

ДЕТЕКТИРОВАНИЕ ДВИЖУЩИХСЯ ОБЪЕКТОВ В ВИДЕОПОТОКЕ НА ПРИМЕРЕ КОНТРОЛЯ УЛИЧНОГО ДВИЖЕНИЯ

Волковский Евгений Станиславович

*Гродненский государственный университет им. Янки Купалы,
факультет математики и информатики,
кафедра системного программирования и компьютерной безопасности
студент 4 курса специальности «Компьютерная безопасность»*

Научный руководитель Кадан Александр Михайлович, к.т.н., доцент

В работе развивается концепция детектирования движущихся объектов в видеопотоке на примере контроля уличного движения. Очевидно, что ручной подсчёт в задаче детектирования движущихся объектов в видеопотоке является весьма трудоёмким. Однако, несмотря на это уже теперь доступны системы, которые будут автоматически анализировать видеоизображения, обнаруживая машины с точностью, схожей с точностью людей, смотрящих видео. Одним из главных направлений использования системы подсчёта и анализа дорожного трафика в современных условиях является помощь в конструировании дорожной сети. Например, это нужно для корректировки расписания дорожного транспорта (автобусов и троллейбусов) в различных условиях.

Ключевые слова: компьютерное зрение, распознавание транспорта, распознавание машин, обнаружение транспорта, обнаружение машин, анализ машинного трафика.

Очевидно, что ручной подсчёт в задаче детектирования движущихся объектов в видеопотоке является весьма трудоёмким. Однако, несмотря на это уже теперь доступны системы, которые будут автоматически анализировать видеоизображения, обнаруживая машины с точностью, схожей с точностью людей, смотрящих видео.

Подобный метод подсчёта транспортных средств имеет несколько преимуществ по сравнению с другими автоматическими системами. Это экономически более выгодно, так как, детектировать объекты можно для нескольких полос или выходов, а на перекрёстках нужна только одна камера.

Подсчёт легко проверить, просто посмотрев видео.

Одним из главных направлений использования системы подсчёта и анализа дорожного трафика в современных условиях является помощь в конструировании дорожной сети. Например, это нужно для корректировки расписания дорожного транспорта (автобусов и троллейбусов) в различных условиях. Например, зная статистику проезжающих машин, можно составить граф, где рёбра будут представлять собой степень загруженности в определённый момент времени. При более близком анализе подобного графа может стать возможным построение расписания с наименьшими временными задержками. Так же подобная статистика, например, может помочь в составлении маршрутов такси, уменьшив тем самым время продвижения к заданной точке, которой может являться как заказчик, так и пункт его назначения.

Использование сети «ResNet»

ResNet — это сокращение от *Residual Network* (дословно — «остаточная сеть»). Данная сеть появилась в ходе эволюций нейронных сетей. Причина в том, что на некотором моменте их развития, результативность обучения сети в зависимости от количества слоёв перестала расти. Т.е. сеть с меньшим количеством слоёв могла решать задачи лучше, чем сеть, в которой этих слоёв больше. Однако, толчком к развитию ResNet стало наблюдение, что данной проблемы можно избежать, если просто пропускать сигнал дальше без изменений. Слой, который без изменений передаёт сигнал дальше называется *identity layer*

Следует добавить, что *identity layers* в данной сети несколько изменяют алгоритмы её обучения. Так же, что не мало важно, уже существует обученная нейросеть ResNet с заранее установленными весами, где обучение проводилось на датасете ImageNet.

ImageNet — это набор данных миллионов помеченных изображений с высоким разрешением, относящихся примерно к 22 тысячам категорий. Изображения были собраны из Интернета и помечены людьми.

Алгоритм применения данной сети оказался весьма прост, а также сама сеть показала очень хороший результат в вопросе распознавания. Однако наибольшим недостатком в контексте решаемой задачи было долгое время одного «предсказания».

Использование сети «YOLOv3»

Данная модель существенно отличается от привычной нейронной сети. Обычно системы применяют модель к изображению в нескольких местах и масштабах. Области с высокой оценкой считаются как обнаруженные объекты. Однако, в модели «YOLOv3» используется совершенно другой подход. Здесь применяется одна нейронная сеть для полного изображения. Эта сеть делит изображение на регионы и предсказывает ограничивающие рамки и вероятности для каждого региона. Эти ограничивающие рамки взвешиваются по прогнозируемым вероятностям. А т.к. модель просматривает все изображение во время тестирования, её прогнозы основываются на общем контексте изображения. Она также делает прогнозы с единой оценкой сети в отличие от иных систем.

Логика работы программы

При решении данной задачи, логика программы состояла из следующих частей, которые можно как использовать вместе (в режиме реального времени), так и разделить:

- 1) Обнаружение объектов
- 2) Сопровождение объекта
- 3) Отображение результатов

Рассмотрим каждый этап.

Обнаружение объектов — на этом этапе происходит распознавание образов объектов и их выделение (так называемая «bounding box» задача).

Сопровождение объектов — попытка сопоставить новые объекты со старыми. Для этого производится поиск по прошлым объектам. Если подходящий старый объект не был найден, или слишком далеко — происходит создание нового.

Отображение результатов — может происходить как в отдельном окне, так и на видео.

В программе для каждого из этапов создан свой модуль. В первом модуле происходит обнаружение объектов. Т.к. это самый длительный этап, происходит сохранение данных. После чего только что созданный файл используется в модуле для поиска связи между объектами. Этот поиск занимает сравнительно малое время и имеет сложность, близкую к линейной. Результатом этого поиска являются количество распознанных объектов, их тип и расположение, что после и будет отображаться. Основной проблемой, пожалуй, может стать количество данных для обработки, т.к. нужны все найденные объекты на всех кадрах видео.

Модуль сопровождения объектов

Рассмотрим работу модуля для сопровождения объектов.

Входными данными для этого модуля будут являться данные, полученные из прошлого модуля (модуль обнаружения объектов). Основные данные, которые будут использоваться — это: название объекта и его положение в кадре. И так для каждого кадра.

Получив эти данные, модуль приступает к их обработке. При этом следует указать 3 основных параметра для работы: *max_age*, *min_square* и *threshold_distance*.

Переменные несут следующий смысл:

- 1) *max_age* — изменяется в кол-ве кадров. Если условная цель наблюдения была потеряна в данном кадре, то, по прошествию данного кол-ва кадров, данная цель считается окончательно потерянной из виду.
- 2) *min_square* — минимальная площадь объекта. Если площадь будет ниже указанного значения — этот объект не учитывается.
- 3) *threshold_distance* — дистанция, за пределами которой не будет происходить поиск нового объекта. Т.е. если расстояние от условной цели до любого другого объекта будет больше указанной величины, то цель переходит в статус потерянной.

Используя указанные выше параметры, происходит итеративное сопровождение старых объектов (с прошлого кадра) и сопоставление их с новыми (с текущего кадра). На выходе данного модуля для каждого кадра получаем кол-во обнаруженных объектов до этого кадра включительно (с учётом их сопровождения), а также уникальный идентификатор каждого обнаруженного объекта и его координаты.

Обнаруженные проблемы

В ходе проверок работы программы были обнаружены проблемы, которые весьма сложно решить приведённым способом:

- 1) При перекрытии машин другими объектами (статуями, деревьями или другими машинами) будет возрастать счётчик их количества. Решения этой проблемы в данный момент не найдено. Решением может стать введение точек, из которых автомобили появляются на экране и отсеивание новых обнаруженных объектов, находящихся слишком далеко.

2) При статичном фоне из машин, стоящих на парковке, программа может посчитать и их. Решением может стать введение зон, из которых машины появляются на экране и отсеивание новых обнаруженных объектов, находящихся слишком далеко от них.

3) Для отсеивания слишком далёких объектов из рассмотрения из-за нестабильности их обнаружения, был введён коэффициент, вводящий ограничение на минимальную площадь. Однако, если машина, находящаяся вблизи и которая была обнаружена ранее, начнёт поворачивать, то она может на время пропасть из обработки, и, будучи обнаруженной после, посчитаться как новая. Решением может стать введение разграничения на области для кадров из видео. При этом, следует для каждой области указать свой коэффициент.

4) При слишком быстром движении автомобиля и при малом количестве кадров в секунду у видео, данный метод будет засчитывать эту машину несколько раз. Для решения можно использовать фото автомобиля в каждом кадре для сравнения между собой. Однако подобное сравнение весьма трудоёмко и такой подход весьма непрост в применении при сложной дорожной обстановке.

Заключение

В данной работе были проведены исследования по анализу дорожного трафика. Были рассмотрены программные способы подсчёта проезжающего транспорта.

- 1) Был проведён анализ существующих методов решений в данной области
- 2) Были исследованы сильные и слабые стороны данных методов
- 3) Была спроектирована программа, использующая данные методы
- 4) Был проведён анализ полученных результатов

Литература

1. Machine Learning for Beginners: An Introduction to Neural Networks [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9>. – Дата доступа: 21.04.2019

2. Convolutional Neural Networks [Электронный ресурс]. – Режим доступа: <https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a0508>. – Дата доступа: 20.04.2019

3. Эволюция нейросетей для распознавания изображений в Google: Inception-ResNet [Электронный ресурс]. – Режим доступа: <http://security-corp.org/companies/microsoft/33752-evolyuciya-neyrosetey-dlya-raspoznavaniya-izobrazheniy-v-google-inception-resnet.html>. – Дата доступа: 03.05.2019

4. YOLO: Real-Time Object Detection [Электронный ресурс]. – Режим доступа: <https://pjreddie.com/darknet/yolo>. – Дата доступа: 03.05.2019