



Рис. 3. Виды покрышек для спортивного картинга: 1 – Стики; 2 – Дождевые; 3 – Зимние

Различия в конструкции спортивных и прокатных картов объясняется различными целями, которые ставятся перед ними. Спортивный картинг направлен на достижение лучшего результата, увеличения скорости болида и его управляемости, поэтому на данные карты устанавливаются коробки переключения передач, гидравлические тормоза на переднюю и заднюю ось и т.д. Прокатный картинг имеет развлекательный характер, поэтому в данных картах делается все для повышения безопасности, облегчения управляемости карта и снижения стоимости конструкции. [5]

Список литературы

1. Исаенко А.Н. История развития тормозных систем автомобиля / А.Н. Исаенко, Е.В. Прохорова // Современные автомобильные материалы и технологии (САМИТ-2016): VIII Международная научно-техническая конференция. 24-25 ноября 2016 г. – Курск, 2016. – С. 158–161.
2. Устройство карта // Kart racing club URL: <https://www.karttracing.ru> (дата обращения: 10.12.2019).
3. Устройство гоночного автомобиля «карт» // URL: <https://kartodrom.com.ua/news/articles/ustroystvo-gonochnogo-avtomobilya-kart> (дата обращения: 10.12.2019).
4. Картинг // DriveContact URL: <https://drivecontact.ru/avtosport/karting/> (дата обращения: 10.12.2019).
5. Картинг: развлечение или спорт? // URL: <https://www.redbull.com/ru-ru/karting-entertainment-or-sports> (дата обращения: 10.12.2019).

АЛГОРИТМ АВТОМАТИЧЕСКОГО ПОДСЧЕТА ПАССАЖИРОПОТОКА В ОБЩЕСТВЕННОМ ТРАНСПОРТЕ

Головкин М.В., Бондарь А.С., Боровской А.Е.

Белгородский Государственный
Технологический Университет «БГТУ», Белгород,
e-mail: mixail.golovkin.1997@mail.ru

В данной статье рассматривается принцип работы и практическая реализация алгоритма автоматического подсчета количества пассажиров в общественном транспорте. Он позволяет автоматизировать данный трудоемкий процесс и повысить эффективность работы пассажирского транспорта. Данный алгоритм состоит из

реализации трех задач: детектирования пассажиров, определения траектории движения и их анализ. Детектирование осуществляется при помощи используемой нейронной сети, которая обучена для распознавания людей в кадре. Для определения траектории движения пассажиров используется алгоритм центроидного отслеживания. Анализ траектории движения заключается в сравнении направлений движения. По результатам этого анализа появляется возможность отличать входящих пассажиров от выходящих. Алгоритм создан на языке программирования Python. Также в статье приводятся результаты тестирования алгоритма подсчета пассажиров на примере видеозаписи с камеры уличного наблюдения, направленной в область пешеходного перехода. В результате получения данных автоматической фиксации и дальнейшего ручного подсчета приведено сравнение данных в таблице и определена общая точность подсчета. Система автоматического подсчета количества пассажиров необходима для того, чтобы вести учет пассажиров, оптимизировать расписание движения, маршрутную сеть, тарифную политику, контролировать выручку, сдаваемой персоналом и обеспечить контроль оплаты проезда.

Транспортным компаниям необходимо знать, сколько пассажиров они перевозят в разные моменты времени. Это дает возможность оценить загруженность общественного транспорта и проводить контроль оплаты проезда. Для того, чтобы правильно провести анализ, необходимо получить достоверные сведения о количестве пассажиров в конкретном автобусе. Это можно сделать путем подсчета людей на входе и на выходе. При реализации системы подсчета пассажиров решаются следующие задачи: мониторинг загруженности автобуса; оптимизация работы общественного транспорта; выявление времени, когда происходит пик или спад количества пассажиров; контроль оплаты проезда.

Цель исследования. Целью исследования является разработка алгоритма подсчета пассажиров для получения достоверных сведений о подсчете количества пассажиров для оптимизации работы маршрутных транспортных средств.

Для реализации данной системы на языке Python было разработано программное обеспечение, позволяющее подсчитывать количество входящих и выходящих пассажиров [1]. Для работы данной системы в верхней части дверного проема автобуса необходимо установить видеокамеру, направленную вниз. Данные, полученные с этой камеры обрабатываются на стационарном компьютере.

Принцип работы алгоритма подсчета пассажиропотока. Программа для обработки видеозаписи и подсчета пассажиров состоит из двух частей. В первой части видео считываются, изменяется масштаб, так как используемая нейронная сеть работает с разрешением видео 300x300 с целью обеспечения оптимальной производительности. Далее видео подается на обработку нейронной сети [2], которая возвращает список детектированных объектов [3]. Каждый объект представлен в виде прямоугольной рамки и вероятности нахождения этого объекта внутри этой рамки.

Координаты рамок сохраняются лишь в том случае, если вероятность определения больше, чем 0,2. Данное значение установлено по умолчанию, его можно изменить. Если установлено значение слишком большое, то часть детектированных пассажиров будет проигнорирована. А если установить значение слишком маленькое, то возможно ошибочное детектирование случайных областей изображения. Координаты рамок сохраняются с привязкой к конкретному кадру и далее создается файл с форматом «.json».

Вторая часть считывает этот файл с сохраненными данными. На основе этих данных с помощью центроидного метода отслеживания [4] формируются траектории движения пассажи-

ров. Проанализировав эти траектории, можно посчитать количество вошедших и вышедших пассажиров. В алгоритме центроидного отслеживания используется библиотека OpenCV [5]. Центроид - это центр прямоугольной рамки, которая визуализируется по видеопотоку при детектировании пассажиров. Центроиды одного и того же объекта в разных кадрах формируют траекторию движения. Далее траектории визуализируются, сохраняются и происходит процесс их анализа. Суть анализа заключается в считывании вектора, направленного из начальной точки траектории в конечную. Если составляющая координаты X вектора отрицательная, то увеличивается количество вошедших пассажиров, если положительная, то количество вышедших. Если траектория состоит менее чем из двух точек, то она не учитывается в подсчете. На видеозаписи с разрешением 300x300 отображается два счетчика: «input» и «output», то есть вошедшие и вышедшие пассажиры из автобуса, ID - это уникальный идентификатор каждого объекта. Таким образом на выходе получается два значения, по которым можно оценивать загруженность общественного транспорта.

Практическая реализация алгоритма. Перед началом работы подключаются все необходимые модули и библиотеки. При запуске первой части программы загружается нейронная сеть, используя аргументы «prototxt» и «model»:

Далее происходит процесс чтения первого кадра и вычисляются координаты для правильной обрезки кадра.

В ветвлении *if* происходит проверка, успешно ли прочитан файл. Далее создается словарь для описания конкретного кадра:

```
1. frame_dict = {'id': frame_id, «coord»: []}
```

После этого запускается процесс обработки всех детектированных объектов в цикле. Если вероятность больше указанной в аргументе «confidence» (по умолчанию 0.2) и индекс класса соответствует классу «person»(человек), то координаты рамок записываются в созданный ранее словарь:

```
1. for i in np.arange(0, detections.shape[2]):
2.     confidence = detections[0, 0, i, 2]
3.     idx = int(detections[0, 0, i, 1])
4.     if confidence > args[«confidence»] and idx == 15:
5.         coord = (detections[0, 0, i, 3:7] * 100).astype(«int»)
6.         frame_dict[«coord»].append(coord.tolist())
7.         data[«data»].append(frame_dict)
```

В последней строчке записывается в словарь «data» описание текущего кадра. Идет проверка на совпадение количества кадров. Как только номер текущего кадра равен количеству кадров цикл заканчивается и выводится сообщение, что файл обработан.

Далее создается файл с форматом «.json», в который записывается словарь «data».

При запуске второй части программы подключаются все необходимые библиотеки. Создается объект класса «CentroidTracker». Загружаются файлы, созданные первым алгоритмом и преобразуются из формата «.json» в словарь. Создается список для передачи рамок в трекер и в цикле идет процесс обработки. Рамки преобразуются в формат «numpy».

Далее добавляется преобразованная рамка в список «*rects*» и она визуализируется. Обновляется трекер, который возвращает список объектов. Если номер объекта больше количества траекторий, то в список «*trajectory*» добавляется новая траектория. В противном случае добавляется к существующей траектории новая точка (*centroid*).

В новом цикле создаются два счетчика: «*output*» и «*input*». После обработки траектории из списка «*trajectory*» программа получает количество точек траектории. Если траектория состоит менее чем из 2-х точек, то она игнорируется и начинается следующая итерация. В цикле обрабатываются все отрезки траектории, приходят координаты первой точки, второй точки, рисуется отрезок по траектории движения пассажира.

Определяется X составляющая суммарного вектора траектории:

$$1. x = (tr[length - 1] - tr[0])[0]$$

Если X больше нуля, то увеличивается счетчик «*output*» на единицу, иначе увеличивается счетчик «*input*» на единицу. Выполняются команды:

1. $if x > 0:$
2. $output += 1$
3. $else:$
4. $input += 1$

Оценка точности подсчета. Для определения точности детектирования и подсчета был проведен тест, который заключался в обработке часовой видеозаписи, снятой с камеры уличного видеонаблюдения. Камера была установлена так, чтобы угол наклона объектива и высота примерно совпадали с высотой и направлением камеры, установленной в общественном транспорте. Затем количество людей было подсчитано ручным методом и определен процент погрешности. После автоматического и ручного подсчета были получены результаты, приведенные в таблице.

Анализ подсчета количества людей

	Автоматический подсчет	Ручной подсчет
Счетчик «input» (вошедшие)	76	76
Счетчик «output» (вышедшие)	76	81
Всего, чел.	152	157
Общая точность подсчета, %	96,82	

По данным этой таблицы можно увидеть, что погрешность в результате проведения теста составила примерно 3%. Соответственно, при помощи разработанного алгоритма можно производить достаточно точный подсчет пассажиров в общественном транспорте, что в дальнейшем поможет развитию транспортных предприятий и улучшит работу общественного транспорта.

Заключение. Преимущество данной системы подсчета пассажиров перед другими

заключается в простоте использования и минимальных затратах на оборудование. Система проводит слежение за каждым пассажиром с момента его входа/выхода до его исчезновения из поля зрения камеры, что выгодно отличает ее от классических систем учета пассажиропотока, фиксирующими только момент пересечения некоторой границы (инфракрасный луч, ступенька на входе), в которых человек может пересечь эту границу несколько раз, например, когда пассажир находится в дверях автобуса. Также данная система помогает отслеживать размеры выручки автотранспортного предприятия. Имея данные по каждому автобусу о количестве пассажиров за какой-либо промежуток времени можно оптимизировать работу подвижного состава маршрутных транспортных средств.

Список литературы

1. Python: официальный сайт [Электронный ресурс]. – URL: <https://www.python.org/> (дата обращения: 01.12.2019).
2. Как работает нейронная сеть: алгоритмы, обучение, функции активации и потери: сайт [Электронный ресурс]. – URL: <https://neurohive.io/ru/osnovy-data-science/osnovnye-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/> (дата обращения: 05.12.2019).
3. Рохит Ганди. R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Алгоритмы обнаружения объектов [Электронный ресурс]. – URL: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> (дата обращения: 10.12.2019).
4. Adrian Rosebrock. Simple object tracking with OpenCV [Электронный ресурс]. – URL: <https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/> (date of the application: 12.12.2019).
5. OpenCV: официальный сайт [Электронный ресурс]. – URL: <https://opencv.org/releases/> (дата обращения: 15.12.2019).

ПРИМЕНЕНИЕ СТАТИСТИЧЕСКИХ МЕТОДОВ УПРАВЛЕНИЯ КАЧЕСТВОМ ДЛЯ АНАЛИЗА ПРОЦЕССА ПРОИЗВОДСТВА СТЕНОВОЙ ВНУТРЕННЕЙ ЖЕЛЕЗОБЕТОННОЙ ПАНЕЛИ ПСВ 47.27.16

Горчакова А.В., Назина Л.И., Клейменова Н.Л.
 ФГБОУ ВО «Воронежский государственный университет инженерных технологий», Воронеж,
 e-mail: lyudmila.nazina.62@yandex.ru

В работе использовались статистические методы контроля и управления качеством для анализа точности и устойчивости технологического процесса производства стеновой внутренней железобетонной панели. Проанализированы законы распределения показателей качества и определен возможный брак готовой продукции. Для оценки устойчивости процесса производства стеновой панели выбраны контрольные карты Шухарта для количественных данных. Для анализа причин брака продукции по показателю прочность проведен поиск причин возникновения дефектов при помощи диаграммы Исикавы, которая показывает возможные факторы, влияющие на появление дефектов стеновой панели. Для предотвращения возникновения