

Оценка людского и транспортного трафика густонаселенных районов города Москвы с использованием видео со стационарных камер

Р. М. Воробьёв

Аннотация— В данной работе рассматривается задача получения информативных данных с использованием видеоаналитических подходов, основанных на компьютерном зрении и глубоких нейронных сетях, а также их дальнейшей аналитической оценки с целью автоматизации процесса подсчета трафика в густонаселенных районах Москвы, актуального для Правительства Москвы. Предлагается подход к решению данной задачи, позволяющий осуществить подсчёт в выбранной локации двух различных трафиков: людского и автомобильного. Создаётся набор данных, необходимый для процедуры обучения нейронной сети. Проводится оценка точности алгоритма, путём измерения меры качества детектора.

В работе рассматривается задача определения (выявления) каждого отдельно взятого объекта (человека и транспортного средства) и дальнейшего анализа его движения. Она заключается в том, что для заданного видеofайла требуется определить количество людей, транспортных средств, движущихся в одном из нескольких заданных направлений. В дальнейшем эти данные анализируются (соотносятся с временными промежутками, помогают выявить «популярные направления», «горячие зоны» (зоны повышенного интереса)), и на их основе строятся графики и тепловые карты. Эта задача относится к области детекции объектов на изображении.

Ключевые слова—Детекция объектов, видеоаналитика, компьютерное зрение, создание набора данных.

I. ВВЕДЕНИЕ

На данный момент для решения задачи оценки трафика используются специально нанятые люди, которые отсматривают огромный набор видеозаписей. Из минусов существующего решения задачи оценки трафика можно указать следующие: 1) требуется участие человека (1.1: необходимо нанять большое количество людей; 1.2: большая погрешность при подсчёте огромного потока людей); 2) огромные временные затраты (видеоматериалов много, а их обработка требуется каждый день). В связи с чем появляется острая необходимость в автоматизации данного

процесса путем полного исключения человека из решения данной задачи.

На сегодняшний день сфера компьютерного зрения активно развивается, в том числе и в направлении решения подобных задач. Методы с использованием нейросетевых подходов так же получают всё больший охват по использованию в различных сферах и успешно демонстрируют хороший результат.

На основании всего вышесказанного можно сделать вывод, что рассматриваемая проблема ещё не решена, но имеет направления развития, следуя которым, можно ускорить и улучшить процесс аналитики.

II. ОБЗОР МЕТОДОВ В ДАННОЙ ОБЛАСТИ

Так как основной проблемой для данной задачи является детекция объектов на входном изображении, поиск методов происходил именно по данной теме для дальнейшего составления базового алгоритма.

Для детекции, обычно, применяется один из двух основных подходов (см. [18], [19]): 1) Метод, опирающийся на качественные характеристики детектируемого объекта (цвет, размер, форма и т.д.); 2) Метод, базирующийся на нейронных сетях

Суть подхода, использующего качественные характеристики, заключается в том, что заранее известны какие-то важные и отличительные черты искомого объекта. То есть от того, что известно, будет зависеть и сам алгоритм поиска.

Наиболее часто используются алгоритмы, основанные на поиске краев объектов на изображении и дальнейший их анализ с данными, что отвечают за параметры искомого объекта. К примеру, если известен примерный размер детектируемого предмета, то если отыскать края всех объектов на изображении, то можно будет предположить какой у них размер и найти наиболее подходящий критерию поиска.

Поиск краев осуществляется путем оценки сигнала яркости. То есть на границах объекта, скорее всего, будет присутствовать перепад яркости. Именно данные перепады и анализируются в базовых подходах. Для нахождения перепадов используется алгоритм свертки (часто оператором Собеля), то есть по всей матрице изображения проходятся «маской», которая

является матрицей с элементами, представляющими из себя весовые коэффициенты разных знаков. После прохождения «маской» на выходе получается матрица, состоящая из приближенных производных по одной из осей (зависит от того, какой маской пройтись по изображению). Далее находится квадратный корень из суммы квадратов элементов полученных матриц и формируется финальная матрица, которая и подлежит дальнейшему анализу, а именно - на краях объектов будет заметен высокий перепад. Данные подходы подробнее описаны в алгоритмах [3], [4], [5], [8], [9], [10].

Альтернативным алгоритмом является анализ матрицы изображения на основе цветовой составляющей. То есть если заранее известен цвет искомого объекта и известно, что он будет отличаться от фона, можно произвести поиск пикселей, лежащих в определенном диапазоне и параллельно заносить в матрицу единицу, если пиксель принадлежит диапазону, и ноль - если нет. Таким образом получается матрица, где остаются только объекты с заданным цветом, и далее можно по примерным оценкам размеров оставшихся объектов найти искомым.

В качестве альтернативы для описанных выше подходов следует рассмотреть метод, базирующийся на нейронных сетях. В современных подходах, в связи с активным развитием технической составляющей, всё большую популярность набирают методы, базирующиеся на глубинном обучении. Особенность данного метода заключается в том, что нейронные сети можно «затачивать» под различные и, в то же время, конкретные задачи.

С постепенным развитием данной сферы сложность построения нейронных сетей сводится к задаче нахождения правильной последовательности слоев и коэффициентов для них. То есть, по сути, современные нейронные сети представляют из себя «правильно собранный конструктор», которому необходимо «подать» набор данных, на котором он сможет обучиться и, впоследствии, выдавать прогнозы в качестве ответа на поставленный вопрос.

В ходе поиска алгоритмов, подходящих в качестве базового, была найдена статья, в которой был составлен топ современных нейросетевых подходов, позволяющих решить определенную задачу, на основе скорости их работы, результативности и востребованности в исходных тренировочных данных. [23]

Наиболее примечательными в сфере детекции объектов оказались R-CNN, смысл которых заключается в разбиении входного изображения на большое количество регионов, каждый из которых подвергается классификации. Далее, наиболее подходящий регион к искомому объекту считается задетектированным объектом.

В статье были выделены две сети - YOLO и SSD, а также приведено их сравнение.

A) SSD

Одним из методов обнаружения объектов на изображении с использованием одной глубокой

нейронной сети является SSD. В данном подходе авторы предлагают дискретизировать пространство с ограничивающими прямоугольниками в набор полей для разных пропорций и масштабов объектов. При каждом вызове сеть генерирует процент наличия каждого объекта внутри каждого поля и производит корректировки размера этого поля, чтобы лучше соответствовать формам объектов. Кроме того, сеть объединяет прогнозы нескольких карт функций различных разрешений для более качественного определения объектов разного размера.

SSD, в связи с тем, что он полностью исключает создание предположений и последующие этапы передискретизации пикселей (функций) и инкапсулирует все вычисления внутри одной сети, является простым для сборки интеграции в системы, требующие компонента обнаружения объектов, относительно других методов, требующих предположений для объектов.

Экспериментальные результаты на различных наборах данных (PASCAL VOC, COCO) дают понять, что SSD имеет конкурентоспособную точность для методов, которые используют дополнительный шаг предположения объекта, и, при этом, осуществляет подсчет намного быстрее, обеспечивая единую структуру как для обучения, так и для тестирования. Так же SSD хорош в применении к изображениям с маленьким исходным размером.

Более подробно ознакомиться с SSD можно в статье [24].

B) YOLO

Альтернативой для сети SSD может послужить сеть YOLO [15]. По словам авторов статьи [14], YOLO v2 являлась самой современной (state-of-the-art) на тот момент моделью, для задачи обнаружения объектов на изображении и была протестирована на известных наборах данных (PASCAL VOC, COCO), показав при этом отличный результат по показателю точности, превзошедший альтернативные подходы (Faster R-CNN и SSD), при этом ещё и выигрывая по времени обработки.

Одним из главных плюсов данной сети является ее многомасштабный метод обучений, что позволяет использовать модель YOLO для различных размеров исходного изображения, при этом не теряя в скорости обработки и точности.

Вторым, и не менее важным плюсом YOLO является то, что в этой сети реализован метод совместного обучения как обнаружения объектов, так и их классификации. То есть данная сеть может спрогнозировать на изображении местоположение объекта, который отсутствовал в разметке набора данных, используемого для детектора.

Третьим важным плюсом этой сети является её скорость обработки. Вся процедура нахождения и классификации объектов на исходном изображении настолько велика, что позволяет обрабатывать видеофайлы с 25 FPS в режиме реального времени.

На основе изученной литературы в работе было решено разрабатывать и реализовывать алгоритмы с включением базовых подходов, позволяющих решить поставленную задачу. Вызвано это тем, что проблема остается далеко не тривиальной и требует определенного подхода. Также требуется разработка полноценного функционала, который позволит полностью решить поставленную задачу.

Из рассмотренных ранее сетей, опираясь на сравнительные характеристики, было решено выбрать к рассмотрению нейронную сеть YOLO, так как одним из главных преимуществ, по отношению к другим сетям, является её скорость обработки (поддержка обработки видео в режиме реального времени). Так же её плюсом является масштабируемость (то есть отсутствие привязки к размерам изображений в обучающей выборке). И, наконец, возможность определения мелких объектов, благодаря использованию якорей.

Таким образом для решения поставленной задачи потребуется разработать алгоритм, который будет использовать в себе изученные базовые подходы, и переделать рассмотренную сеть Darknet Yolo под данную задачу (подобрать коэффициенты, сформировать набор данных, переобучить нейросеть, протестировать её на тестовой выборке).

III. СОЗДАНИЕ НАБОРА ДАННЫХ

Для выполнения процедуры обучения нейронной сети возникла необходимость создания собственного набора данных. Произошло это в следствие возникновения проблем с качеством детекции объектов на уже существующих наборах данных, взятых с интернета и используемых в качестве обучающей выборки для нейросети. В связи с чем было решено создать новый набор данных.

В качестве изображений использовались фотографии, взятые (вырезанные) из видеопоследовательностей с камер, на которых необходимо осуществлять подсчет. Бралась изображения с различным ракурсом и разрешением камеры для большего разнообразия данных.

Для создания необходимой базы следовало на изначальных изображениях разметить все отдельно взятые транспортные средства и людей своим конкретным классом. То есть в конечном итоге ожидалось сформировать набор изображений из двух классов.

В качестве базового аппаратного средства по разметке объектов использовался скрипт LabelImg, взятый из Интернета и, впоследствии, промодифицированный для более удобной и быстрой разметки, основанной на требованиях конкретной задачи.

Данный скрипт позволяет подгружать последовательно из директории изображения, производить их разметку путем выделения прямоугольного бокса и сохранять размеченные координаты в виде xml-файла. Так как на вход нейросети Yolo нужно подавать текстовый файл с координатами центров размеченных боксов,

потребовалось написать скрипт, который бы выполнял операцию преобразования информации с xml-файла к требуемому формату.

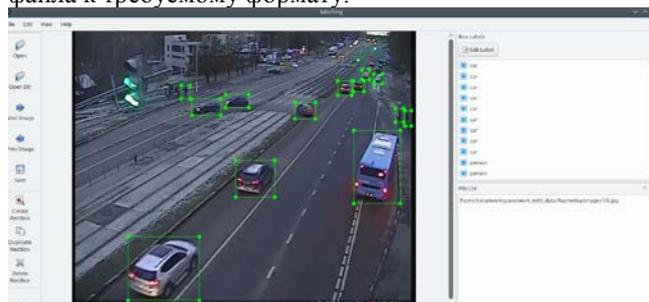


Рис. 1. Разметка набора данных в приложении LabelImg.

Таким образом проводилась разметка набора данных, необходимого для нейронной сети Darknet.

IV. ПРЕДЛОЖЕННЫЙ АЛГОРИТМ

Предложенный алгоритм содержит четыре основных шага:

- Шаг1: определение местоположения всех отдельно взятых объектов (транспорт, человек).
- Шаг2: оценка количества движущихся объектов в заданных зонах (трекинг движения всех объектов) с помощью анализа последовательных кадров и данных о местоположении.
- Шаг3: подсчет трафика для каждого из указанных направлений (связей зон «начало-конец движения»).
- Шаг4: оценка данных Шага №3 и формирование «горячих зон» и графиков движения.



Рис.2. Краткая иллюстрация алгоритма.

V. ОПИСАНИЕ ШАГОВ

Шаг №1:

Было испробовано множество подходов к сегментации объектов на изображении, но самым лучшим из них оказался метод сегментации с

использованием градиентов. Вызвано это тем, что объекты, как правило, имеют одинаковый цвет (в том числе и яркость пикселей на ней) и форму, следовательно, для поиска людей требуется отыскать на изображении одноцветные небольшие в ширину и продолжительные в высоту фигуры, которые и будут являться людьми. Из данных условий, градиент по горизонтальной и вертикальной координатам должен сильно меняться в местах «краёв» человека. Для транспортных средств всё то же самое, только формы будут намного больше людских. Таким образом проведётся сегментация изображения. Все подходящие под данные условия объекты проходят дальнейшую проверку на ложные срабатывания (если ширина объектов сильно меньше или, наоборот, больше, чем определенный процент для каждого из классов ширины и высоты изображения, то такие объекты игнорируются).

На данном Шаге возникает очень большая погрешность по отношению к подсчету людей в связи с тем, что они могут стоять очень близко друг к другу (случай толпы) и, следовательно, будут просегментированы как один объект, что неверно.

Чтобы избежать подобных проблем был испробован подход, в котором просегментированные объекты отправлялись «на уточнение» в детектирующую нейронную сеть, которая, в свою очередь, должна определить каждый отдельно взятый объект (человек в толпе или автомобиль среди огромного транспортного потока).

Но он так же оказался неэффективным в связи с неверной сегментацией объектов. К примеру, иногда человек попадал в два различных сегмента, что приводило к тому, что, после уточнения, результат удваивался, что вызывало сильную погрешность.

Чтобы избавиться от проблем ошибочной сегментации был предложен альтернативный подход к определению местоположения отдельно взятых объектов с использованием детектирующей нейронной сети YOLO на исходном изображении. То есть переложить задачу детекции объектов полностью на нейронную сеть. Она должна на исходном изображении найти все предобученные классы (объекты).

Для корректного решения поставленной задачи потребуется собранный ранее набор данных (люди, транспорт).

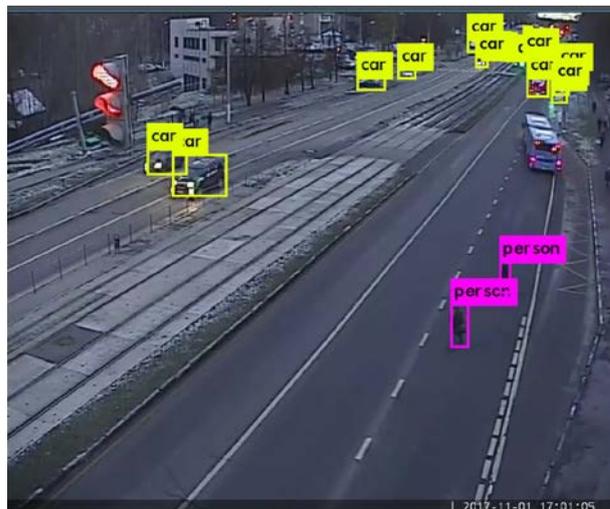


Рис.3. Детекция объектов.

Шаг №2:

На данном этапе имеются координаты найденных объектов обоих классов (транспорт, человек).

Теперь необходимо осуществить трекинг данных объектов с использованием анализа данных, полученных с последовательных кадров. То есть произвести связь координат, принадлежащих одному движущемуся объекту внутри каждой из заданных зон, в последовательность.

Данные связи образуются путем поккадровой оценки евклидовых расстояний между задетектированными объектами. То есть для каждого найденного объекта на текущем кадре происходит поиск близлежащих координат местоположения объекта на предыдущем кадре, используя ограничивающие условия, и, если таковые найдены - происходит связь этих координат объектов в единый вектор.

Таким образом для каждого объекта формируется собственный вектор движения внутри заданных зон (то есть начальную и конечную координаты).

Далее, происходит подсчёт этих векторов для всех указанных зон, что в конечном итоге даст понятие о количестве объектов, пересёкших данную область.



Рис.4. Разметка двух локаций на зоны, в которых происходит подсчёт треков.

Шаг №3:

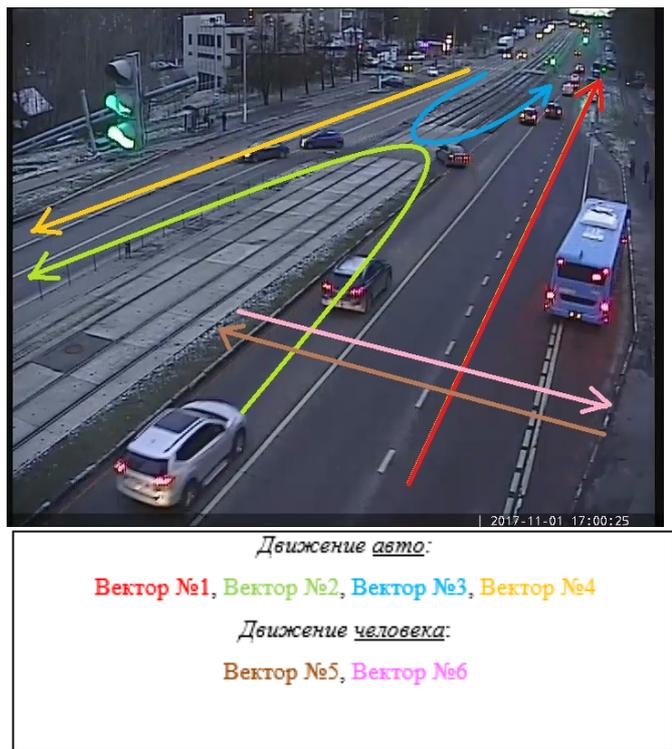


Рис.5. Направления движений объектов

Благодаря априорной информации о количестве начальных и конечных зон, а также их связях, происходит распределение трафика между областями. То есть строится система из уравнений, в которой в качестве неизвестных переменных выступают количественные характеристики движущихся вдоль заданных направлений объектов.

Для примера рассмотрим 4 направления движений автомобилей (Рис.5 – v1, v2, v3, v4), 2 начальных зоны (Рис.4, локация №1 – start_1, start_2), 2 конечных зоны (Рис.4, локация №1 – end_1, end_2) и 1 промежуточная (Рис.4, локация №1 – middle).

Рассмотрим систему уравнений из всех возможных направлений движения. Получаем количество автомобилей, которые двигались вдоль каждого из заданных направлений:

$$\begin{cases}
 v1+v2 = start_1 & 1100 & start_1 & \rightarrow & 1000 & (start_1+end_1+middle)/2 \\
 v1+v3 = end_1 & 1010 & end_1 & \rightarrow & 0010 & (end_1-start_1+middle)/2 \\
 v3+v4 = start_2 & 0011 & start_2 & \rightarrow & 0001 & (start_2+end_2+middle)/2 \\
 v2+v4 = end_2 & 0101 & end_2 & \rightarrow & 0100 & (end_2-start_2+middle)/2 \\
 v2+v3 = middle & 0110 & middle & \rightarrow & & &
 \end{cases}
 \Rightarrow
 \begin{cases}
 v1 = (start_1+end_1+middle)/2 \\
 v2 = (end_2-start_2+middle)/2 \\
 v3 = (end_1-start_1+middle)/2 \\
 v4 = (start_2+end_2+middle)/2
 \end{cases}$$

В случае, если направлению движения соответствуют только одна начальная и одна конечная зоны (Рис.4, локация №2) – следует разметить только одну зону, через которую и будет происходить движение объекта.

Таким образом количество объектов, пересекших данную зону, будет соответствовать количеству объектов, двигавшихся вдоль данного направления

Шаг №4:

На конечном этапе имеется набор из векторов, описывающих направление движения всех обнаруженных объектов (людей, автомобилей). С помощью данной информации появляется возможность построить два отдельных графика (для

людей и для транспорта), которые будут объединять в себе данные за заданный промежуток времени и отображать по одной оси количественную характеристику прошедших людей (проехавших автомобилей) вдоль каждого из направлений, заданных на начальном этапе, а другая ось будет отвечать за каждое из заданных направлений соответственно.

Так же, используя информацию из Шага №3, можно определить «горячие зоны» в локации, за которой идет наблюдение с видеокamеры. А именно, можно за указанный промежуток времени проанализировать данные для различных направлений движения и выявить среди них наиболее популярные.

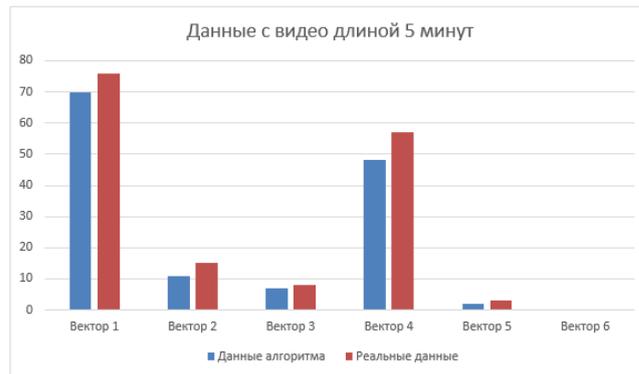


Рис.6. График трафика за 5 минут

Таким образом, наиболее популярными направлениями движения для автомобилей являются №1 и №4, а для людей №5.

VI. ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ДЕТЕКТОРА

Для оценки полноты и точности работы детектора понадобятся текстовые файлы, полученные в ходе экспериментов на изображениях тестовой выборки N (test_N, где N – номер изображения, которому принадлежит данный текстовый файл) и текстовые файлы с заранее размеченными координатами объектов, которые принадлежат уже размеченным классам (true_N, где N – номер изображения, которому принадлежит данный текстовый файл).

Для оценки точности реализованных алгоритмов был написан скрипт на языке Python.

1. Суть скрипта заключается в том, что в цикле от 1 до N считываются текстовые файлы test_N и true_N.
2. Далее, в цикле, идёт работа со строками файла true_N - преобразовываются все координаты боксов к координатам центра этого бокса и заносятся в словарь по ключу «Номер класса» координаты этого центра. Таким образом получается словарь, состоящий из координат центров всех боксов объектов, содержащихся на исходном изображении, которые должна определить наша нейронная сеть / базовый алгоритм.
3. Далее, необходимо пройтись по строкам файла test_N. Если достигнут конец файла, то подсчитывается количество оставшихся

координат в списке и фиксируется каждый список координат как пропущенный объект (++FN). Происходит переход на следующий шаг. Иначе, таким же образом преобразовываются координаты, и по ключу номера класса происходит поиск в созданном на предыдущем шаге списке координаты. Если такой ключ имеется, то сравниваются координаты. Если отличие между заранее известными координатами и теми, что выдал алгоритм, меньше чем в 20 пикселей – считается, что объект определен верно (++TP), удаляются эти координаты из списка. Если же отличие больше, чем в 20 пикселей, или такого ключа нет, то это засчитывается как ложное срабатывание (++FP). Далее следует возвращение на начало данного шага.

4. На данном шаге имеется количество срабатываний для одного изображения, а именно: TruePositive, FalsePositive, FalseNegative, что позволяет оценить точность (precision) и полноту (recall) полученного результата для конкретного изображения:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN}$$

5. Далее, находится F-мера:

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}$$

6. Данное значение добавляется в конец списка, и происходит возвращение в начало цикла.

Таким образом получается список из N F-мер. Далее, следует рассчитать среднее значение:

$$F_{avg} = \frac{F[0] + \dots + F[N-1]}{N}$$

Это значение и являлось ключевым в оценке качества детекции.

Идентификатор локации	Время видео (мин)	Средняя F-мера (Точность)
1	1	0.96
1	5	0.95
2	5	0.98

Рис.7. Результаты оценки детектора на тестовой выборке.

VII. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШЕЕ ИССЛЕДОВАНИЕ

В ходе данной работы были решены следующие задачи:

1. Проведен обзор существующих методов выделения и классификации объектов на изображениях. Среди них была позаимствована и модифицирована нейронная сеть Darknet Yolo.
2. Был сформирован и размечен набор данных, состоящий из изображений с камер, на которых присутствуют автомобильные средства и люди.
3. Был разработан и реализован базовый алгоритм, который на входной видеопоследовательности для указанных зон произведёт подсчёт

движущихся объектов, распределит данные по указанным связям-направлениям и построит график трафика.

4. Была произведена экспериментальная оценка работы детектора.

В качестве направлений для дальнейшего развития метода предлагается:

1. Изменение конфигурации нейронной сети Yolo с целью увеличения скорости работы алгоритма.
2. Использование сегментирующих сетей, которые будут обучены так же на двух классах (человек, транспорт). С помощью данного подхода станет возможна точная детекция групп объектов (толпы людей и скопления машин). А далее можно будет использовать детектирующую нейронную сеть Yolo.

БИБЛИОГРАФИЯ

- [1] Haykin S., Network N. A comprehensive foundation //Neural Networks. – 2004. – Т. 2. – №. 2004. – С. 41.
- [2] Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – С.1097-1105.
- [3] Girshick R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2014. – С.580-587.
- [4] Viola P., Jones M. Rapid object detection using a boosted cascade of simple features //Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. – IEEE, 2001. – Т. 1. – С. I-I.
- [5] Viola P., Jones M. Robust real-time object detection //International Journal of Computer Vision. – 2001. – Т. 4. – №. 34–47.
- [6] Szegedy C. et al. Going deeper with convolutions //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – С. 1-9.
- [7] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.
- [8] Opelt A. et al. Weak hypotheses and boosting for generic object detection and recognition //Computer vision-ECCV 2004. – 2004. – С.71-84.
- [9] Lienhart R., Maydt J. An extended set of haar-like features for rapid object detection //Image Processing, 2002. Proceedings. 2002 International Conference on. – IEEE, 2002. – Т. 1. – С. I-I
- [10] Dalal N., Triggs B. Histograms of oriented gradients for human detection //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – Т. 1. – С. 886-893.
- [11] Suard F. et al. Pedestrian detection using histograms of oriented gradients //Intelligent Vehicles Symposium, 2006 IEEE. – IEEE, 2006. – С.206-212.
- [12] Lee H., Center S. Multiple Object Class Detection & Localization with Deep Learning (CNN). – 2015.
- [13] Buhler K., Lambert J., Vilim M. Real-time Object Tracking in Video CS 229 Course Project.
- [14] Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger //arXiv preprint arXiv:1612.08242. – 2016.
- [15] Redmon J. et al. You only look once: Unified, real-time object detection //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С.779-788.
- [16] Girshick R. et al. Detectron. – 2018.
- [17] Redmon J., Farhadi A. YOLOv3: An incremental improvement //arXiv preprint arXiv:1804.02767. – 2018.
- [18] CV-Tricks. Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD. cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/ 2018
- [19] Object Detection with Deep Learning: The Definitive Guide. tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning/ 2018

- [20] Girshick R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2014. – C.580-587.
- [21] Uijlings J. R. R. et al. Selective search for object recognition//International journal of computer vision. – 2013. – T. 104. – №. 2. – C. 154-171.
- [22] Ren S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks //Advances in neural information processing systems. – 2015. – C.91-99.
- [23] One-shot object detection. machinethink.net/blog/object-detection/2018
- [24] Liu W. et al. Ssd: Single shot multibox detector //European conference on computer vision. – Springer, Cham, 2016. – C.21-37.

Assessment of human and transport traffic in densely populated Moscow areas using video from stationary cameras

R. Vorobyev

Abstract—In this paper, the task is to obtain informative data using video analytic approaches based on computer vision and deep neural networks, as well as their further analytical evaluation in order to automate the process of calculating traffic in Moscow's densely populated areas. An approach to solving this problem is proposed, allowing to calculate in the selected location two different traffic: human and automobile. A data set is needed for the neural network learning procedure. The accuracy of the algorithm is evaluated by measuring the quality measure of the detector.

The paper deals with the task of determining (identifying) each individual object (person and vehicle) and further analyzing its movement. It lies in the fact that for a given video file is required to determine the number of people, vehicles moving in one of several predetermined directions. In the future, these data are analyzed (correlated with time intervals, help to identify "popular destinations", "hot zones" (zones of increased interest)), and on the basis of them build graphs and heat maps. This task relates to the field of detection of objects in the image.

Keywords—Object detection, video analytics, computer vision, dataset creation.

REFERENCES

- [1] Haykin S., Network N. A comprehensive foundation //Neural Networks. – 2004. – T. 2. – №. 2004. – C. 41.
- [2] Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – C.1097-1105.
- [3] Girshick R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2014. – C.580-587.
- [4] Viola P., Jones M. Rapid object detection using a boosted cascade of simple features //Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. – IEEE, 2001. – T. 1. – C. I-I.
- [5] Viola P., Jones M. Robust real-time object detection //International Journal of Computer Vision. – 2001. – T. 4. – №. 34-47.
- [6] Szegedy C. et al. Going deeper with convolutions //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – C. 1-9.
- [7] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.
- [8] Opelt A. et al. Weak hypotheses and boosting for generic object detection and recognition //Computer vision-ECCV 2004. – 2004. – C.71-84.
- [9] Lienhart R., Maydt J. An extended set of haar-like features for rapid object detection //Image Processing. 2002. Proceedings. 2002 International Conference on. – IEEE, 2002. – T. 1. – C. I-I
- [10] Dalal N., Triggs B. Histograms of oriented gradients for human detection //Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. – IEEE, 2005. – T. 1. – C. 886-893.
- [11] Suard F. et al. Pedestrian detection using histograms of oriented gradients //Intelligent Vehicles Symposium, 2006 IEEE. – IEEE, 2006. – C.206-212.
- [12] Lee H., Center S. Multiple Object Class Detection & Localization with Deep Learning (CNN). – 2015.
- [13] Buhler K., Lambert J., Vilim M. Real-time Object Tracking in Video CS 229 Course Project.
- [14] Redmon J., Farhadi A. YOLO9000: Better, Faster, Stronger //arXiv preprint arXiv:1612.08242. – 2016.
- [15] Redmon J. et al. You only look once: Unified, real-time object detection //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – C.779-788.
- [16] Girshick R. et al. Detectron. – 2018.
- [17] Redmon J., Farhadi A. YOLOv3: An incremental improvement //arXiv preprint arXiv:1804.02767. – 2018.
- [18] CV-Tricks. Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD. cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/ 2018
- [19] Object Detection with Deep Learning: The Definitive Guide. tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning/ 2018
- [20] Girshick R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2014. – C.580-587.
- [21] Uijlings J. R. R. et al. Selective search for object recognition//International journal of computer vision. – 2013. – T. 104. – №. 2. – C. 154-171.
- [22] Ren S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks //Advances in neural information processing systems. – 2015. – C.91-99.
- [23] One-shot object detection. machinethink.net/blog/object-detection/ 2018
- [24] Liu W. et al. Ssd: Single shot multibox detector //European conference on computer vision. – Springer, Cham, 2016. – C.21-37.