нейронов предыдущего слоя, за счет чего сеть обучается двухмерной структуре входного образа);

2) *разделяемые веса* (нейроны слоя объединены картами, в которых они обладают общими весами, при этом карты формируют различные признаки и сокращают количество параметров, настраиваемых в ходе обучения);

3) *пространственные подвыборки* (локальное усреднение карт приводит к синтезу высокоуровневых признаков, повышая устойчивость к искажениям).

Обучение сверточной нейронной сети осуществляется модификацией алгоритма обратного распространения ошибки на основе метода Левенберга – Марквардта, обеспечивающей рост сходимости из-за индивидуальной настройки для весов каждого нейрона параметра η, что позволяет замедлять процесс на крутых областях весового пространства и ускорять на плоских [3]. Таким образом, сверточные нейронные сети служат эффективным средством решения задач обработки изображений. Их преимуществом является объединение в рамках одной модели экстрактора признаков и классификатора. При этом высокоуровневые признаки синтезируются в режиме «черного ящика» путем чередования процедур свертки с настраиваемыми фильтрами и подвыборки полученных откликов, а обучение классификатора полностью контролируется.

Для обнаружения текстовых образов на изображениях была разработана модель детектора в виде сверточной нейронной сети (рис. 2). Ее входной слой содержит 32×32 нейрона, которые получают сигнал в виде яркости полутонового изображения аналогичного размера. Первый из четырех скрытых слоев (C1) является сверточным с двенадцатью картами, содержащими по 28×28 нейронов, которые разделяют по одному фильтру размером 5×5 и параметру смещения (244 608 связей, 312 настраиваемых параметра). За ним следует подвыборочный слой (S2), усредняющий отклики нейронов предыдущего слоя по неперекрывающимся окрестностям размером 2×2, поэтому он содержит 12 карт по 14×14 нейронов, разделяющих по одному параметру весового усреднения и смещения (11 760 связей, 24 параметра).

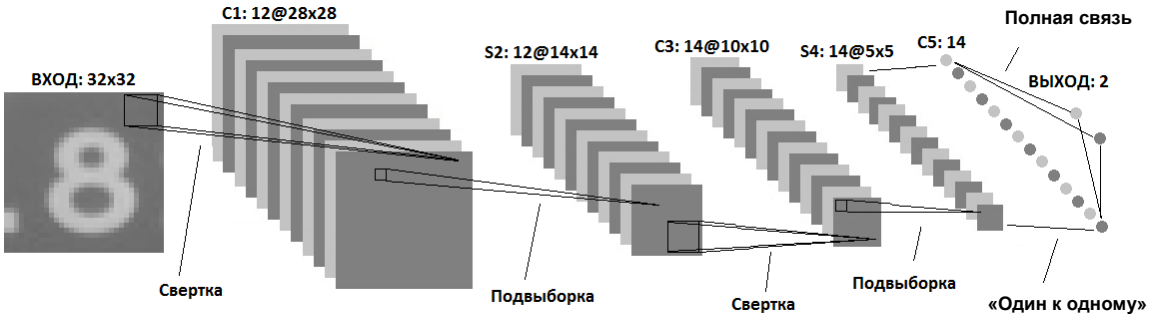


Рис. 2. Архитектура сверточного нейросетевого детектора

Следующий слой (C3) – сверточный, содержит 14 карт размером 10×10 нейронов, каждая соединена лишь с подмножеством карт слоя S2: первые восемь карт слоя C3 соединены с тремя из S2 (по одной из каждой четверки), остальные шесть связаны с половиной карт предыдущего. Для каждой пары настраивается свой фильтр размером 5×5 и один параметр смещения (151 400 связей, 1514 параметров). Целью разреженного соединения является ликвидация симметрии в топологии сети, что заставляет карты формировать различные признаки, так как они получают разный входной сигнал. Второй подвыборочный слой (S4) аналогичен S2 и уменьшает размер 14 карт до 5×5. Следующий слой (C5) содержит 14 нейронов, соединенных с одной

картой предыдущего слоя фильтром размером 5×5 и одним параметром смещения (364 связи, 364 параметра). Выходной слой представлен двумя нейронами, соединенными с каждым нейроном из C5 с помощью одного параметра веса и смещения (30 связей, 30 параметров). Всего представленная архитектура содержит 409 912 связей и 2272 настраиваемых параметра.

Первые четыре слоя сети являются экстрактором высокоуровневых признаков, в то время как последние два – классификатором в форме многослойного персептрона [4]. Выбор размера входного изображения и числа карт в слоях объясняется стремлением к достижению баланса между высокими обобщающими свойствами сети и эффективностью ее практического применения. Масштаб фильтров и окрестностей усреднения определяется так, чтобы в ходе послойного уменьшения размера карт количество нейронов в первом классифицирующем слое (C5) равнялось числу карт. В качестве функции активации нейронов был выбран гиперболический тангенс ввиду наличия горизонтальных асимптот, симметричности, простого вычисления производной и распространенности при решении задач, аналогичных рассматриваемой [2, 6].

Обучение детектора выполняется с использованием базы маркированных образов, разделенной на тренировочную и тестовую части. База должна содержать множество пар вида (x, y), где x – графический образ, y – номер класса (0 – текстовый, 1 – фоновый). Текстовым образом является полутоновое изображение масштаба 32×32 пиксела, в которое вписан символ с сохранением пропорций размеров. Выбор яркостных, текстурных и других признаков символов связан со свойствами сцен, на изображениях которых планируется выполнять детектирование. При этом эффективность обучения напрямую зависит от разнообразия образов, число которых определяется неравенством Вапника – Червоненкиса: обобщаемость сети прямо пропорциональна отношению объема тренировочной выборки к мере сложности модели (количеству параметров) [5].

Подготовка процесса обучения нейронной сети включает:

– присваивание весам случайных значений, равномерно распределенных на интервале $[-2{,}4; 2{,}4\ F_i]$, где F_i – число входных связей i-го нейрона;

– выбор количества эпох и методики изменения коэффициента η: обучение проводится за 34 эпохи с начальным значением $\eta = 0{,}000\ 85$, которое изменяется каждую эпоху путем умножения на коэффициент 0,85, в итоге конечное значение η составляет 0,000 004;

– настройку параметров искажений входных образов: величины поворота (например, в пределах ±5°), изменения масштаба (в пределах ±10 %).

Коррекция весов нейронов проводится после обработки каждого образа, при этом для ускорения обучения сети может применяться методика пропуска этапа обратного распространения ошибки в случае, если ее величина была меньше заданного значения ε. Также может быть выполнено дообучение нейросети, которое, по мнению ряда авторов, в частности С. Осовского, является весьма эффективным, так как позволяет выполнить «встряхивание весов» с минимальной вероятностью вывода поиска из сферы притяжения ранее найденного локального минимума, в отличие от обучения «с чистого листа». Сеть в такой ситуации должна проявить способности к усвоению наиболее характерных признаков и после кратковременной «амнезии» быстро восстановиться, а затем в большинстве случаев улучшить свои показатели [5].

Сравнительный анализ представленной модели и аналогичных показал, что в отличие от предложенной в [6] она обладает большей универсальностью, так как позволяет выполнять посимвольное, а не только построчное детектирование. При этом по сравнению с моделью, описанной в [7], данная модель не требует бесконтрольного обучения и содержит значительно меньше настраиваемых параметров (более чем в 40 раз), что повышает эффективность ее практического использования.

2. Алгоритм прохода детектора по изображению

Основной сферой применения детектора являются изображения реальных сцен, которые могут содержать произвольно распределенные текстовые объекты различного размера в отличие от объектов фиксированных размеров 32×32 пиксела для его входного слоя. С учетом указанных особенностей был разработан алгоритм применения детектора на основе мультимасштабного скользящего окна (рис. 3), состоящий из следующих шагов:

1) выполним масштабирование изображения к фиксированному размеру $H \times W$;

2) зададим диапазон изменения масштаба, например [–30, 30 %] с шагом 15 %;

3) выберем величину смещения скользящего окна по осям x (dx) и y (dy);

4) в цикле по каждому масштабу дополним изображение рамкой толщиной 16 пикселов с усредненной локальной яркостью края (для выделения текста, прилегающего к границам);

5) будем перемещать скользящее окно размером 32×32 пиксела слева направо и сверху вниз с шагами dx и dy так, чтобы центр окна (x, y) не попадал в рамку;

6) в текущей позиции окна выделим окрестность с координатами (x – 15, y – 15), (x + 16, y + 16) левого верхнего и правого нижнего угла, которую подадим на вход детектора;

7) сохраним отклики детектора в каждой позиции и каждом масштабе.

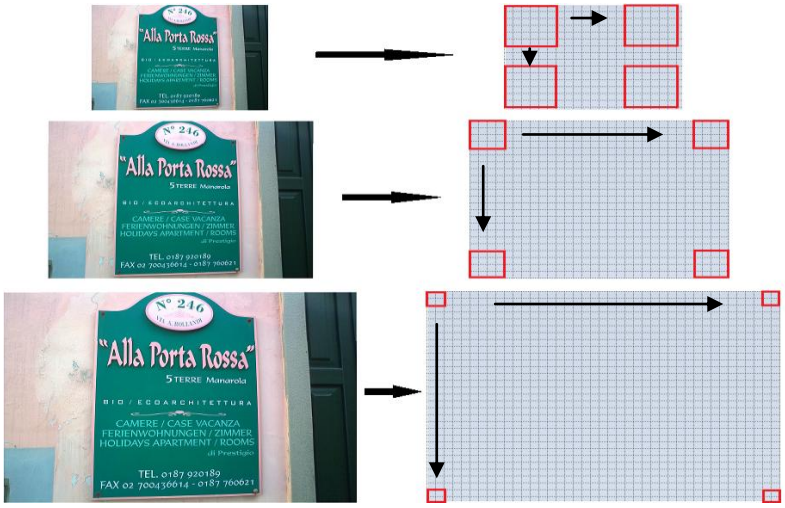


Рис. 3. Использование детектора в соответствии с алгоритмом мультимасштабного скользящего окна

Предложенный алгоритм позволяет обрабатывать текстовые образы различного размера, расположенные в произвольной части изображения, что значительно повышает универсальность детектора. Его недостатком являются значительные временные затраты ввиду необходимости выполнения многочисленных этапов прямой передачи сигнала в нейросети. Возможными путями их сокращения являются:

– использование информации о предполагаемых размерах текстовых образов, позволяющей уменьшить число масштабов (например, при обработке изображений сцен, полученных со стационарной камеры с фиксированными расстояниями до текстовых объектов);

– учет типичного пространственного распределения текстовых блоков (при анализе однотипных изображений с похожей композицией, например дорожных сцен);

– увеличение шага скользящего окна вдоль координатных осей, что приводит к уменьшению числа вызовов детектора;

– получение дополнительных данных о возможном размещении текстовой информации (например, исключение из обработки областей с недостаточной плотностью контуров).

Указанные направления оптимизации позволяют сократить время работы алгоритма на основе не всегда доступной априорной информации о структуре и свойствах сцены. Поэтому поиск путей дальнейшего усовершенствования был связан с анализом архитектуры сети и передачей сигналов в ней, который привел к следующему выводу: ввиду высокой вероятности пересечения окрестностей изображения, подаваемых на вход детектора в каждом проходе (при $dx < 32$ или $dy < 32$), целесообразным является вычисление его откликов не для отдельных, а для всех возможных окрестностей сразу. Другими словами, для нейронов каждого слоя можно последовательно сформировать входные сигналы и затем, группируя и передавая их, вычислить отклики детектора по всем позициям скользящего окна одновременно.

Преимущество данного подхода можно оценить на примере вычисления входных сигналов для нейронов сверточного слоя C1, содержащего 12 карт по 28×28 нейронов, которые разделяют по одному фильтру размером 5×5. Рассмотрим два локальных окна размером 32×32 пиксела с центрами в (x, y) и (x, y + 1), для которых необходимо выполнить свертку с не-

которым фильтром слоя. Пространственная фильтрация проводится путем вычисления сумм поэлементных произведений каждого подокна размером 5×5 с фильтром в 28×28 позиций с шагом 1. У двух рассматриваемых локальных окон результаты фильтрации будут отличаться лишь двумя крайними столбцами, поэтому более 90 % расчетов для первого окна могут быть использованы и для второго. Следовательно, для вычисления всех возможных входных сигналов для нейронов C1 нужно один раз выполнить свертку изображения с каждым из 12 фильтров и объединить результаты в соответствии со структурой слоя.

Аналогичным образом рассчитываются входные сигналы для остальных слоев, т. е. локальная интерпретация откликов детектора эффективно сочетается с глобальным характером вычислительной процедуры. Кроме того, расчеты можно сократить, учитывая величины смещений скользящего окна. Например, когда $dx = dy = 2$, вычисление входов для нейронов первого подвыборочного слоя S2 путем усреднения выходов C1 по окрестностям 2×2 может выполняться с шагом, равным 2, а не 1. Для сравнения, использование описанных оптимизаций при обработке изображения размером 640×480 пикселов позволяет сократить время расчетов более чем в 10 раз для оборудования следующей конфигурации: процессор Intel Core 2 Duo i3-530 2.93 GHz, ОЗУ DDR3 2048 Mb, ОС Windows 7 Ultimate, платформа .NET.

3. Алгоритм локализации текста на основе анализа откликов детектора

Результатом прохода нейросетевого детектора по изображению в одном масштабе является матрица откликов, каждый элемент которой представлен парой (t_1, t_2), где t_1 – выход фонового (первого) нейрона, t_2 – текстового. Если для соответствующего точке (x, y) элемента матрицы $t_2 > t_1$, 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 нейросетевого детектора является гиперболический тангенс, а обучение проводится так, чтобы $t_1 \to 1$ (–1), $t_2 \to -1$ (1) при подаче на вход фоновой (текстовой) области. Однако проведенные эксперименты показали, что для различных сигналов значения t_i могут быть произвольно распределены в диапазоне [–1, 1]. Поэтому для оценки решений детектора о наличии текстовых образов целесообразным является применение матрицы уверенностей (графическая интерпретация представлена на рис. 4, *а*), элементы которой в каждой точке (x, y) вычисляются по формуле

$$U(x, y) = max\ \{1 - (t_1 + 1) / (t_2 + 1), 0\}.$$

В основе предлагаемого алгоритма выделения текстовых областей лежит анализ пространственного распределения уверенностей откликов детектора. Как видно из рис. 4, *б*, окрестностям текстовых образов соответствуют высокие значения уверенностей (чем темнее, тем выше). При этом детектор может давать уверенный отклик и в окрестностях с неотцентрированным образом, а также на фоновых участках с текстурой, подобной текстовой. Таким образом, матрицу уверенностей преимущественно можно использовать для «поблочной» интерпретации распределения текстовой информации на изображении.

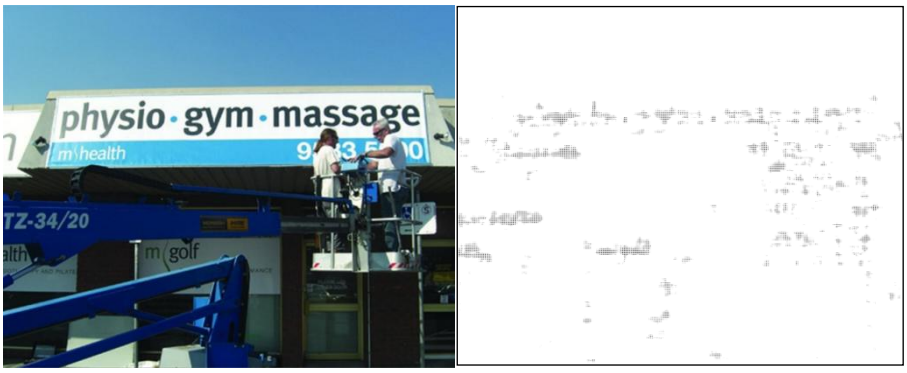
Основными этапами предлагаемого алгоритма локализации являются:

1) вычисление матриц уверенностей на разных масштабах изображения;
2) образование первичных текстовых блоков на отдельных масштабах;
3) формирование итоговых текстовых блоков путем объединения первичных.

Первый этап реализуется с помощью алгоритма мультимасштабного скользящего окна, описанного в разд. 2, для дискретного набора масштабов.

Образование первичных текстовых блоков. Детектор имеет уверенный отклик на фрагментах изображения (размером 32×32 пиксела), содержащих отцентрированный текстовый образ (центры таких фрагментов будем называть истинными прообразами). Следовательно, высота выделяемых символов не превышает 32 пиксела (точное значение определяется ее величиной у примеров выборки, используемой для обучения детектора), что позволяет ввести в рассмотрение следующие параметры: $max\_dy = 32$, $min\_dy = max\_dy / 3 \approx 10$ – максимальное и минималь-

ное расстояния по горизонтали в пикселах между символами. Кроме того, необходимо задать минимальный порог уверенности для истинного прообраза символа, например *min_U* = 0,8.



а) *б)*

в) *г)*

Рис. 4. Примеры изображений: *а)* сцены; *б)* матрицы уверенности детектора для изображения в одном масштабе; *в)* блоков различных масштабов изображения; *г)* итоговых текстовых блоков

Поиск истинных прообразов проведем с помощью *немаксимального подавления*:

– для текущей строки изображения сформируем список прообразов, уверенность которых больше порога *min_U*;

– обойдем список в порядке убывания уверенности элементов;

– отбросим 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, но при этом его уверенность была больше не менее чем на величину $dU\_one\_max = min\_U / 10 = 0{,}08$.

В результате в списке останутся элементы, являющиеся локальными максимумами текущей строки матрицы уверенностей. Применение параметра *dU_one_max* позволяет корректно обработать случай, когда один прообраз мог удалить остальные, относящиеся к тому же символу, а сам он мог быть исключен прообразом другого, близко расположенного к нему. Используя полученный список, можно сформировать первичные текстовые блоки следующим образом:

– обойдем список в порядке убывания уверенности элементов; первый из свободных элементов, не отнесенных на текущий момент ни к одному текстовому блоку, образует новый;

– просмотрим список с начала, пока не встретим прообраз, уверенность которого превышает среднюю уверенность блока или меньше ее не более чем на величину $dU\_two\_max = 2 \cdot dU\_one\_max = 0{,}16$ и отдаленный не более чем на *max_dy* относительно любого элемента блока;

– если элемент был найден, добавим его к блоку, пересчитаем среднюю уверенность и повторим действия предыдущего пункта, иначе сохраним характеристики текущего блока, включая минимальный прямоугольник, текущий масштаб изображения, среднюю уверенность, а также данные о позиции и уверенности каждого элемента.

Процедуру немаксимального подавления и формирования блоков проведем для каждой строки изображения в каждом его масштабе из дискретного набора, зафиксированном на первом этапе. В результате получим первичную информацию о распределении текста.

Объединение первичных блоков в итоговые. Построенные блоки могут относиться к одному текстовому объекту ввиду того, что их формирование проходило на близких строках изображения. Поэтому необходимо выполнить объединение первичных блоков одного масштаба и пересчитать их характеристики:

– будем обрабатывать блоки в порядке убывания средней уверенности, пока не найдем два с достаточной площадью пересечения минимальных прямоугольников (более 50 % от площади любого);

– выполним их посимвольное слияние на базе более уверенного (первого), обходя элементы второго в порядке убывания уверенности:

- сравним координаты и уверенность текущего прообраза (x_2, y_2) второго блока со средней вертикальной координатой (x^1_{aver}) и уверенностью (U^1_{aver}) элементов первого; если $|x^1_{aver} - x_2| > max\_dx$ или $U^1_{aver} - U(x_2, y_2) > dU\_two\_max$ (где $max\_dx = 10$), отбросим этот прообраз;
- если для некоторого прообраза (x_1, y_1) первого блока справедливо $|y_1 - y_2| \le min\_dy$, считаем, что это прообразы одного символа, при этом, когда $|U(x_1, y_1) - U(x_2, y_2)| < dU\_one\_max$, усредним их координаты, иначе запомним характеристики более уверенного прообраза;
- в противном случае добавим текущий прообраз второго блока к первому;
- если изменились характеристики элементов, выполним проверку, аналогичную первым двум подпунктам для всех прообразов первого блока;

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

Примеры тренировочной выборки, используемые для создания детектора, содержат символы различной высоты. Одной из причин данного явления помимо естественной вариативности размеров является искажение образов на каждой эпохе обучения. В связи с этим блоки, сформированные на близких масштабах, так же, как и первичные блоки одного масштаба, могут представлять одинаковый текстовый объект. Справедливость данного замечания подтверждается представленными на рис. 4, *в* блоками, сформированными на разных масштабах изображения.

На заключительном этапе выполним следующие шаги:

– приведем координаты блоков и их образов к единому масштабу;

– объединим блоки с помощью описанной выше процедуры, наложив дополнительное ограничение: исходный масштаб каждого символа блока должен отличаться от среднего не более чем на величину $max\_dS = 0{,}5$;

– отбросим блоки, средняя уверенность которых меньше максимальной более чем на величину dU_one_max.

Изображение итоговых блоков представлено на рис. 4, *г*, при этом помимо координат минимальных прямоугольников результатом работы алгоритма являются также матрицы уверенностей и масштабы блоков.

4. Экспериментальная работа и анализ результатов

Для обучения детектора необходимы маркированные примеры двух классов: фоновые и текстовые, являющиеся полутоновыми изображениями размером 32×32 пиксела. Первые должны содержать фрагменты фонов с различным цветовым составом, текстурой, искусственными и естественными объектами, вторые – текстовый образ, вписанный в изображение с сохранением пропорций его размера. Так как символы слова могут располагаться плотно друг за другом, высока вероятность наличия в его квадратной окрестности сразу нескольких символов, поэтому подобные примеры также нужно включать в обучающее множество, при этом один из символов должен находиться в центре изображения. В целом разнообразие учебных примеров определяется композицией сцен, на которых планируется применять детектор, а также алфавитной принадлежностью искомого текста. Ниже демонстрируется пример создания детектора цифр и заглавных символов английского языка.

Способность детектора автоматически обучаться выделению признаков, по которым возможно разделение текстовых и фоновых объектов, осложняется необходимостью создания объемной базы примеров. Данный процесс связан с определением в множестве изображений сцен позиций минимальных объемлющих прямоугольников отдельных символов, что с гарантированно высоким качеством может быть выполнено только вручную. Некоторые авторы предлагают процедуры генерации искусственных образов путем наложения изображений символов на фиксированные фоны, что облегчает формирование базы [6]. Однако больший потенциал имеют детекторы, обученные на реальных данных, информативность которых значительно выше. В этой связи для выполнения экспериментальной работы была собрана обучающая выборка, включающая следующие базы:

ICDAR 2003 – одна из главных точек отсчета в исследованиях по детектированию, на основе которой проводится сравнение алгоритмов поиска и сегментации текста на изображениях реальных сцен, содержит 12 469 образов символов [8];

74K – представлена изображениями текстов на улицах индийских городов; помимо минимальных прямоугольников содержит маркеры символов, разделена на основную (7705 примеров) и зашумленную (4798 примеров) части [9];

KAIST – представлена изображениями сцен в корейских городах; как и в предыдущих базах, текстовые блоки описаны на уровне строк, слов и символов; разделена на экземпляры, полученные камерой с высоким и низким разрешением (камерой мобильного телефона); содержит 11 537 текстовых образов [10];

SVHN – создана путем сегментации текстовых образов на изображениях табличек номеров домов, содержит 99 289 примеров цифр [11];

CVL OCR DB – словенская база данных, включает 7014 изолированных текстовых образов, выделенных на изображениях рекламных плакатов, дорожных знаков, названий магазинов и др. [12].

Из перечисленных баз были выделены изображения квадратных окрестностей цифр и заглавных букв английского языка, которые масштабировались к размеру 32×32 пиксела. Из получившейся выборки были удалены образы, имеющие низкую уникальность, для оценки которой использовалось суммарное квадратичное отклонение. Кроме того, было сокращено представительство классов, значительно превышающее остальные (например, часто используемых символов 'A', 'C', 'S' и др.). При этом ввиду схожести начертания было усреднено число экземпляров символов 'I' и '1', 'O' и '0', а к классам с недостаточным количеством примеров добавлены их прописные образы в случае их схожести с заглавными, например 'w' и 'W', 'k' и 'K'. В итоге была получена выборка, содержащая 29 172 образа букв и 13 500 цифр, которая была разделена на тренировочную и тестовую части путем отнесения к первой 80 % образов каждого класса. Для получения контрпримеров были использованы фоновые фрагменты изображений сцен указанных баз, в результате общее число образов выборки составило 85 344. Создание детектора выполнялось на базе описанной выше модели в течение четырех циклов обучения: после первого точность классификации образов тренировочной части составила 94,97 %, тестовой – 93,56 %, после четвертого – 96,21 и 94,95 % соответственно. Анализ неверно классифицированных образов показал, что ошибки на текстовых примерах были вызваны их расфокусировкой и значительным искажением шрифта, фоновые же примеры содержали фрагменты, весьма похожие на символы.

Для оценки эффективности предложенной модели созданный детектор был применен для решения задачи автоматического обнаружения минимальных прямоугольников слов в рамках соревнования ICDAR 2013 Robust Reading Competition – Challenge 2: Reading Text in Scene Images – Task 1: Text Localization [13]. Качество обнаружения определялось по результату обработки 233 тестовых изображений реальных сцен (рис. 5), что в количественном измерении выражалось тремя параметрами: *recall* = (число верно локализованных слов) / (общее число слов в выборке), *precision* = (число верно локализованных слов) / (общее число выделенных слов), $F\text{-score} = 2 \cdot recall \cdot precision / (recall + precision)$. При этом использовались специальный способ определения корректности локализации слова с вероятностным выходом и система штрафов в ситуациях соответствия типа «один ко многим» и «многие ко многим» (подробную информацию можно найти в [14]).

Учитывая, что по условию задачи оценке подвергалось качество обнаружения минимальных прямоугольников слов, алгоритм локализации был дополнен следующей процедурой сегментации текстовых блоков:

– обрабатываются образы блока в порядке убывания уверенности;

– первый образ, не отнесенный на текущий момент ни к одному слову, образует новое;

– просматриваются образы блока с начала, пока не будет найден образ, который может быть добавлен к текущему слову при выполнении следующих условий:

- расстояние от образа до какого-либо элемента слова не должно превышать $min\{max\_dy, 1.5 \cdot aver\_dy\}$, где *aver_dy* – средняя удаленность образов слова;
- уверенность, масштаб и горизонтальная координата образа не должны отличаться от их усредненных значений для слова более чем на установленные выше пороговые величины.

Рис. 5. Примеры локализации текста, выполненной с помощью предложенной модели нейросетевого детектора, на изображениях базы ICDAR 2013

Используя дополненную версию алгоритма, была получена следующая оценка локализации слов: *recall* = 60,73, *precision* = 55,14, *F-score* = 57,80 (14-я в списке лучших). Данный результат подтверждает работоспособность созданного детектора, учитывая сложность композиции изображений и высокую стилистическую вариативность текстовых образов. При этом оценка локализации может быть существенно улучшена за счет совершенствования процедуры сегментации блока на слова (для уменьшения штрафов в ситуациях «один ко многим»), что не являлось главной задачей при разработке алгоритма интерпретации откликов детектора.

Качественное сравнение разработанного алгоритма с аналогичными из [6, 7] позволяет отметить, что он обладает большей универсальностью по следующим причинам:

– формирование итоговых блоков происходит путем объединения блоков-кандидатов с учетом индивидуальных характеристик символов, а не простым выбором наиболее уверенного кандидата, что, в частности, не накладывает завышенных требований к точности детектора, позволяя использовать в его основе значительно менее громоздкую нейросетевую архитектуру;

– алгоритм позволяет обнаруживать наклонные текстовые строки, причем максимальный уровень наклона может быть увеличен за счет изменения параметра *max_dx*;

– объединение результатов обработки на разных масштабах позволяет формировать блоки с регулируемым уровнем отличия размеров символов (*max_dS*), что необходимо для учета отличия регистра символов и вариативности их стилистического оформления.

Заключение

В статье представлена модель текстового детектора на базе сверточной нейросети авторской архитектуры, преимуществом которой, по сравнению с аналогичными, является сочетание высокой обобщающей способности и низкой вычислительной стоимости обучения и применения. Предложен алгоритм прохода детектора по изображению, позволяющий обнаруживать текстовые образы произвольного размера и пространственного распределения. Описаны как универсальные пути его оптимизации, так и основанные на априорной контекстной информации. Разработан алгоритм локализации текстовых блоков на изображениях реальных сцен путем интерпретации откликов детектора. В отличие от аналогичных он не предъявляет завышенных требований к точности детектора, позволяет применять емкие нейросетевые архитектуры и обнаруживать наклонные текстовые объекты, обладает большей универсальностью за счет объединения откликов детектора, полученных на различных масштабах входного изображения.

Список литературы

1. Sumathi, C.P. A Survey on various approaches of text extraction in images / C.P. Sumathi, T. Santhanam, G. Gayathri // International Journal of Computer Science & Engineering Survey. – 2012. – Vol. 3, № 4. – P. 27–42.

2. LeCun, Y. Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou // Proceedings of the IEEE. – 1998. – Vol. 86, № 11. – P. 2278–2324.

3. Кузьмицкий, Н.Н. Сверточная нейросетевая модель в задаче классификации изображений изолированных цифр / Н.Н. Кузьмицкий // Доклады БГУИР. – Минск, 2012. – № 7. – С. 64–70.

4. Головко, В.А. Нейронные сети: обучение, организация и применение : учеб. пособие / В.А. Головко. – М. : ИПРЖР, 2001. – Кн. 4. – 256 с.

5. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2002. – 344 с.

6. Delakis, M. Text detection with convolutional neural networks / M. Delakis, Cr. Garcia // Intern. Conf. on Computer Vision Theory and Applications. – Cambridge, 2008. – P. 290–294.

7. Wang, K. End-to-end scene text recognition / K. Wang, B. Babenko, S. Belongie // IEEE Intern. Conf. on Computer Vision (ICCV). – Barcelona, 2011. – P. 1457–1464.

8. ICDAR 2003 robust reading competitions / S.M. Lucas [et al.] // Proc. of Seventh Intern. Conf. on Document Analysis and Recognition. – Edinburgh, 2003. – P. 682–687.

9. Campos, T.E. Character Recognition in Natural Images / T.E. Campos, B.R. Babu // VISAPP. – 2009. – Vol. 2. – P. 273–280.

10. Touch TT : Scene text extractor using touchscreen interface / J. Jung [et al.] // ETRI Journal. – 2011. – Vol. 33, № 1. – P. 78–88.

11. The Street View House Numbers (SVHN) Dataset [Electronic resource]. – 2011. – Mode of access : http://ufldl.stanford.edu/housenumbers. – Date of access : 03.07.2014.

12. Ikica, A. An improved edge profile based method for text detection in images of natural scenes / A. Ikica, P. Peer // Intern. Conf. on Computer as a Tool (EUROCON). – Lisbon, 2011. – P. 1–4.

13. ICDAR 2013 Robust Reading Competitio / D. Karatzas [et al.] // Proc. 12th Intern. Conf. of Document Analysis and Recognition, IEEE CPS. – Washington, 2013. – P. 1115–1124.

14. Wolf, C. Object Count Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms / C. Wolf, J.M. Jolion // International Journal of Document Analysis. – 2006. – Vol. 8, № 4. – P. 280–296.

Поступила 07.04.2015

Брестский государственный
технический университет,
Брест, ул. Московская, 267
e-mail: knnbrest@yandex.ru

N.N. Kuzmitsky

DETECTION OF TEXT OBJECTS IN IMAGES OF REAL SCENES BASED ON CONVOLUTIONAL NEURAL NETWORK MODEL

A model of text image detector based on a convolutional neural network architecture is presented, capable of synthesizing high-level features of images in the «black box» mode. An implementation of the detector application, based on algorithms of multi-scale scanning and local responses interpretation is described, allowing to find out text samples on images of real scenes. Advantages in comparison with analogs are shown and efficiency evaluation on an example of a known database is conducted.