



НАЦИОНАЛЬНЫЙ ЦЕНТР
КОГНИТИВНЫХ РАЗРАБОТОК
УНИВЕРСИТЕТА ИТМО

НЦКР



Программная библиотека инструментов анализа изображений фасадов зданий

описание программы

руководство оператора

руководство системного программиста

Санкт-Петербург, 2020 г.



АННОТАЦИЯ

Документ содержит описание, руководство оператора и руководство системного программиста по работе с программной библиотекой инструментов анализа изображений фасадов зданий, реализующей алгоритмы и математические модели автоматизированного распознавания элементов фасадов зданий на основе панорамных изображений улиц с применением технологий машинного зрения на базе нейронной сети для автоматизации процессов аудита размещения рекламных и информационных конструкций на фасадах зданий.



ОГЛАВЛЕНИЕ

1. Общие сведения	4
2. Функциональное назначение.....	4
3. Описание логической структуры	4
3.1 Архитектура программной библиотеки.....	4
3.2 Корректировка геометрии исходного изображения фасада.....	4
3.3 Сегментация изображения фасада здания	5
3.4 Применение комплекса правил для распознавания незаконных элементов фасадов.....	6
4. Используемые технические средства	9
5. Настройка программы	9
6. Выполнение программы	10
7. Входные данные	11
8. Выходные данные	11



1. Общие сведения

Программная библиотека инструментов анализа изображений фасадов зданий реализует алгоритмы и математические модели автоматизированного распознавания элементов фасадов зданий на основе панорамных изображений улиц. Инструменты программной библиотеки предназначены на выполнение полного цикла автоматизированной аналитики, включая предварительную обработку исходных панорамных изображений, распознавание элементов фасадов, выделение на них информационных вывесок и идентификация вероятных несогласованных конструкций на основе группы правил. Распознавание изображений реализовано с применением технологий машинного зрения на базе нейронной сети. Программная библиотека нацелена на автоматизацию процессов аудита (проверки законности) размещения рекламных и информационных конструкций на фасадах зданий.

2. Функциональное назначение

Программная библиотека предназначена для решения следующих задач:

- Выравнивание исходных панорамных изображений городских улиц для выполнения дальнейшего распознавания элементов фасадов.
- Определение при помощи машинного зрения элементов фасадов зданий.
- Определение при помощи машинного зрения информационных вывесок и иных конструкций на фасадах зданий.
- Выявление информационных вывесок и иных конструкций, которые могут носить признаки незаконных (не получивших согласования).

3. Описание логической структуры

3.1 Архитектура программной библиотеки

Программная библиотека реализует полный цикл анализа изображений фасада от предварительной корректировки геометрии фотографий до генерации наглядных результатов детектирования незаконных элементов фасадов. Логическая структура программной библиотеки приведена на рисунке 3.1.1.

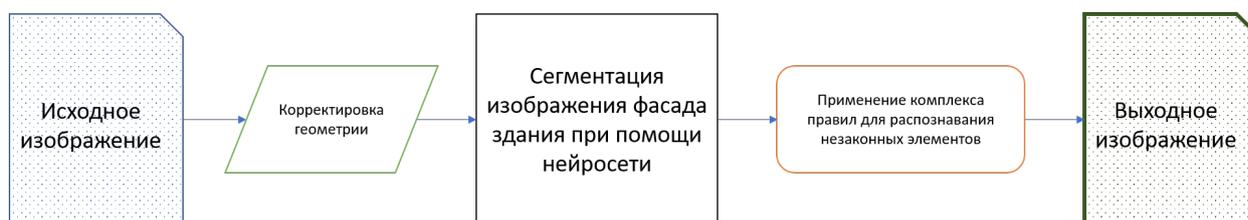


Рисунок 3.1.1 – Логическая структура программной библиотеки

Пользователь взаимодействует с программной библиотекой инструментов анализа изображений фасадов зданий на локальном компьютере. Библиотека имеет вид программного кода, запускаемого в интегрированной среде разработки Jupyter Notebook. После запуска библиотека взаимодействует с открытой программной библиотекой для машинного обучения TensorFlow и формирует выходное изображение с размеченными элементами фасадов.

3.2 Корректировка геометрии исходного изображения фасада

Корректировка геометрии исходного панорамного изображения фасада (выпрямление изображения) требуется для того, чтобы гарантировать, что изображение не искажается таким образом,



который повлиял бы на результат применения проверок на законность. Поскольку программная библиотека нацелена на унифицированное использование с различными источниками панорамных изображений, невозможно гарантировать наличие двух фотографий каждого фасада, необходимых для стерео-выпрямления. Поэтому в библиотеке используется подход, приближающий исходное изображение к фронтально-параллельному виду. Для этого выполняется аффинная ректификация на изображении, которая восстанавливает параллельность линий в фронтально-параллельном виде изображения.

Выпрямление реализовано в Python с использованием библиотек `scikit-image` и `numpy`.

Работа алгоритма представлена на рисунке 3.2.1 на примере исходного (а) и выпрямленного (б) изображений. На рисунке можно проследить, что вывеска, которая до обработки делит свои вертикальные координаты с окнами второго этажа, после обработки может быть отделена от них прямой линией.

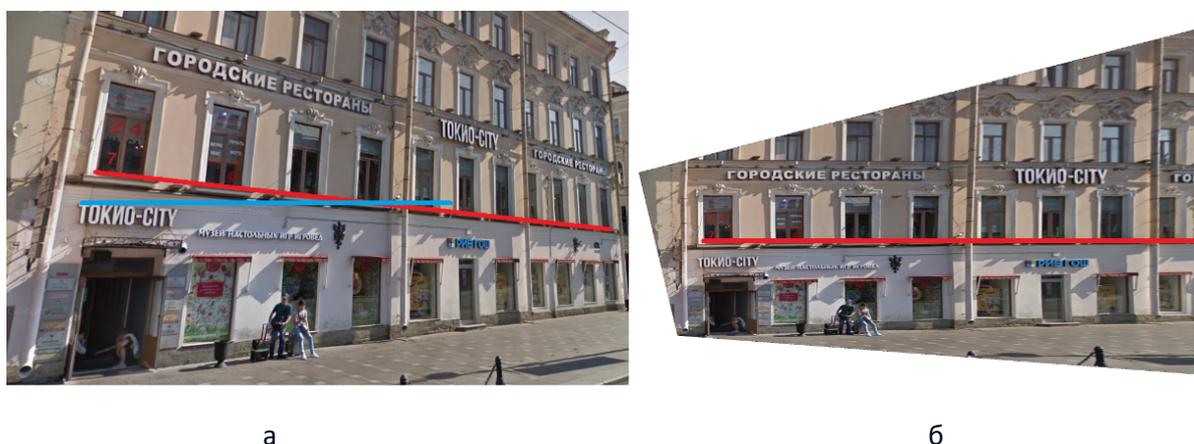


Рисунок 3.2.1 – Пример работы алгоритма выпрямления изображения: до обработки (а) и после обработки (б)

3.3 Сегментация изображения фасада здания

Поскольку ключевой задачей программной библиотеки является поиск подозрительных (незаконных) рекламных и информационных вывесок на фасадах зданий, алгоритм опирается на строгие признаки, по которым может быть сделан вывод о незаконности объекта. Большая часть таких признаков, зафиксированных в законодательстве, описывает ограничения на размер и пространственное размещение вывески на фасаде здания. Это означает, что для принятия решения о соответствии размещённых вывесок законодательству необходимо оценить их расположение относительно других элементов фасада. Для определения расположения необходимо решение задачи сегментации. В рамках задачи сегментации выполняется разбиение фотографии фасада на ключевые элементы: окна, двери, вывески и другие элементы фасада.

Существуют различные подходы к обнаружению объектов, наиболее распространенными из которых являются обнаружение объектов по признакам низкого уровня, метод опорных векторов с использованием признаков более высокого уровня и нейронные сети. Поскольку вывески на фасадах существуют в очень разных формах, невозможно обобщить их внешний вид с помощью разработанных вручную признаков. Самой современной технологией для решения таких задач является применение нейронных сетей. Более медленные модели R-CNN лучше подходят для случаев, когда требуется высокая точность, а время обработки имеет более низкий приоритет. В данной задаче применяется метод TensorFlow. Существует несколько доступных архитектур, которые предварительно обучены на наборе



данных СОСО. Наилучшая производительность была достигнута с архитектурой Inception v2, которая имеет 42 скрытых слоя, используется предварительно обученная модель.

Для целей тестирования был разработан собственный набор данных из 500 фотографий. Вхождения каждого элемента фасада показаны в таблице 3.3.1. Пример иллюстрации фасада, содержащего различные элементы из набора данных, приведён на рисунке 3.3.1.

Таблица 3.3.1 – Частота вхождений элементов фасадов в датасет

Объект	Вхождения
Arch (Арка)	136
Cantilever (Консольная вывеска)	401
Door (Дверь)	520
Signboard (Вывеска)	1413
Window (Окно)	5174

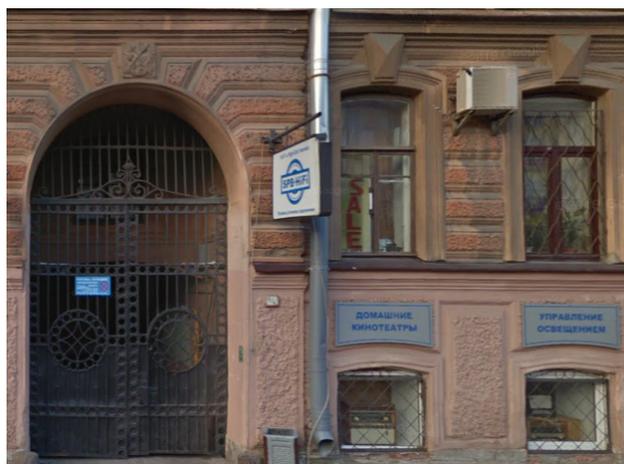


Рисунок 3.3.1 – Фрагмент фасада с 1 аркой, 1 консольной вывеской, 2 вывесками и 4 окнами

Для обработки нейронной сетью используется предварительно выпрямленное изображение. Каждый обнаруженный элемент фасада представляется в виде двух точек на изображении или четырёх пиксельных координат: минимальная и максимальная по горизонтали (x_{min} и x_{max}) и минимальная и максимальная по вертикали (y_{min} и y_{max}).

3.4 Применение комплекса правил для распознавания незаконных элементов фасадов

Правила для определения законности (незаконности) рекламных и информационных вывесок составлены на базе законодательства Санкт-Петербурга. В Санкт-Петербурге действуют четыре региональных закона, которые диктуют условия размещения рекламы на фасадах. На фасаде здания можно разместить только три типа рекламных и информационных объектов: вывески, консольные вывески и наклейки на окнах. В данной версии программной библиотеки реализовано детектирование первых двух типов объектов. Вывески могут быть размещены между первым и вторым этажами здания, они должны оставаться на одном уровне на протяжении одного цельного фасада, они не могут занимать место выше арки, их фон должен быть прозрачным. Консольные вывески могут быть установлены только



в тех случаях, когда на фасаде больше нет места для обычных вывесок. Существуют ограничения по размеру вывесок. Кроме того, для размещения любого рекламного объекта на жилом доме требуется официальное согласие жильцов.

Правило 1. Общий алгоритм детектирования незаконных консольных вывесок.

Если на изображении обнаружена консольная вывеска, то выполняется проецирование всех координат X вывесок на одну линию и вычисляется, сколько пикселей длины изображения осталось не покрытыми вывесками. Если свободно больше пикселей, чем самая короткая вывеска, присутствующая на изображении, или более 30% ширины изображения, то это означает, что консольная вывеска является незаконной, поскольку они разрешены только тогда, когда на фасаде не осталось места для вывески. Пример применения правила 1 показан на рисунке 3.4.1.

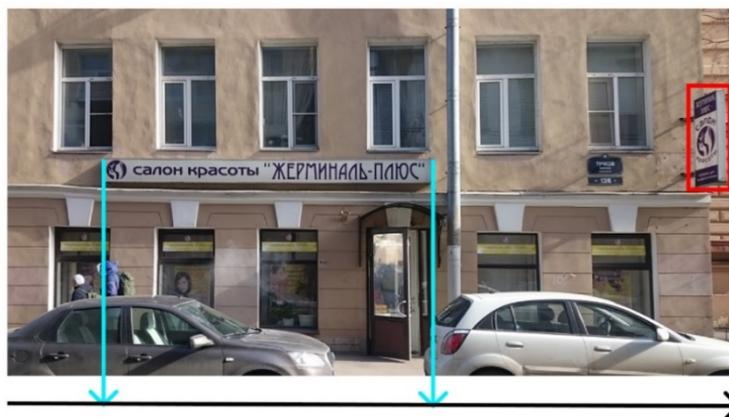


Рисунок 3.4.1 – Иллюстрация правила 1

На фасаде присутствует консольная вывеска и обычная вывеска. Координаты вывесок x_{min} и x_{max} проецируются на линию длиной, равной ширине изображения. Длина вывески составляет 490 пикселей, общая длина линии составляет 1080 пикселей, и есть 590 "свободных" пикселей. Обе проверки правила 1 показывают признаки нарушения допустимых значений, и на основании этого принимается решение, что консольная вывеска является незаконной.

Правило 2. Общий алгоритм детектирования незаконных вывесок, находящихся выше второго этажа.

Для детектирования таких вывесок определяются все объекты с меткой «окно», после чего проецируются все их вертикальные координаты на одну линию в виде двумерных сегментов. Затем определяются точки пересечения, которые разделяют все сегменты на несколько кластеров. Каждый из этих кластеров представляет собой ряд окон на исходном изображении, то есть этаж. Если какой-либо из объектов с меткой «вывеска» имеет координату y выше второй точки этажей, то эта вывеска является незаконной, так как она выше второго этажа здания. На рисунке 3.4.2 изображён случай с двумя вывесками на фасаде, из которых только одна проходит проверку правилом.

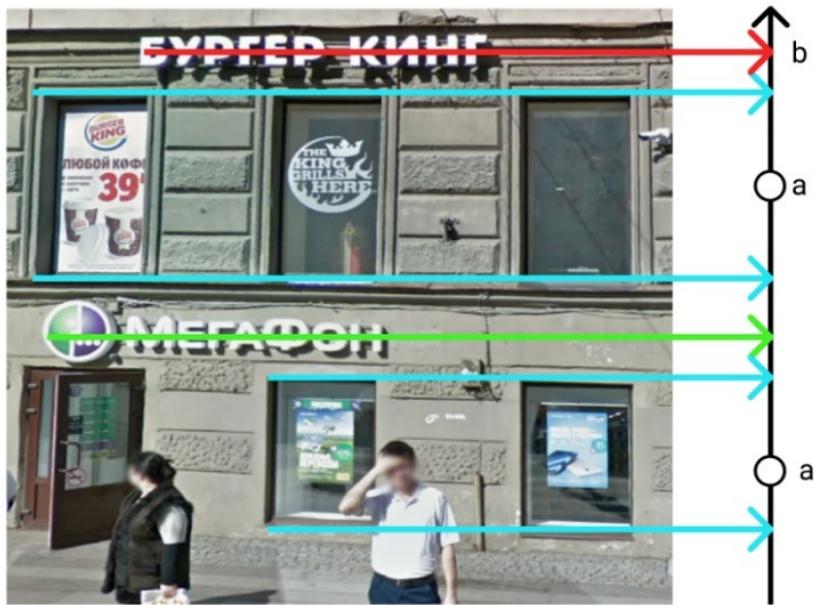


Рисунок 3.4.2 – Иллюстрация правила 2; точки a соответствуют координатам этажей, точка b – вертикальная координата незаконной вывески

Правило 3. Общий алгоритм детектирования незаконных вывесок над арками.

Как и при детектировании соблюдения правила 1, производится проецирование всех координат x вывесок на одну линию, если на изображении обнаружена арка. Если какая-либо координата x вывески вписывается в сегмент арки, это означает, что она находится либо прямо над аркой, либо выше второго этажа. Оба эти условия делают вывеску незаконной. На рисунке 3.4.3 изображён случай с тремя вывесками на фасаде, из которых одна является незаконной.

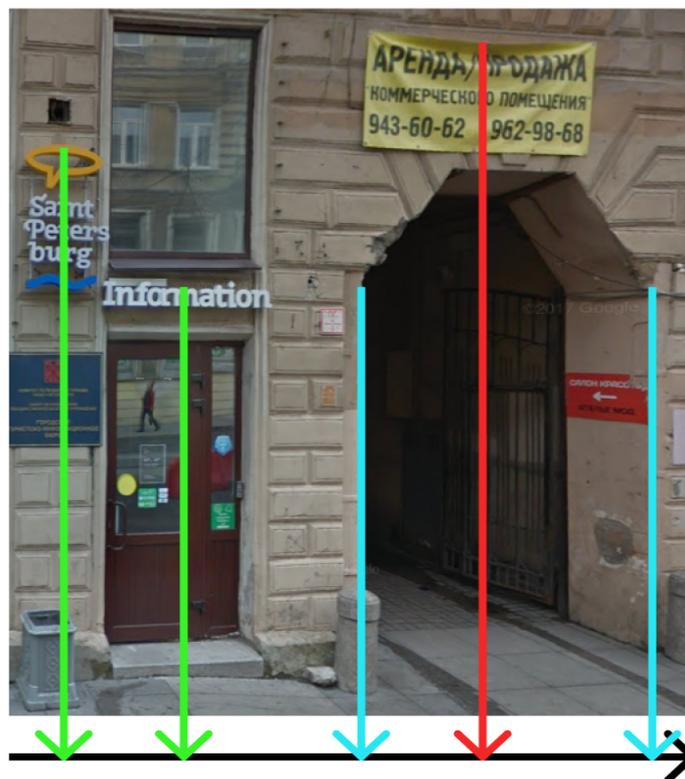




Рисунок 3.4.3 – Иллюстрация правила 3, незаконная вывеска над аркой

Правило 4. Общий алгоритм детектирования незаконных вывесок на разных уровнях.

В рамках реализации алгоритма определяется средняя координата у каждой вывески и детектируется, находятся ли они все в пределах координат у самой маленькой вывески. Если нет, то они расположены не на одном уровне, а это значит, что они противозаконны. На рисунке 3.4.4 изображён пример применения этого правила.



Рисунок 3.4.4 – Иллюстрация правила 4, вывески на разных уровнях

4. Используемые технические средства

Программная библиотека способна выполняться на аппаратных системах со следующими характеристиками:

- архитектура процессора – x86, x86–64;
- минимальный объем оперативной памяти – 1 Гб;
- минимальная тактовая частота процессора – 2,5 ГГц.

Программный модуль требует для своей работы наличия следующего системного программного обеспечения: 32/64-разрядную OS Microsoft Windows 7-8-10, Linux, интегрированная среда разработки Jupyter Notebook. Язык программирования: Python.

5. Настройка программы

Для корректного функционирования программной библиотеки необходимо установить и настроить следующие компоненты:

- интерпретатор Python;
- дистрибутив Anaconda для работы с Jupyter Notebook;
- исходные данные в виде тестового набора данных из 1000 фотографий зданий для обучения нейросети.

Для запуска программной библиотеки автоматизированного распознавания элементов фасадов зданий на основе панорамных изображений улиц необходимо наличие предварительно обученной нейросети на базе тестового набора данных.



6. Выполнение программы

Запуск программной библиотеки производится из интегрированной среды разработки для решения задач машинного обучения и data science – «Jupyter Notebook» путем ввода необходимых параметров и запуска кода библиотеки на языке Python.

Чтобы начать работу с Jupyter Notebook, необходимо установить дистрибутив Anaconda. Anaconda является наиболее широко используемым дистрибутивом Python для работы с данными. Необходимо загрузить последнюю версию Anaconda для Python 3 и установить её, следуя инструкциям на странице загрузки и/или в исполняемом файле.

Jupyter Notebook запускается в качестве графической веб-оболочки для IPython, которая расширяет идею консольного подхода к интерактивным вычислениям. Для запуска Jupyter Notebook необходимо перейти в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке набрать:

```
> ipython notebook
```

В результате будет запущена оболочка в браузере.

Изображение окна с запущенным Jupyter Notebook и созданным новым файлом представлено на рисунке 4.1.

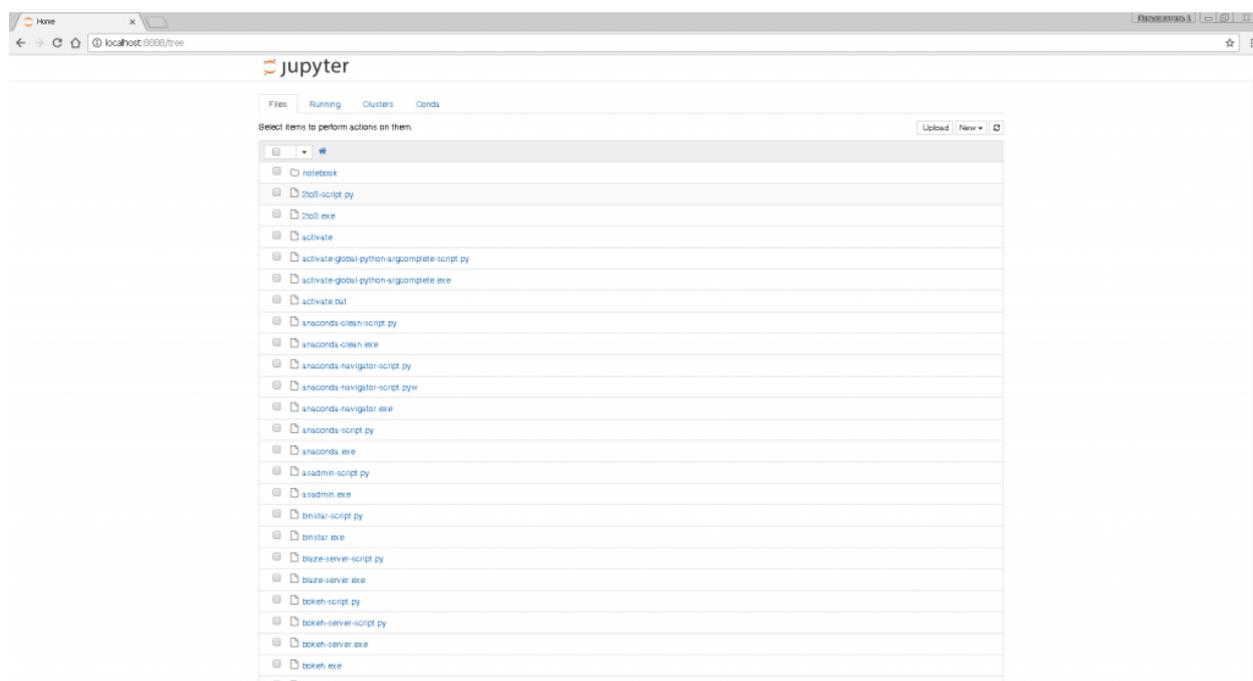


Рисунок 4.1 – Окно Jupyter Notebook в браузере.

Код на языке Python или текст в нотации Markdown нужно вводить в ячейки программной оболочки после создания в программе нового «ноутбука» (записной книжки). При вводе кода на панели инструментов необходимо выставить свойство “Code”.

После загрузки в интегрированную среду разработки программного кода библиотеки необходимо указать путь к изображению, подлежащему анализу, через переменную `image_path` и осуществить запуск программного кода через команду выполнения содержимого ячейки.



После успешного окончания работы библиотеки в подпапке проекта будет создан файл изображения формата png, содержащий выходные данные – модифицированное исходное изображение, нанесённое на координатную плоскость с отмеченными элементами фасадов, вероятностью распознавания элементов и выделенными элементами, нарушающими законодательство. Пример такого изображения представлен на рисунке 4.2.



Рисунок 4.2 – Изображение, полученное в результате обработки исходного изображения программной библиотекой.

7. Входные данные

Входные данные для программной библиотеки представляют собой изображения фасадов. Рекомендованное разрешение изображений 1920x1080. Модель является хорошо обобщённой для окон и вывесок, что является результатом сильного их преобладания в обучающем наборе данных. Некоторые консольные вывески и арки при обработке могут быть пропущены.

8. Выходные данные

Выходными данными программной библиотеки являются файлы изображений, содержащие разметку выявленных элементов фасада с оценкой точности определения.

Результаты тестов для 50 изображений с 363 окнами, 100 вывесками, 30 дверями, 18 консолями и 5 арками приведены в таблице 7.1. Все 10 ложноположительных результатов были ложными обнаружениями окон. Согласно tensorboard (инструмент TensorFlow), средняя точность для всех 5 классов составляет 0,59.



Таблица 7.1. Результаты тестов

Объект	Recall
Arch	0.60
Cantilever	0.72
Door	0.77
Signboard	0.76
Window	0.96

Пример обработанного изображения, полученного в качестве выходных данных, приведён на рисунке 7.1.

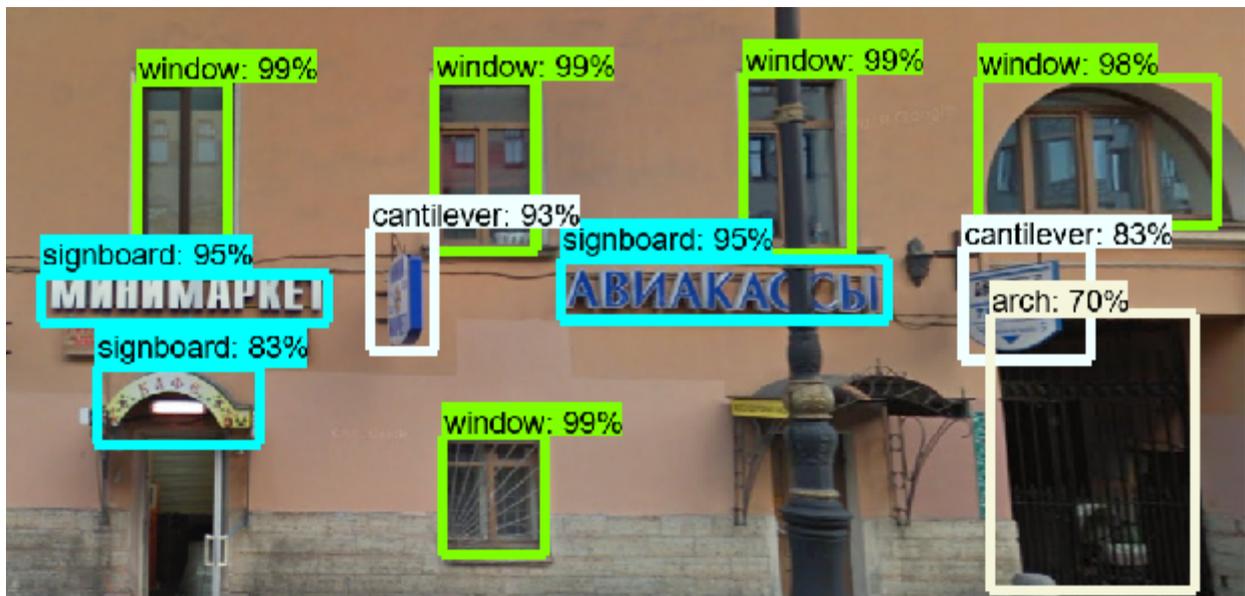


Рисунок 7.1 – Пример изображения, полученного в качестве выходных данных