

УДК 004.852

К.В. Святлов, А.В. Михеев, М.А. Шлямов

ПРОГРАММНЫЙ КОМПЛЕКС ВИЗУАЛЬНОГО МОНИТОРИНГА ГРУЗОПЕРЕВОЗОК

Святлов Кирилл Валерьевич, кандидат технических наук, окончил Ульяновский государственный технический университет. Декан факультета информационных систем и технологий УлГТУ. Имеет статьи, монографии, учебное пособие и свидетельства о регистрации программы для ЭВМ. Область научных интересов – машинное обучение и робототехника. [e-mail: k.svyatov@ulstu.ru].

Михеев Александр Вячеславович, окончил УлГТУ, программист Ульяновского областного центра новых информационных технологий при УлГТУ. Область научных интересов – машинное обучение. [e-mail: a.miheev@simcase.ru].

Шлямов Максим Андреевич, студент факультета информационных систем и технологий УлГТУ. Область научных интересов – машинное обучение. [e-mail: m.shlyamov@simcase.ru].

Аннотация

В статье рассматривается процесс проектирования и реализации программного комплекса для контроля полноты загрузки грузового транспорта при проезде контрольных пунктов с использованием алгоритмов технического зрения и методов машинного обучения.

Контроль полноты загрузки используется в рамках государственного надзора за эксплуатацией российских дорог, а также контроля объема перевозимого груза в частных компаниях при исполнении заказов в рамках услуг по грузоперевозке. В статье рассматривается задача контроля объема перевозимого груза в грузовых автомобилях с открытым бортом, при этом в качестве груза рассматривается сыпучий нерудный материал (песок, щебень, уголь).

В настоящее время задача весового контроля автомобильных грузоперевозок решается в большинстве случаев с использованием весов. Также известны решения с использованием ультразвуковых датчиков расстояния, которые генерируют сигнал, соответствующий форме автомобиля при проезде под несколькими такими датчиками. Однако данные решения достаточно дороги в установке, не учитывают разницу в сигналах в случае использования разных типов или моделей автомобилей на одном маршруте перевозки. Кроме того, масса не всегда отражает полноту загрузки кузова, т. к. некоторые материалы при малом объеме весят гораздо больше полезного груза. А при разных погодных условиях (при повышенной влажности, например) масса определенных грузов может существенно изменяться в пределах 15%. При этом эффективность перевозок оценивается из расчета минимально возможного количества проездов для вывоза груза заданного объема.

Решить задачу контроля объема перевозимого сыпучего груза представляется возможным с использованием системы технического зрения. Описанию разработки такой системы и посвящена данная статья.

Ключевые слова: автоматизированное проектирование, искусственный интеллект, нейронные сети, машинное обучение, компьютерное зрение.

SOFTWARE PACKAGE FOR VISUAL MONITORING OF THE CARGO TRANSPORTATION

Kirill Valerevich Sviatov, Candidate of Science in Engineering; graduated from Ulyanovsk State Technical University; Dean of the Faculty of Information Systems and Technologies at Ulyanovsk State Technical University; an author of articles, monographs, textbooks, and certifications of software registration in the field of machine learning and robotic engineering. e-mail: k.svyatov@ulstu.ru.

Aleksandr Viacheslavovich Mikheev, graduated from Ulyanovsk State Technical University; software engineer of the Ulyanovsk Regional Center of New Information Technologies at UlSTU; interested in the field of machine learning. e-mail: a.miheev@simcase.ru.

Maksim Andreevich Shlyamov, Student of the Faculty of Information Systems and Technologies at Ulyanovsk State Technical University; interested in the field of machine learning. e-mail: m.shlyamov@simcase.ru.

Abstract

The article considers the process of designing and implementing a software package to check the completeness of trucks loading when passing checkpoints using technical vision algorithms and machine learning methods.

Load completeness checking is used in the governmental supervision of the Russian roads operation as well as checking the cargo transportation volume by private companies when executing orders under the agreements for cargo transportation services. The article deals with the problem of supervising the volume of transported cargo in open-top trucks with bulk non-metallic materials (sand, gravel, coal).

Currently, the problem of road-freight transport weight-checking is solved in most cases by the use of weighing units. Also, there are solutions when ultrasonic distance sensors are used that generate a signal corresponding to the shape of the vehicle when driving under several sensors. However, these solutions are quite expensive to install, they do not consider the difference between signals when various vehicle types or models are used on one route. In addition, the weight does not always reflect the completeness of the load body because some materials weigh much more than payload at low volume. And under different weather conditions (e.g. high humidity), the mass of certain goods can vary significantly within 15 percent. At the same time, the transportation efficiency is estimated based on the minimum possible number of passes for the export of cargo of the given volume.

The vision systems allow to solve the problem on checking the transported bulk cargo volume. The article focuses on the development description of such a system.

Key words: computer-aided design, artificial intelligence, neural networks, machine learning, computer vision

ВВЕДЕНИЕ

Целью реализуемого проекта является обеспечение максимальной загрузки машин (по объему кузова), вывозящих из карьера грунт, за счет контроля объема этого грунта с применением средств технического зрения. Проблема заключается в том, что плата за перевезенный груз рассчитывается исходя из количества рейсов, при этом не производится учет перевозимого грунта, а водители зачастую выполняют рейс с пустой или недостаточно загруженной машиной. При этом контроль должен осуществляться в любое время суток в любых погодных условиях с любыми типами грузовых автомобилей с открытым бортом.

Задача состоит из трех подзадач классификации изображений:

1. Определение грузовика в кадре.
2. Расшифровка бортового номера найденного грузовика (задача классификации с использованием сегментации).
3. Проверка наличия в кузове горки с грузом.

На основании современных исследований в области нейронных сетей можно выдвинуть требующую экспериментального подтверждения гипотезу о том, что решить указанные подзадачи возможно с использованием нейронных сетей совместно с классическими алгоритмами анализа изображений: сверточные нейронные сети [1] для определения грузовиков в кадре с предобученными весами [2] и заданной архитектурой. Архитектуры нейронных сетей в настоящее время разработаны для решения задач классификации разного масштаба и типа (например, VGG16, VGG19 [3], Inception V3 [4], ResNet50 [5], Xception и другие). Выбор конкретной архитектуры нейронной сети [6] зависит от используемого набора данных, объема обучающей выборки, вычислительных возможностей серверов, на которых будет производиться обучение. В статье описываются эксперименты по выбору и адаптации параметров нейронных сетей для решения за-

дач проекта, а также архитектура разработанного программного решения.

ОПИСАНИЕ АРХИТЕКТУРЫ РЕШЕНИЯ

Для решения задачи разработана специализированная система, позволяющая вести видеонаблюдение за грузовиками, проезжающими через контрольно-пропускной пункт (КПП), фиксировать параметры грузовиков и передавать их заказчику. Система распределенная и состоит из двух типов компонент: модуля сбора данных (МСД) и модуля анализа проездов или центрального модуля (ЦМ).

Эта простая архитектура, представленная на рисунке 1, позволяет легко масштабировать систему на несколько КПП, то есть увеличивать количество точек мониторинга. Обусловлена такая архитектура тем, что на узлах сбора данных необходимо выполнять часть операций обработки и уменьшать объем данных для передачи на сервер, т. к. без подобной оптимизации трафик

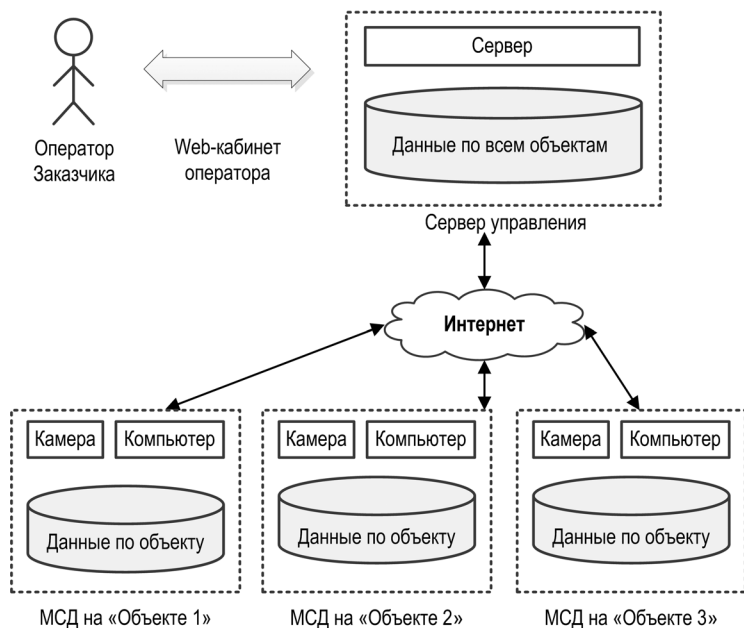


Рис. 1. Общая архитектура системы

становится очень большим. А в типовых условиях эксплуатации системы (горная местность за пределами городов) обеспечение стабильного и высокоскоростного интернета становится проблематичным, а иногда и вовсе невозможным. Однако полностью отказаться от серверной части и тем самым исключить необходимость передачи большого объема данных также нельзя по двум причинам. Во-первых, полный анализ проезда требует высокопроизводительных серверов, закупка которых для каждой точки сбора данных затратна. Во-вторых, требуется централизованное хранилище данных и web-портал с возможностью получить данные о проездах с подтверждениями в виде фотографий. Таким образом, часть графической информации все равно придется передавать на сервер, а значит и проводить фильтрацию на стороне узла сбора данных.

МСД предназначен для наблюдения за грузовиками, вывозящими из карьера грунт. Он устанавливается на выезде из каждого объекта. Оборудование МСД:

1. стационарный компьютер, имеющий доступ в Интернет;
2. видеочкамера для наружного наблюдения за грузовиками;
3. видеорегистратор для ведения резервных записей.

Функции МСД

МСД предназначен для решения следующих задач:

1. Слежение за дорогой через видеочкамеру:
 - 1.1. Запись видео и его сохранение в архив («Захват кадров» на рисунке 2). Резервный архив ведется для подстраховки на случай ошибок системы или спорных

ситуаций (для ручного анализа человеком в случае необходимости). Объем хранилища всегда составляет последний месяц, более поздние записи заменяются более новыми.

1.2. Непрерывное (с постоянным переподключением при разрывах связи) получение видеопотока с камеры по протоколу RTSP и передача кадров в программу для дальнейшей обработки.

2. Сбор кадров с проездом грузовика:

2.1. Предобработка видеопотока с камеры («Фильтрация» на рисунке 2), включающая в себя коррекцию яркости и контрастности, изменение размеров кадра, чтобы увеличить скорость дальнейшего распознавания, удаление шума, повышение резкости фото для обеспечения стабильности дальнейшего распознавания.

2.2. Выделение событий появления движущегося объекта в кадре («Детектор движения» на рисунке 2). Это позволяет системе выявлять и анализировать на видео только те кадры, на которых есть грузовики, для уменьшения объема обрабатываемых и передаваемых данных (например, на рисунке 3). Узел детекции движения выдает бинарную (черно-белую) маску, где каждый черный пиксель интерпретируется как отсутствие движения, а каждый белый – как его наличие. Пиксели сегментируются по принципу отличия от опорного кадра. Чем больше текущее значение цвета пикселя отличается от предыдущих состояний, тем больше вероятность, что в данном пикселе определено движение. Было использовано несколько методов нахождения движения (в основном из библиотеки OpenCV), среди лучших оказались: MOG2, ASBL, GMG.

ASBL – быстрый метод, и он довольно хорошо справ-

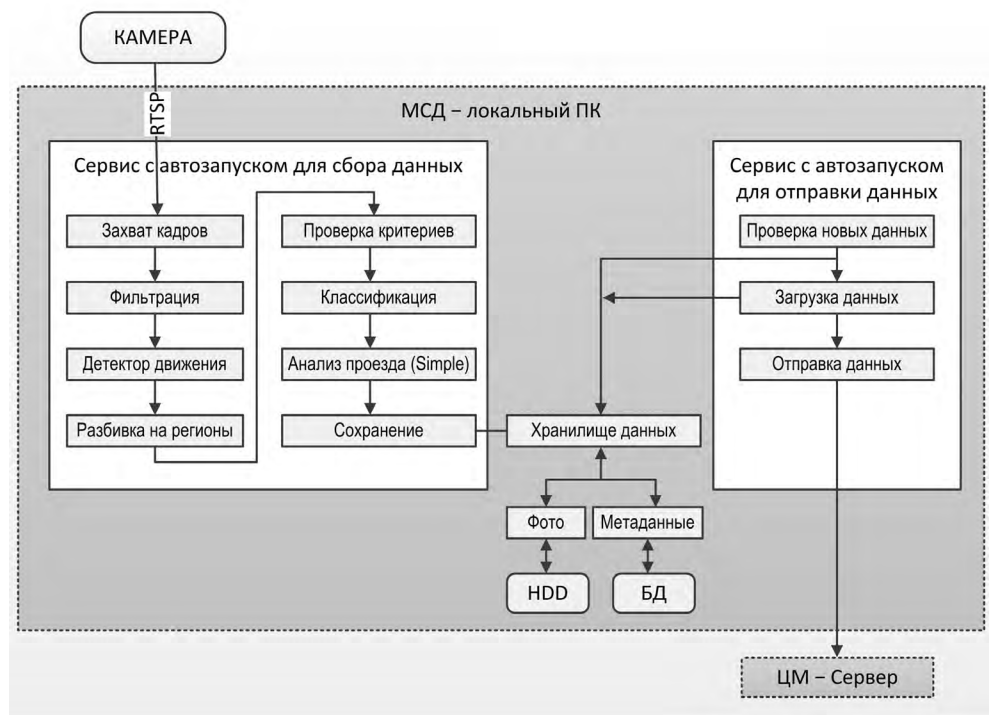


Рис. 2. Подробная логика работы МСД (HDD – дисковое хранилище, БД – база данных)

ляется с крупными движущимися объектами (целостно выделяет грузовики, в то время как GMG, например, разрывает целые объекты на множество пиксельных облаков). Однако этот метод дает большие артефакты при изменении освещенности сцены. А поскольку эксплуатация системы производится на открытой местности, а не в помещении, то погодные условия постоянно меняются и вносят значительную ошибку. В итоге в качестве детектора движения был выбран MOG2.

2.3. Фильтрация областей движения по размеру и положению для выявления потенциальных областей с грузовиками. Это необходимо, потому что в кадр, кроме грузовиков, могут попасть и другие движущиеся объекты и автомобили вне зоны интереса.

2.4. Классификация движущейся области («Классификация» на рисунке 2), определение грузовика. Это необходимо, так как в кадре могут появиться объекты разных типов: другой автомобиль, люди или животные, артефакты от погодных явлений (облака, тени, пыль от сильного ветра, которая может приводить к срабатыванию детектора движения). Для решения задачи классификации была выбрана предобученная сверточная нейронная сеть [1] на выборке данных imagenet [2] с архитектурой VGG16 [3]. Данную предобученную сеть можно легко найти в Интернете (например в составе библиотеки Keras [7]).

Свертка, показанная на рисунке 4, – это всего лишь линейное преобразование входных данных особого вида. Если x^l карта признаков в слое под номером l , то результат двумерной свертки с ядром размера $2d+1$ и матрицей весов W размера $(2d+1) \times (2d+1)$ на следующем слое будет таким [8]:

$$y_{i,j}^l = \sum_{-d \leq a,b \leq d} W_{a,b} x_{i+a,j+b}^l$$

где $y_{i,j}^l$ – результат свертки на уровне l , а $x_{i,j}^l$ – ее вход, то есть выход всего предыдущего слоя. Иначе говоря,



Рис. 3. Пример детектирования грузовика с выводом отладочной информации. В реальной системе прямоугольник и ключевые слова не отображаются в кадре, более того, кадры сжимаются специальным образом, чтобы снизить объем передаваемой информации

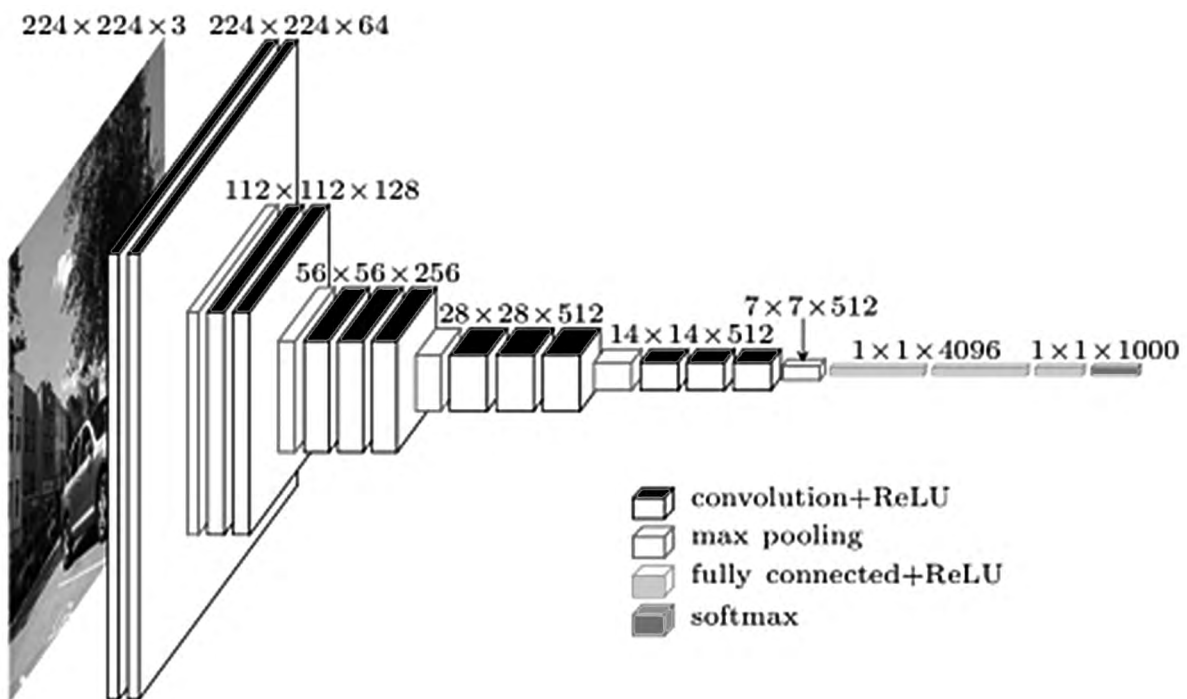


Рис. 4. Архитектура сверточной сети VGG16

чтобы получить компоненту (i, j) следующего уровня, мы применяем линейное преобразование к квадратному окну предыдущего уровня, то есть скалярно умножаем пиксели из окна на вектор свертки, как показано на рисунке 5.

Для обучения карты признаков [8] предположим, что мы оптимизируем некоторую функцию ошибки E и знаем ее значения на выходах нашего сверточного слоя. Чтобы провести итерацию обучения, нужно понять, как через них выражаются значения градиентов функции ошибки от весов. Через функцию взятия максимума ошибка проходит без изменений, слой субдискретизации ничему не обучает. Однако он делает разреженными градиенты, проходящие по графу вычисления, ведь из всех элементов окна субдискретизации частная производная относится только к одному, максимальному, а остальные получают нулевой градиент, и на этом их обучение можно будет считать законченным:

$$\frac{\partial E}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} \frac{\partial z_{i,j}^l}{\partial y_{i,j}^l} = \frac{\partial E}{\partial z_{i,j}^l} h'(y_{i,j}^l).$$

Детектор движения, рассмотренный выше, в сочетании с подобным классификатором обеспечивает точность обнаружения проездов 90%. В оставшихся 10% потери возникают из-за солнечных артефактов (4%), пересечения грузовиков (3%), ошибок нейронной сети (1%), прочих артефактов (2%).

2.5. Далее информация о каждом кадре и найденных на нем областях с грузовиками попадает в модуль «Simple Session Analyzer» (простой анализатор сессий проезда). Это агрегирующий компонент, который отмечает момент появления грузовика в кадре и начинает запись проезда до тех пор, пока грузовик (грузовики) не исчезнет из кадра. Данный экземпляр анализатора называется Simple, т. к. он не учитывает многих возможных сценариев взаимного движения грузовиков (например, обгон одного грузовика другим может привести к тому, что компонент запишет один проезд с двумя грузовиками вместо двух отдельных проездов). Данный компонент обрабатывает только простейшие сценарии проезда с одним грузовиком в кадре за раз. Кроме того, данный компонент фильтрует грузовики по заданному направлению движения.

ЦМ предназначен для сбора всей информации о грузовиках в одном месте и предоставления оператору доступа для работы с программной системой. ЦМ разворачивается в облаке и имеет доступ в Интернет.

Функции ЦМ

ЦМ предназначен для решения следующих задач:

1. Сбор информации со всех объектов учета (МСД). Между каждым МСД и ЦМ устанавливается соединение и на ЦМ регулярно загружаются данные о выездах грузовиков с грунтом. В случае нарушения связи МСД накапливает данные локально и при повторном соединении они загружаются на ЦМ без потери.

2. Предоставление оператору интерфейса для рабо-

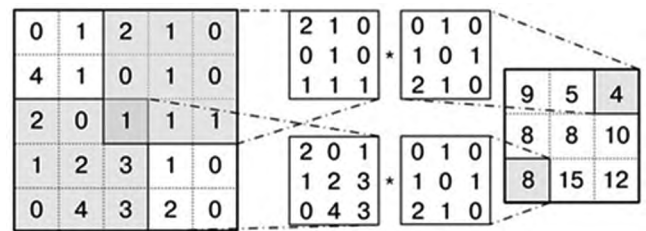


Рис. 5. Пример подсчета результата свертки: два примера подматрицы и общий результат

ты с системой. Оператор может получить доступ к данным о проездах грузовиков через КПП и настройке всей системы через панель управления (web-страница в Интернете). Панель управления предоставляет:

2.1. Защищенный доступ к данным.

2.2. CRUD-операции (чтение, изменение, удаление данных) для работы с транспортными компаниями. Оператор может просматривать и изменять информацию о текущих грузовиках, добавлять новые грузовики или удалять старые. Для каждого грузовика хранятся все данные, необходимые для формирования отчетов.

2.3. Просмотр данных о проездах грузовиков через КПП. Оператор может просматривать данные о проезде грузовиков через КПП по каждому объекту. При этом у него есть возможность осуществлять поиск и фильтрацию нужных данных для удобства.

2.4. Генерация отчетов. Оператор может формировать отчеты в виде Excel-файлов для дальнейшей работы с ними. При этом у него есть возможность выбирать временной интервал, объекты и подрядчиков, для которых необходим отчет.

3. Более глубокий анализ кадра для выделения одного грузовика на проезд. В случае если произошла ситуация, когда в кадре находилось одновременно N грузовиков, то данный компонент разбивает последовательность кадров на N проездов, в каждом из которых есть области только с одним уникальным грузовиком. Работа данного компонента основана на алгоритмах слежения за объектами (исходя из сравнения траектории движения и визуальных признаков между соседними кадрами).

4. Распознавание бортового номера по проезду.

Номер представляет собой не стандартизованный номерной знак на автомобиле, а рукописный текст на борту грузовика, как показано на рисунке 6. Цифры в номере могут отличаться друг от друга по стилю написания, размеру, толщине, цвету и местоположению.

На первый взгляд задача распознавания номеров относится к задачам классификации. При этом необходимо определить координаты номера грузовика в кадре и подавать выделенные области с цифрами на вход нейронной сети последовательно. В случае распознавания номерных знаков, которые характеризуются одинаковой структурой начертания цифр, одинаковым фоном и стилем написания, эта методика работала бы, и локализацию номера можно было бы реализовать относительно простыми методами. Но в данном случае

номера появляются всегда с разным размером, цветом, стилем, в разном месте на грузовике и на разном фоне. Реализовать классический алгоритм (или простой алгоритм) в таком случае кажется слишком сложным. Но на сегодняшний день нейронные сети вышли за пределы простых задач. Одна из таких задач – это сегментация изображения на объекты [9]. Сегментация по объектам отличается от задачи сегментации изображения по цвету тем, что к одному кластеру объектов относятся объекты по их семантической принадлежности (например, на фотографии улицы разными сегментами будут деревья, здания и автомобили).

Переход от задачи классификации к задаче сегментации позволил локализовать цифры каждого класса (т. е. от 0 до 9), то есть сразу оказались решенными задачи локализации и классификации. Окончательная постобработка заключается в том, что каждая область (пиксельная маска класса) сопоставлена с другой для того, чтобы понять направление надписи и сделать вывод об итоговом номере. Для решения задачи сегментации использовалась сверточная сеть типа Fully Convolutional Net (FCN), которая отображает картинку (маску классов), подобная архитектура в широком смысле является энкодер-декодером.

5. Распознавание уровня загруженности по проезду. Уровень загруженности в общем случае определяется дробным значением от 0 до 1, где 1 – полная загрузка кузова, а 0 – это такая загрузка, при которой груз не виден за бортом (это не означает, что грузовик совсем пуст). В рамках разработки прототипа задача упрощалась до целочисленной жесткой оценки 0, 1.

ОБЩИЕ ФУНКЦИИ МОДУЛЕЙ

Общие возможности модулей – это сквозная функциональность, которая реализована во всех компонентах системы для обеспечения стабильности работы разработанного программного обеспечения:

1. Обработка нештатных ситуаций. Это повышает отказоустойчивость системы. Например, если временно перестала работать камера или были проблемы со связью с ЦМ, то после исправления ситуации, МСД автоматически продолжит работать в штатном режиме, без дополнительных действий Оператора. Данные функции достигаются путем оборачивания приложений в службы и настройки автозапуска, а также сохранением и постоянным мониторингом состояния имеющихся и отпущенных данных.

2. Логирование событий и нештатных ситуаций.

ЗАМЕЧАНИЯ ПО РАСПОЗНАВАНИЮ НОМЕРОВ

В отличие от задачи классификации грузовиков для задачи сегментации номеров не было подходящей предобученной сети, поэтому была проведена серия экспериментов с разными архитектурами нейронных сетей и разными параметрами в рамках выбранной архитектуры:

Поскольку обучение глубокой сети на большом чис-



Рис. 6. Внешний вид бортового номера, написанного краской от руки

ле данных (более 5 тысяч изображений при разрешении 640×480) может занимать до 24 часов на видеокарте Nvidia GTX1070, потребовалось определенное планирование экспериментов. Каждый эксперимент проводился в несколько стадий:

1. Проверка работоспособности кода и сходимости модели;
2. Проверка качества на малой выборке;
3. Проверка качества на расширенной выборке.

После серии таких экспериментов отбирались лучшие модели, делались новые гипотезы и по итогу самые лучшие модели из нескольких партий обучались на полной выборке. Далее в таблице 1 приведены параметры и результаты некоторых экспериментов после определения стабильной и подходящей архитектуры нейронной сети (т. е. после первой стадии).

Таблица 1

Результаты части экспериментов, отображающие влияние размера выборки

Модель	Комментарии	Количество эпох	Выборка	Качество, %
Conv8	RGB, 960x547	50	1500	32
Conv10	RGB, 960x547	50	1500	45
Conv20	RGB, 960x547	50	1500	52
Conv26	RGB, 960x547	50	1500	54
Conv10	RGB, 960x547	50	10000	83
Conv20	RGB, 960x547	50	10000	89
Conv20	Только серый канал, 960x547	50	10000	82
Conv26	RGB, 960x547	25	10000	80

В названии модели указано количество слоев сети (например, Conv10 означает архитектуру сверточной сети с 10 слоями). Все архитектуры просты и подразумевают лишь слои свертки и развертки, а через каждые 3 слоя свертки идет слой MaxPooling, как в типовой архитектуре VGG.

Видно, что даже первые варианты архитектур (менее глубокие) стали показывать лучшие результаты после увеличения выборки. Поэтому следующая же гипотеза заключалась в том, чтобы увеличить выборку в разы. Однако разметка данных для сегментирующей сети довольно трудоемкая задача. Требуется не просто присвоить метку картинке, а требуется создать маску, на которой будут разными цветами закрашены соответствующие области (классы) на изображении. Для тысяч изображений это потребует много ручного труда.

Простейший способ обойти эту проблему – генерация синтетических данных путем небольших наклонов, смещений, масштабирований, изменений перспективы, цвета и яркости изображения. Подобная техника (data augmentation) является довольно эффективной в задачах компьютерного зрения, где применяются нейронные сети. На каждый оригинальный снимок грузовика, содержащий уникальное начертание номера, генерировалось N синтетических образов. Так на выборке в 10 тысяч изображений (2 тысячи оригинальных изображений) сеть Conv20 обеспечила точность 89%. Эта модель и была включена в прототип системы. У Conv26, вероятно, был больший потенциал, но обучалась такая модель на 25% медленнее, и потому на стадии прототипа был выбран более быстрый вариант.

Важно отметить, что качество распознавания цифр нейронной сетью не говорит о финальной точности

распознавания номера, т. к. номер получается в ходе совмещения отдельных цифр. Более того, поскольку цифры могут быть написаны в произвольном положении (со смещением и т. п.), то соединение отдельных сегментов не является тривиальной задачей. Возможно множество сценариев, в которых распознавание затрудняется, вот некоторые из них:

1. Много лишних областей (например, в цифре 8 она увидит несколько мелких зон с цифрой 3 или 0). Чтобы обработать такие ситуации, вводится фильтрация небольших областей. Однако грузовики проезжают под разными углами к камере – номера и даже цифры на одном грузовике могут быть написаны с сильным разбросом по размерам. Поэтому фильтрация размера по жесткой пороговой функции даст не очень хороший результат. Поэтому использовался относительный порог. Рассчитывалась площадь областей, и все области меньше $0,8 * M$, где M – средняя площадь, удалялись.

2. «Слипание» областей. В номере 115 идут подряд две цифры 1, соответственно сеть предоставит одну маску с большим пятном, описывающим большую цифру «1» вместо двух цифр. Подобные ситуации можно обработать эвристиками. Например, если номера 15 нет в базе, то наиболее вероятно это номер 115. Для решения этой проблемы также использовалась кластеризация областей, так «разорванные» области могли быть разделены простым алгоритмом с учетом их удаленности от центра облаков.

3. Потеря области. На одном кадре потеря происходит в 5% случаев. Но если обработать несколько кадров и затем выбрать наиболее частый номер, то это сгладит ошибки на отдельных кадрах («Распознавание номера» на рисунке 7).

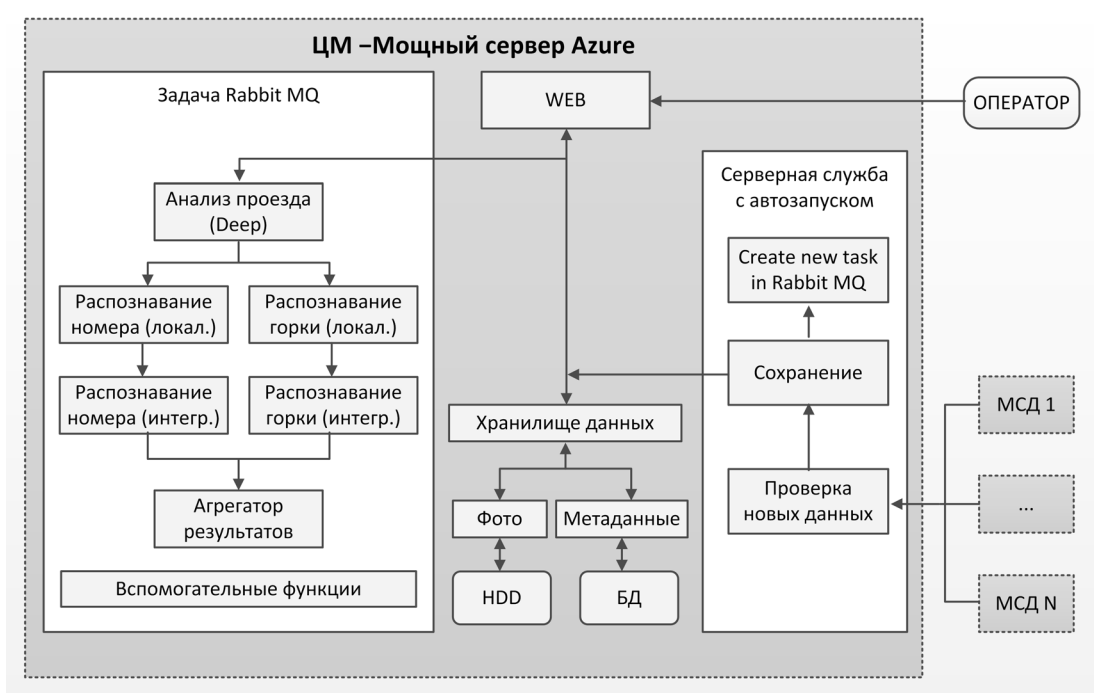


Рис. 7. Подробная логика работы модуля анализа данных или ЦМ

Ниже представлена таблица 2 с показателями качества на разных этапах распознавания для лучшей модели.

Таблица 2

Воронка качества распознавания от отдельных цифр до номера за один проезд грузовика

Стадия	%
Цифры	89
Группировка	85
«Слипание»	81
Номера, интегрально за проезд	94

ЗАМЕЧАНИЯ ПО РАСПОЗНАВАНИЮ ЗАГРУЖЕННОСТИ БОРТА

Если рассмотреть задачу распознавания уровня загруженности как задачу бинарной классификации (1 – есть горка над бортом, 0 – нет горки над бортом), то первая гипотеза может заключаться в том, чтобы подавать на вход сверточной сети только изображение грузовика, а не весь кадр (область, которая была выделена на предыдущем этапе по детектору движения). Но при таком подходе ни одна из сетей Conv8, Conv10, Conv20, Conv26 не достигла уровня выше 55% (т. е. не сильно превысила уровень случайного распознавания) на выборке 1500 изображений.

Вторая гипотеза заключалась в том, чтобы выделить контур горки по контуру движущейся области. Таким образом можно сообщить нейронной сети дополнительную информацию для анализа. Однако и такой подход не дал приемлемых результатов.

Третья гипотеза заключалась в том, чтобы использовать только контур горки и по форме этого контура делать вывод об уровне загрузки. Но и этот метод не дал хорошего качества. После детального анализа выяснилось, что только лишь контур горки нес слишком мало информации, кроме того, он часто выделялся неверно из-за облаков пыли, теней, солнечных бликов и т. п.

В итоге было решено, что нужно локализовать на изображении верхнюю область кузова грузовика, чтобы упростить распознавание («Распознавание горки» на рисунке 7). Поэтому сеть типа Conv20 была обучена выделять кузов грузовика, после чего производился морфологический анализ верхней части кузова. Горка выделялась посредством детектора движения, т. к. нейронная сеть не давала приемлемого уровня сегментации. Особенно в ночное время, когда горка горной породы слишком темная и сливается с фоном позади грузовика. Финальный шаг обработки заключался в обычном сравнении длины контура горки и длины борта грузовика, с учетом того, что горка поднимается выше некоторого порогового значения над уровнем борта. Этот подход дал лучшие результаты и был взят в качестве прототипа. Далее в таблице 3 приведены результаты подобного алгоритма.

Таблица 3

Воронка качества распознавания уровня загруженности за один проезд грузовика

Стадия	%
Уровень загрузки по одному кадру	72
Уровень загрузки интегрально за проезд	78

ЗАКЛЮЧЕНИЕ

Современное состояние исследований в области нейронных сетей, а также активные разработки программных библиотек для использования разработанных архитектур позволяют достаточно быстро прототипировать решения в области технического зрения для решения задач классификации, сегментации, анализа изображений в различных отраслях народного хозяйства. При этом стоимость проведения экспериментов снижается за счет использования предобученных моделей, доступных в глобальной сети. В статье был рассмотрен опыт быстрого прототипирования моделей для сложного анализа изображений и последующего внедрения этих моделей в реальную систему. Разработка подобных систем, сочетающих в себе как технологии машинного обучения, так и классическое клиент-серверное приложение, а также специфику работы с «железом» (камерой) на сегодняшний день возможна даже для очень малых групп разработчиков (до трех человек) и не требует обширных исследований или кодирования сотен сценариев обработки входных данных. В статье была подробно рассмотрена архитектура системы компьютерного зрения, которая позволяет значительно сократить издержки за счет контроля перемещения пустых грузовых автомобилей и предоставления статистики подобных перемещений.

СПИСОК ЛИТЕРАТУРЫ

1. Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation / Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel // Neural computation. 1989. Vol. 1, Is. 4. pp. 541–551.
2. Princeton university. 2016. – URL: <http://www.image-net.org>.
3. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. – URL: http://www.robots.ox.ac.uk/~vgg/research/very_deep/.
4. Rethinking the Inception Architecture for Computer Vision, 2005 / C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. – URL: <https://arxiv.org/abs/1512.00567>.
5. Deep Residual Learning for Image Recognition, 2015 / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. – URL: <https://arxiv.org/abs/1512.03385>.
6. Шишкин В.В., Стенюшкин Д.И., Михеев А.В. Автоматизация проектирования нейросетевых классификаторов // Автоматизация процессов управления. – 2014. – № 3 (37). – С. 112–119.

7. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions, 2016. – URL: <https://arxiv.org/abs/1610.02357>.

8. Николенко С., Кодурин А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб. : Питер, 2018.

9. Badrinarayanan V., Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. – URL: <https://arxiv.org/pdf/1511.00561.pdf>.

REFERENCES

1. LeCun Y., B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation. *Neural Computation*, 1989, vol. 1, iss. 4, pp. 541–551.

2. *Princeton: Princeton University*, 2016. Available at: <http://www.image-net.org>.

3. Simonian K., Zisserman A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Available at: http://www.robots.ox.ac.uk/~vgg/research/very_deep/.

4. Szegedy C., V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. *Rethinking the Inception Architecture for Computer Vision*, 2015. Available at: <https://arxiv.org/abs/1512.00567>.

5. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*, 2015. Available at: <https://arxiv.org/abs/1512.03385>.

6. Shishkin V.V., Steniushkin D.I., Mikheev A.V. Avtomatizatsiia proektirovaniia neirosetevykh klassifikatorov [Automated Design of Artificial Neural Network Based Classifiers]. *Avtomatizatsiia protsessov upravleniia* [Automation of Control Processes], 2014, no. 3 (37), pp. 112–119.

7. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions, 2016. Available at: <https://arxiv.org/abs/1610.02357>.

8. Nikolenko S., Kodurin A., Arkhangel'skaia E. *Glubokoe Obuchenie. Pogruzhenie v mir neironnykh setei* [Deep Learning. Jumping into the Neural Network World]. St.Petersburg, Piter Publ., 2018.

9. Badrinarayanan V., Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. Available at: <https://arxiv.org/pdf/1511.00561.pdf>.