Правительство Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Кафедра Системного Программирования

Луценко Александр Николаевич

Обнаружение и распознавание потребительских товаров на изображениях на основе свёрточных нейронных сетей

Бакалаврская работа

Допущена к защите.
Зав. кафедрой:
д. ф.-м. н., профессор Терехов А. Н.

Научный руководитель:
д. ф.-м. н., профессор Терехов А. Н.

Рецензент:
ведущий разработчик ООО «НМТ» Невоструев К. Н.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY
Software Engineering Chair

Alexander Lutsenko

Detection and recognition of consumer goods in images based on convolutional neural networks

Bachelor's thesis

Admitted for defence.
Head of the chair:
Professor Andrey Terekhov

Scientific supervisor:
Professor Andrey Terekhov

Reviewer:
Senior developer at "NMT" LLC Constantin Nevostruev

Saint-Petersburg
2016

Оглавление

Введение

Прорывы, произошедшие в компьютерном зрении в последние несколько лет, открыли ранее недостижимые горизонты и вдохновили специалистов искать новые сферы применения технологиям компьютерного зрения.

Одна из таких новых задач, которую всё чаще ставят перед собой крупные зарубежные торговые холдинги, – это задача автоматической проверки правильности раскладки товаров на полках магазинов.

Выкладка товаров является особым инструментом достижения целей мерчендайзинга. Практика показала, что отдельные точки пространства торгового зала магазина по-разному стимулируют продажи. Следовательно, товары и их марки могут получить конкурентные преимущества в зависимости от занимаемых ими мест на прилавках.

Раскладку товара нужно периодически, по мере опустошения полок, корректировать, а необходимость этого – сначала проверить. Кроме того, такая проверка требуется начальству торгового заведения для оценки качества работы соответствующего персонала. Сейчас этим, за редким исключением, занимаются люди, что стоит торговым сетям ощутимых денег. Закономерно, что появилось желание ради сокращения издержек проверять раскладку товара автоматически. А в последние годы, в связи с прорывами в области компьютерного зрения, подобные автоматические системы становятся реальностью. Схема работы следующая:

1. Сотрудник магазина или аппаратура делает снимок витрины
2. На изображении находятся и распознаются товары
3. Расположение товаров сравнивается с планограммой, генерируется отчёт о выкладке

Сложная часть схемы – второй пункт, автоматическое обнаружение и распознавание товара на фото. Именно это и является предметом исследования в данной работе.

1. Постановка задачи

Цель работы – создать алгоритм обнаружения и распознавания товаров на магазинных витринах. В качестве полигона для апробации прототипа взята более узкая задача поиска и опознавания напитков (разные виды бутылок и банок).

Имеющиеся тренировочные данные неудовлетворительного качества, их предварительно нужно подготовить.

Можно выделить следующие задачи:

- Скорректировать тренировочные данные
- Разработать алгоритм обнаружения и распознавания напитков. Задача естественным образом разбивается на две подзадачи:
    1. Локализовать целевые объекты на изображении, т. е. найти их пространственные координаты и границы (ограничивающие прямоугольники)
    2. Классифицировать найденные объекты

2. Описание и подготовка данных

Обучающие данные – фотографии магазинных витрин с бутылками и банками – были предоставлены частной зарубежной компанией в рамках конкурса по автоматическому распознаванию напитков на изображениях.

Количество изображений – 970. К сожалению, качество разметки данных оставляло желать лучшего, было принято решения взять часть данных и доразметить вручную. Для подготовки данных была написана программа со следующими функциями:

- Разметка изображений через графический интерфейс
- Генерация тренировочных / тестовых данных
  - Вырезка участков, содержащих целевой объект (crop)
  - Получение тепловой карты изображения (heat map)
  - Аугментация (augmentation, "раздутие") данных – деформация имеющейся выборки с целью увеличить объём данных (масштабирование, поворот, зеркальное отражение, цветовые искажения)

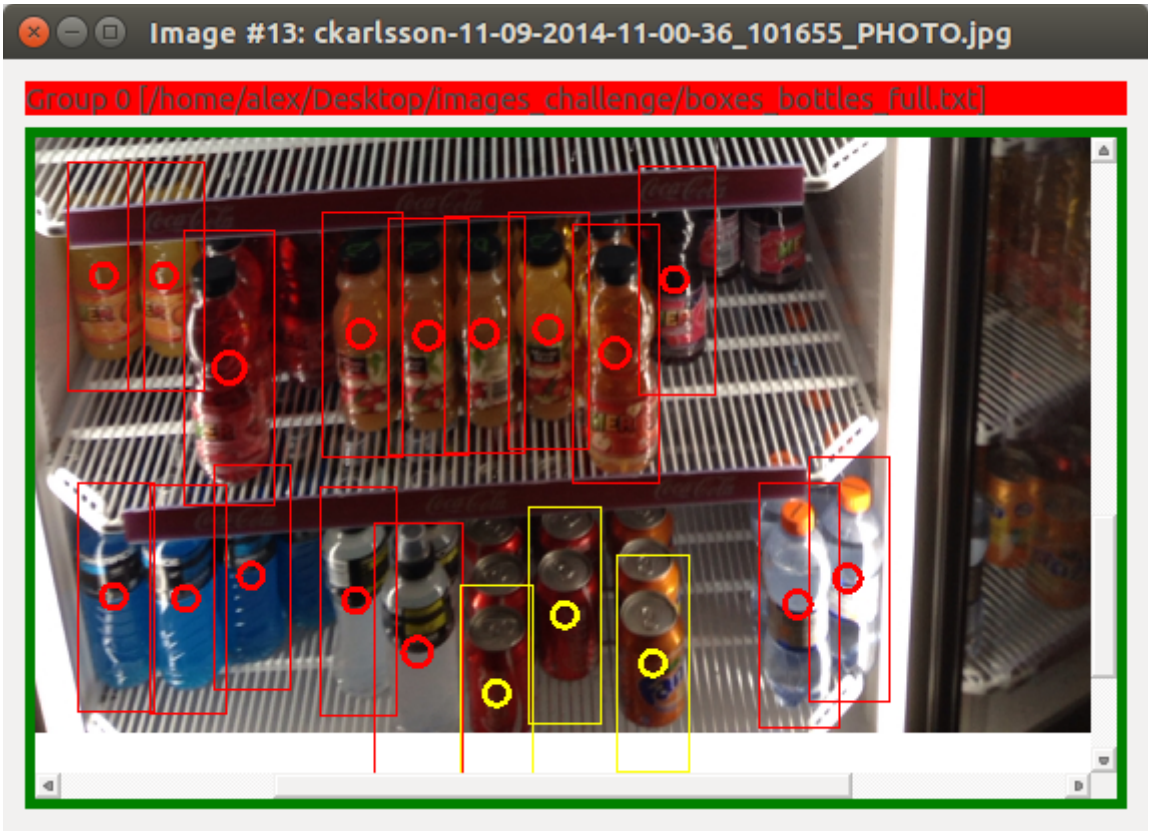


рис. 1: Пример размеченного тренировочного изображения

Количество качественно доразмеченных фото – 325 на них бутылок – 3675, банок – 3850.

Предоставленные данные содержали не только тип объектов (бутылка/банка), но и конкретный класс

- Бутылки – 18 классов
- Банки – 16 классов

Тренировочные данные разбалансированы, классы сильно отличаются по количеству данных и сложности опознавания.

3. Специфика задачи

Стоящая передо мной задача локализации и классификации напитков на витринах имеет определённую специфику. С одной стороны, это является осложняющим фактором, делающим существующие подходы менее применимыми в данном конкретном случае. С другой стороны, некоторые особенности можно выгодно использовать для повышения точности алгоритма.

Особенности задачи локализации

- Товары находятся примерно на одном расстоянии от камеры, т. е. имеют почти одинаковый масштаб
  - \+ Можно найти масштаб и искать объекты только в определённом узком диапазоне размеров, уменьшив время работы алгоритма и количество ложных срабатываний
- Товаров на магазинных полках, как правило, много
  - – Самые точные на данный момент методы локализации, нейросетевые, пока плохо справляются с большим количеством объектов на одном изображении
  - \+ Как уже упоминалось, интересующие нас объекты имеют примерно одинаковый масштаб. И чем их больше, тем легче этот масштаб определить.
- Товары зачастую расположены вплотную друг к другу
  - – Слишком плотно расположенные объекты частично перекрывают друг друга, затрудняя обнаружение
  - \+ Можно использовать для повышения точности: участок рядом с уже найденными объектами имеет бóльшую вероятность содержать объект

Особенности задачи классификации

- Большинство производителей напитков стремятся сделать внешний вид товара самобытным и узнаваемым
  + Проще опознавать не только людям, но и машинам
- Многие виды бутылок/банок очень похожи друг на друга формой. При классификации нужно опираться прежде всего на текстуру и цвет.
  – Текстура плохо опознаётся на смазанных изображениях и в малом разрешении
  – На цвет объекта существенно влияет освещение прилавка, которое может сильно варьироваться

4. Метрики оценки качества

Классификация

Так как классы в тестовых выборках для наших задач локализации и классификации разбалансированы, такой широко используемый показатель качества как accuracy (доля объектов выборки, которым класс был присвоен верно) не подходит. В подобных случаях уместно использовать метрики, именуемые точностью ($precision$) и полнотой ($recall$).

Точность ($precision$) для класса – доля корректно классифицированных объектов среди всех объектов, отнесённых классификатором к данному классу. Более формально:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

Полнота ($recall$) для класса – доля корректно классифицированных объектов среди всех объектов, принадлежащих данному классу:

$recall = \frac{true\ positives}{true\ positives + false\ negatives}$, где

$true\ positives$ – количество объектов, верно отнесённых к данному классу

$false\ positives$ – количество объектов, ошибочно отнесённых к данному классу

$false\ negatives$ – количество объектов, ошибочно не отнесённых к данному классу

Локализация

Чтобы определить, насколько точно на изображении выделен объект, считают похожесть его предсказанного и истинного ограничивающих прямоугольников. Распространённый показатель похожести – Intersection over Union (IoU, пересечение над объединением).

IoU – доля общего пространства, занимаемого обоими прямоугольниками:

$IoU = \frac{predicted\ box \cap true\ box}{predicted\ box \cup true\ box}$, где

$predicted\ box$ – предсказанный ограничивающий прямоугольник для целевого объекта

$true\ box$ – настоящий ограничивающий прямоугольник для целевого объекта

Будем считать, что объект локализован верно, если $IoU > 0.5$. По этому критерию предсказания бинаризуются (на верные и неверные), и для них вычисляются точность и полнота.

5. Существующие подходы к локализации объектов на изображении

5.1. Контурный анализ

Бутылки имеют своеобразные очертания, по которым их можно надёжно опознавать.

На изображении выделяются замкнутые контуры, фильтруются и корректируются, а затем сравниваются с несколькими эталонными контурами бутылок.

Выяснилось, что для данной задачи такой подход совершенно непригоден: как уже отмечалось, на изображениях объекты зачастую стоят вплотную, частично перекрываются друг другом или витриной, а плохое освещение ещё сильнее отягчает ситуацию. Всё это делает сколь-нибудь качественное выделение контуров невозможным.

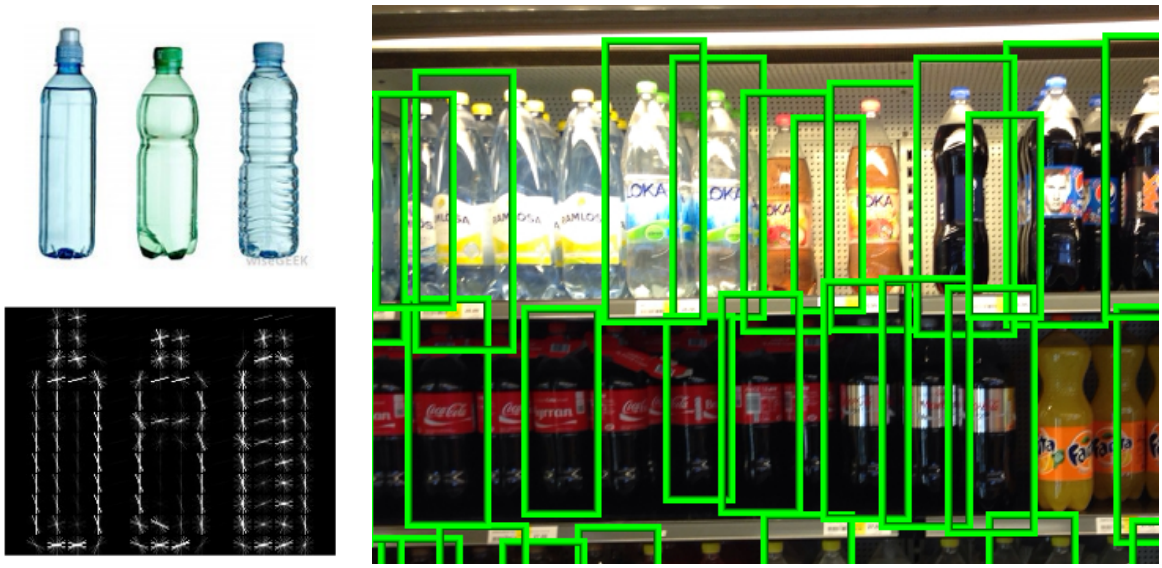
a) b)

рис. 2: Анализ контуров хорошо работает в "лабораторных" условиях (a), но непригоден для данной задачи (b)

5.2. HOG/SVM

Изображение разбивается на небольшие непересекающиеся квадратные сегменты, в каждом из них считаются интенсивности перепадов цветов в различных направлениях (обычно в восьми–девяти), которые затем нормализуются по блоку. На полученных признаках, т. н. HOG-дескрипторах [1], обучается бинарный классификатор (есть объект / нет объекта). Обычно в качестве классификатора выступает машина опорных векторов (Support Vector Machine, SVM), Чтобы уменьшить переобучение, используют линейную SVM, так как размер дескриптора велик (около 1000 вещественных чисел на каждый сканируемый участок), а тренировочных данных довольно мало.

Производится несколько раундов обучения. После каждого раунда специально подобранные изображения, на которых объектов заведомо нет, сканируются классификатором. Участок, где найден объект (ложное срабатывание), добавляется в тренировочную выборку в качестве сложного отрицательного примера. Через 4–5 раундов классификатор достигает своей пиковой точности и перестаёт улучшаться.



a) b)

рис. 3: HoG-дескрипторы бутылок (a) и пример работы алгоритма на фотографии витрины (b)

Такой подход быстр как в обучении, так и в работе (обучение: 3 мин./раунд * 5 раундов, поиск объектов: ~1 сек. на изображение). Находит объекты разных размеров. Основной же недостаток применительно к данной задаче – довольно низкое качество обнаружения.

	Точность	Полнота	Время поиска
Бутылки	0.93	0.70	1–2 сек.
Банки	0.75	0.83	

HOG-дескриптор и SVM – библиотека OpenCV

5.3. Свёрточные нейронные сети

Нейронная сеть – дифференцируемая (почти во всех точках) функция, обычно организованная слоями:

$NN(\mathbf{x}) = L_n \circ L_{n-1} \circ \ldots \circ L_1(\mathbf{x})$, где

L_k – слой нейронной сети, функция, каким-то образом преобразующая входные данные и (как правило) содержащая настраиваемые параметры $\mathbf{w}_k$:

$L_k(\mathbf{x}_k, \mathbf{w}_k) \in \mathbb{R}^{in_k} \times \mathbb{R}^{param_k} \to \mathbb{R}^{out_k}$

Слой может быть линейной функцией (полносвязный, свёрточный слои), нелинейной (т. н. функция активации: tanh, ReLU) либо выполнять какие-то служебные действия (DropOut, Batch Normalization).

Так выглядит поток преобразования данных внутри многослойной нейросети:

$$NN(\mathbf{x}) = \mathbf{y}: \quad \mathbf{x} \to \underset{\uparrow \mathbf{w}_1}{L_1} \to \underset{\uparrow \mathbf{w}_2}{L_2} \to \ldots \to \underset{\uparrow \mathbf{w}_n}{L_n} \to \mathbf{y}$$

Каждый следующий слой принимает на вход данные, произведённые предыдущим. Входные данные сети – признаки анализируемого объекта, с каждым слоем эти признаки преобразуются в новые, более сложные и информативные. В этом и состоит, пожалуй, главное преимущество нейронных сетей: они сами формируют новые признаки из уже существующих, снимая это бремя с человека.

Эта отличительная черта наиболее ярко проявляется в тех задачах, где исходные признаки по отдельности очень мало значат (например, отдельный пиксель изображения). Именно способность формировать информативные признаки обусловила успех нейронных сетей в области компьютерного зрения.

Обучить нейронную сеть – значит найти оптимальные значения настраиваемых параметров $\mathbf{w}_1 \ldots \mathbf{w}_n$. Находятся они стандартным для дифференцируемых функций методом – градиентным спуском. Градиенты по параметрам для многослойной нейронной сети вычисляются методом обратного распространения ошибки, базирующимся на очень известном свойстве производных, цепном правиле.

Свёрточной называют нейронную сеть, в которой в качестве линейного слоя используется функция свёртки.

Обычный полносвязный линейный слой не принимает во внимание пространственную структуру данных. Кроме того, для задач распознавания изображений полная связность избыточна, а огромное количество параметров полносвязного линейного слоя быстро приводит к переобучению.

Свёрточная нейронная сеть вдохновлена биологией и призвана эмулировать работу зрительной коры головного мозга млекопитающего. В задачах компьютерного зрения она превосходит классическую архитектуру за счёт использования сильной корреляции между близкими пикселями, имеющей место на реальных изображениях.

Свёрточный слой имеет следующие отличительные особенности:

1. Локальная связность. Нейроны внутри свёрточного слоя соединены лишь с небольшим регионом предыдущего. Последовательное соединение множества таких слоёв (разделённых слоями активации) приводит к формированию нелинейных фильтров, постепенно становящихся всё более глобальными, т. е. обозревающими более обширные регионы изображения. Это позволяет нейронной сети сначала сформировать хорошие признаки для маленьких кусочков изображения, а затем на их основе создать признаки для участков размером побольше.
2. Общие параметры. В свёрточном слое один и тот же фильтр используется для всех участков входного изображения. Такой подход резко снижает количество настраиваемых параметров слоя, а значит, время обучения и количество потребляемой памяти. Кроме того, это сильно ограничивает запоминающую способность нейронной сети, подталкивая её при обучении к обобщению демонстрируемой информации, а не попиксельному запоминанию каждого показанного изображения.

Из недостатков нейронных сетей следует отметить большое количество требуемых для обучения данных и относительно низкую скорость работы.

5.3.1. R-CNN

Самый простой и очевидный способ решать задачу детектирования – свести её к задаче классификации. Чтобы обнаружить объект, берётся специфический для этого объекта классификатор и применяется к всевозможным участкам изображения в разных масштабах. Более современные методы, такие как R-CNN [2], не проверяют все области на изображении, а генерируют множество окон-кандидатов (прямоугольных областей изображения, которые, предположительно, ограничивают какой-то объект), а затем запускают классификатор только на этих окнах, предварительно отмасштабировав их к единому размеру.

Наиболее распространённые и эффективные подходы к генерации окон-кандидатов:

- Selective search [3]. Основан на том наблюдении, что объекты обычно состоят из одной или нескольких частей, каждая из которых имеет почти монотонный окрас. Кандидатами являются области с примерно одинаковым цветом, а также всевозможные объединения таких областей, смежных друг с другом.
- Edge box [4]. Контуры часто очерчивают границы объекта. В роли кандидатов выступают области, содержащие замкнутые или почти замкнутые контуры.

Недостатки R-CNN применительно к задаче локализации товаров на витринах:

- Упомянутые методы генерации окон-кандидатов на данной задаче работают плохо, что приводит к ошибкам второго рода (пропускам объекта)
- Из-за того, что все кандидаты разного размера, каждый проверяется по отдельности. В результате имеем большое время и поиска, и обучения.
- У нас все объекты на фото примерно одного (но неизвестного) размера, а вот размер окон-кандидатов может быть любым, и их всех нужно проверить. Как результат – подверженность ошибкам первого рода (ложным срабатываниям). Зная размер объектов, можно было бы сразу отсеивать кандидатов неподходящего масштаба.

	Полнота кандидатов	Количество кандидатов	Время поиска кандидатов	Время проверки кандидатов
Бутылки	0.80	5000–6500	3–4 сек.	200–260 сек.
Банки	0.68			

Алгоритм генерации окон-кандидатов Selective Search – библиотека dlib

5.3.2. YOLO

Новый альтернативный подход к локализации объектов на изображении – YOLO (You Only Look Once) [5]. Задача локализации формулируется как одноступенчатая задача регрессии: единственная нейронная сеть принимает на вход всё изображение целиком и выдаёт координаты ограничивающих прямоугольников и вероятности принадлежности классам для них. YOLO делит изображение на решётку размера $S \times S$. За обнаружение объекта ответственна та ячейка решётки, в которую попадает его центр. Каждая ячейка ищет B ограничивающих прямоугольников одного и того же класса. Прямоугольник характеризуется пятью числами – 4 координаты и уверенность в нём нейронной сети. На выходе система выдаёт тензор размера $S \times S \times (B * 5 + C)$ (в оригинальной работе $S = 7$, $B = 2$, C – количество классов).

YOLO накладывает сильные ограничения на пространственное расположение объектов. Из-за своей решётчатой структуры система не справляется с поиском мелких объектов, расположенными плотными группами. В частности, YOLO мало применим к задаче обнаружения товаров на витринах.

6. Предложенный алгоритм локализации напитков

Как было упомянуто выше, вдумчивое использование специфики задачи даёт потенциальную возможность улучшить точность локализации. Предложенный гибридный подход использует эту специфику, комбинируя HOG-дескриптор и SVM с нейросетевыми методами.

Разработанный алгоритм состоит из трёх основных частей:

- Локализатор объектов на основе HOG дескриптора и линейной SVM
- Нейронная сеть-верификатор (рис. 5 а)
    - Входные данные – участки изображения, предположительно содержащие напиток, размером 40 на 60 пикселей. Такой формат позволяет видеть не только подлежащий верификации объект, но и двух его "соседей" по бокам, если таковые имеются.
    - Выход нейронной сети – оценки вероятностей принадлежности входного изображения трём классам: бутылка, банка, что-то другое.
    - Обучение нейронной сети производилось по новой методике, именуемой противоборствующим градиентом (adversarial gradient) [7]. Это позволило достичь большей точности верификации и впоследствии генерировать более аккуратные тепловые карты.
- Нейронная сеть-корректор
    - Входные данные – фото магазинного прилавка произвольного размера и полученная по ней тепловая карта
    - Выход нейронной сети – скорректированная тепловая карта
    - В отличие от верификатора, нейросеть-корректор была натренирована на полных фотографиях. Так как её задача – лишь подправить тепловую карту, а не создать "с нуля", в качестве архитектуры для корректора была выбрана так называемая сеть остатков (residual network, ResNet) [8]. При корректировке тепловой карты её полезно

"видеть" на всех этапах обработки. В стандартной архитектуре в процессе прохода входной информации через нейросеть она сильно преобразуется и становится недоступной в изначальном виде.
Сеть остатков же проносит информацию о входных данных (исходной тепловой карте) через всю сеть практически в первозданном виде, существенно помогая в обучении, ускоряя его.

Схема работы алгоритма

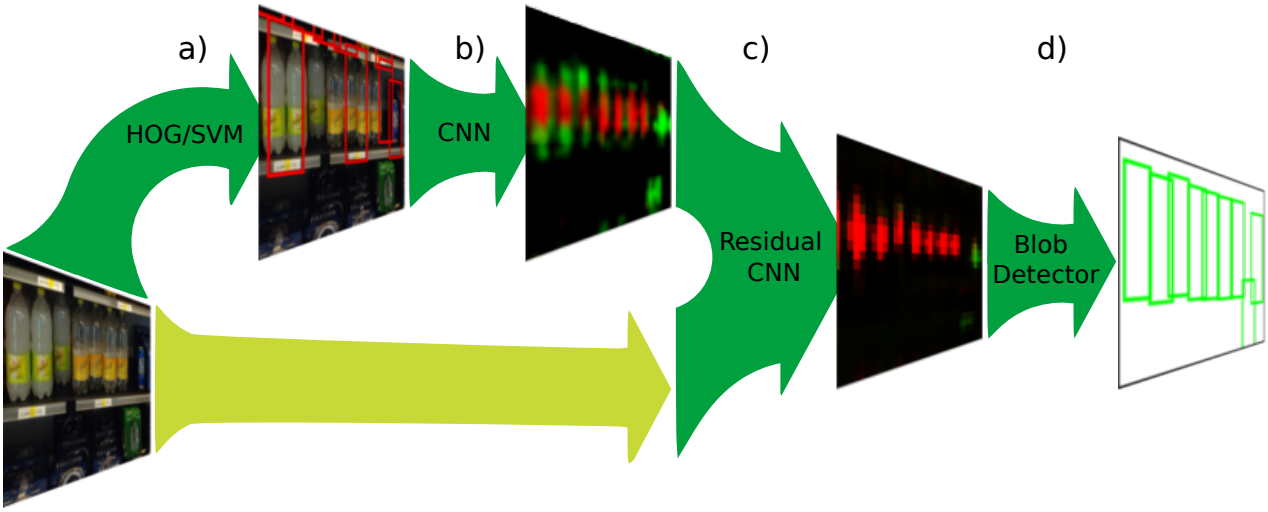


рис. 4: Схема работы алгоритма локализации

1. Определяем размер искомых объектов на изображении (рис. 4 a)

   С помощью связки HOG/SVM получаем набор кандидатов в объекты. Проверяем кандидатов предварительно обученной сетью-верификатором. Замеряем размер тех, что прошли верификацию. В качестве типичного размера объектов на изображении берём их медиану.

2. Масштабируем изображение

   Зная медианный размер объектов на конкретном изображении, масштабируем его таким образом, чтобы объекты приняли заранее установленный эталонный размер (тот, под который обучалась сеть-верификатор).

3. Получаем тепловую карту (рис. 4 b)

   Тепловая карта (heat map) – это способ графического представления данных, при котором значения сгруппированы в матрицу и отображаются

с помощью цвета.

В нашем случае тепловая карта – матрица того же размера, что и анализируемое изображение. Тепловая карта имеет три канала (можно представить их цветами), каждый из которых сигнализирует о присутствии объектов соответствующего каналу типа в определённых частях изображения (бутылка, банка или ни то, ни другое).

Тепловую карту можно легко сгенерировать, пропустив через нейросеть-верификатор всё отмасштабированное изображение целиком. Получившаяся карта недостаточно точна и зашумлена. Основные проблемы состоят в следующем:

- Нижние и верхние части бутылок, с точки зрения нейронной сети, похожи на банки
- Область видимости сети-верификатора (40 на 60 пикселей) может вместить только целевой объект и небольшие участки пространства по бокам. Это вся информация, на основании которой принимается решение о наличии или отсутствии напитка в данном участке изображения.

4. Корректируем тепловую карту (рис. 4 c)

   Первичную карту вместе с исходным изображением подаём на вход нейросети-корректору. На скорректированной карте предположительные местоположения объектов обозначены пятнами.

5. Получаем ограничивающие прямоугольники (рис. 4 d)

   С помощью существующих методик обнаружения пятен (blob detection) на уточнённой тепловой карте находим центры объектов. Зная размер объектов и координаты их центров, получаем искомые ограничивающие прямоугольники.

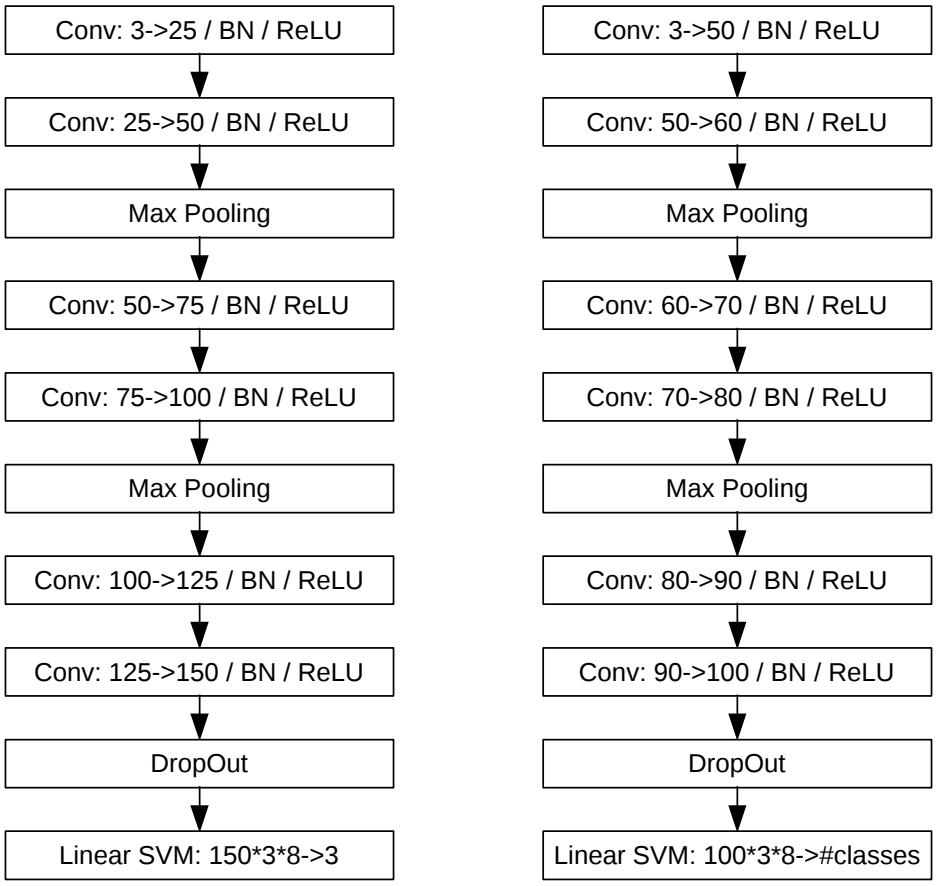
7. Классификация напитков

Тренировочные данные имели тот же вид, что и данные для первого этапа в задаче локализации: изображения размера 40 на 60 пикселей, захватывающие не только подлежащий классификации напиток, но и пространство справа и слева от него. Мотивация следующая: товары на полках сгруппированы по типу, и если для целевого объекта сложно определить класс, полезно будет посмотреть на его "соседей".

Чтобы увеличить количество тренировочных данных, изображения были взяты в четырёх масштабах, от 0.9 до 1.05 от изначального, и зеркально отражены.

Для определения класса бутылок и банок были обучены две глубокие свёрточные нейронные сети (рис. 5 b). Количество обучающих данных для разных классов сильно отличается, поэтому стандартная для классификации функция потерь, кросс энтропия, плохо справляется с этой задачей. Подправить ситуацию можно путём взвешивания классов – домножения функции потерь на объекте из тренировочной выборки на величину, обратно пропорциональную размеру класса, но это не решает проблему полностью.

Чтобы справиться с разбалансировкой классов, была использована уже упомянутая SVM. SVM – дифференцируемая функция, поэтому её можно использовать как часть нейронной сети, заменив ею последний линейный слой и функцию потерь [6]. Обучение нейронной сети с SVM в качестве верхнего слоя оказалось не такой простой задачей. В ходе обучения параметры SVM "дрожали" не сходясь к определённому значению, что существенно снижало точность классификации. На удивление эффективным оказалось простое решение: не обучать SVM вовсе, оставив без изменений первоначальные, случайно присвоенные, параметры, а подстраивать под них нижерасположенные слои глубокой нейронной сети. По сравнению с традиционной архитектурой (обучаемый линейный слой сверху и кросс энтропия в качестве функции потерь), такой подход позволил достичь приемлемой точности даже на тех классах бутылок, для которых было очень мало обучающих данных, и уменьшил совокупную ошибку классификации бутылок на 20%. Для банок, однако, прироста точности не наблюдалось.



a) b)

рис. 5: Схема верификаторующей (a) и классифицирующей (b) нейронных сетей.

Conv (Convolution): <входных карт> -> <выходных карт> – свёрточный слой. Размер окна 3x3, шаг 1.

BN (Batch Normalization) – слой пакетной нормализации. Масштабирует входные данные так, чтобы они попадали в активную зону следующей за ним функции активации. Ускоряет обучение.

ReLU (Rectified Linear Unit) – функция активации. Нелинейно преобразует входные данные.

Max Pooling – слой пулинга. Уменьшает размер входного изображения. Размер окна 2x2, шаг 2.

DropOut. Во время тренировки случайным образом обнуляет часть входных данных, уменьшая переобучение. Вероятность обнуления 0.5.

Linear SVM – линейная машина опорных векторов.

8. Результаты

Локализация

IoU > 0.5

	Бутылки		Банки		Время поиска
	Точность	Полнота	Точность	Полнота	
HOG/SVM (OpenCV)	0.93	0.70	0.75	0.83	1–2 сек.
Гибридный подход	0.95	0.91	0.88	0.90	10–15 сек.

Классификация

Бутылки

Класс	Обучающих примеров	Точность	Полнота
cola	439	0.9814	0.9959
fanta	270	0.9843	0.9921
mer	255	0.9957	0.9943
fanta red	157	0.9906	0.9791
bonaqua	121	0.9844	0.9223
fanta white	113	0.9719	0.9692
sprite	92	1.0000	0.9969
cola white	87	0.9924	0.9705
cola black	75	0.9866	0.9250
cola green	65	0.9937	1.0000
powerade	31	0.9867	0.9802
fanta glass	20	0.9743	0.9500
cola glass	14	0	0
powerade white	12	0.7619	1.0000
fanta big	8	1.0000	1.0000
minute maid	7	1.0000	1.0000
powerade yellow	6	1.0000	1.0000
other	1322	0.9775	0.9903

Банки

Класс	Обучающих примеров	Точность	Полнота
cola	287	0.9888	0.9761
mer green	202	0.9245	0.8750
burn	124	0.9219	0.9486
bonaqua	68	0.9074	0.9800
fanta green	65	0.9179	0.8605
sprite	59	0.9665	0.9099
cola tall	45	0.9719	0.8125
monster 4	39	0.9741	0.7847
cola black	36	0.7593	0.9017
monster 3	35	0.7721	0.8970
fanta yellow	27	0.8675	0.9097
mer yellow	23	0.7710	1.0000
cola green	16	0.7500	0.3750
monster 1	9	0.9696	0.8888
monster 2	5	0.7777	0.2187
other	655	0.9567	0.9773

Проблемы с классификацией банок:

- Тренировочных данных меньше, чем для бутылок
- Банки, в отличие от бутылок, имеют одну и ту же форму. Значит, при классификации остаётся полагаться на цвет и рисунок
  - Цвет сильно зависит от освещения
  - Рисунок плохо различим на некачественных фото (размытых или в низком разрешении). Кроме того, многие банки на прилавках повёрнуты этикеткой вбок или назад.

9. Недостатки предложенного подхода

Локализация

- Неплохо работает, если объекты стоят стройными рядами, заметно хуже – когда по отдельности
- Определение масштаба не всегда хорошо работает, если на изображении мало целевых объектов

Классификация

- Так как классификатор во многом ориентируется на цвет напитка, сильные изменения освещения существенно снижают общую точность классификации
- Выяснилось, что нейронные сети хорошо справляются с классификацией бутылок, однако для банок требуется иной подход

Заключение

В рамках работы были выполнены следующие задачи:

- Написано приложение с графический интерфейсом для разметки и подготовки данных
- Разработан гибридный алгоритм локализации напитков на фотографиях магазинных прилавков. Предложенный алгоритм использует специфику задачи и сочетает HOG-дескрипторы и SVM с глубокими нейронными сетями.
- Спроектированы и обучены нейронные сети, распознающие определённые виды бутылок и банок. Внесённая модификация (SVM с замороженными параметрами) позволила добиться удовлетворительной точности для бутылок даже на тех классах, для которых было очень мало тренировочных данных.

Список литературы

[1] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524

[3] J. R. Uijlings, K. E. van de Sande, T. Gevers, A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

[4] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014, pages 391–405. Springer, 2014.

[5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640v4

[6] Y. Tang. Deep Learning using Linear Support Vector Machines. arXiv:1306.0239

[7] A. Nøkland. Improving Back-Propagation by Adding an Adversarial Gradient. arXiv:1510.04189v2

[8] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385