

УДК 004.4

DOI: 10.15827/0236-235X.141.144-158

Дата подачи статьи: 12.09.22, после доработки: 15.12.22

2023. Т. 36. № 1. С. 144–158

Автоматизированное решение задачи детектирования промышленных объектов на ортофотоплане с помощью нейронной сети

Т.Р. Назаров ¹, магистрант, timnazarovya@gmail.com

Н.А. Мамедова ¹, к.э.н., доцент, ведущий научный сотрудник базовой кафедры цифровой экономики, tamedova.na@rea.ru

¹ Российский экономический университет им. Г.В. Плеханова, г. Москва, 117997, Россия

Задача детектирования объектов является общей для любого типа изображений, отличаются только показатели качества детектирования и технологическое обеспечение процесса. Базой для данного исследования послужили материалы аэрофотосъемки промышленных объектов. Камеральная обработка снимков аэрофотосъемки путем перекрытия исходных фотоснимков для получения ортофотопланов отснятой местности является сложным и затратным процессом, автоматизация которого носит фрагментарный характер.

Настоящая статья содержит решение по автоматизации этапа построения контуров промышленных объектов в рамках процесса их детектирования на ортофотоплане. В качестве подхода, обеспечивающего автоматизацию, использовано моделирование обученной сверточной нейронной сети с одноэтапным прохождением по SSD-алгоритму и на основе метода обратного распространения ошибки. Результатом работы стал программный комплекс, способный выделить и классифицировать несколько объектов на ортофотоплане.

Описаны типы и способы хранения генерируемых данных для оптимальной работы с программным комплексом, а также процесс перехода от системы координат снимка к пространственной системе координат с использованием файлов привязки ортофотопланов.

Практическая значимость результатов заключена в том, что все шаги по разработке программного комплекса описаны: приведено обоснование выбора технологий и алгоритмов, выявлены и сформулированы требования к программному комплексу, описан процесс обучения нейронной сети, приведена структура проекта. Это позволяет не только воспроизвести предложенное решение задачи автоматизации, но и масштабировать его с учетом входных параметров детектирования промышленных объектов на ортофотопланах.

Ключевые слова: детектирование объектов изображения, промышленный объект, ортофотоплан, сверточная нейронная сеть, программный комплекс, автоматизация.

Аэрофотосъемка – один из методов дистанционного зондирования земной поверхности. Полученные снимки имеют высокое качество и детализацию, позволяют выполнять объективную качественную, количественную оценку и инвентаризацию объектов и территорий, создавать и обновлять топографические материалы. Трансформируя изображения местности по перекрывающимся исходным фотоснимкам, получают цифровые ортофотопланы. Благодаря привязке к геодезической основе ортофотоплан местности дает возможность достоверно воспроизвести земную поверхность в цифровой форме.

При обработке результатов аэрофотосъемки целесообразно применение искусственного интеллекта. Так, например, распознавание определенных объектов и их контуров позволяет автоматизировать получение картографиче-

ских материалов. При этом стоит отметить и ряд задач, с которыми традиционно можно столкнуться при разработке и интеграции в процессы компании программного продукта, основанного на применении искусственного интеллекта:

- выбор технологий, которые будут использоваться, ориентируясь на конкретные параметры проекта;
- учет предметной области и глубокое понимание процессов в сфере промышленного мониторинга;
- внимательное слежение за качеством автоматизации процесса и оперативное внесение изменений при интеграции программного продукта.

Тема исследования актуальна для современного промышленного мониторинга. Заказчи-

ками аэрофотосъемки часто являются крупные компании с большим количеством объектов и широкой географией. Для планового мониторинга, обновления ГИС или принятия управленческих решений требуется оперативное предоставление качественной информации. Следовательно, пространственные данные становятся весьма востребованным ресурсом и возрастает потребность в их быстрой обработке.

В настоящее время детектирование и разметка объектов при обработке материалов аэрофотосъемки происходят вручную [1]. В результате полученные цифровые ортофотопланы становятся дорогостоящим продуктом. Причем низкая производительность при их создании усугубляется риском ошибки детектирования. Внедрение информационных систем с применением искусственного интеллекта может значительно улучшить качество промышленного мониторинга и управленческих решений для компаний, владеющих промышленными объектами или инвестирующих в них.

Степень изученности применения технологий искусственного интеллекта при распознавании объектов на ортофотопланах невелика: на рынке попросту отсутствует ПО под эти нужды. Каждая компания вынуждена самостоятельно создавать инструмент для автоматизации данного процесса или нести затраты, связанные с детектированием в ручном режиме. Результаты некоторых успешно реализованных проектов опубликованы в [2–4], но использовать или масштабировать их сложно, поскольку они или недостаточно подробно изложены, чтобы воспроизвести процесс, или ориентированы на решение узкоспециализированных задач.

Постановка и описание задачи

Перед авторами данного исследования стояла задача разработать программный комплекс, позволяющий детектировать промышленные объекты на ортофотопланах, создаваемом по материалам аэрофотосъемки.

Теоретическую основу составили работы преимущественно прикладного характера, хотя были использованы и публикации, освещающие методологические аспекты работы с изображениями, а также решения технологического развития предметной области [5, 6]. Были подобраны и обобщены результаты решения задачи детектирования объектов на изображении [7], а также изучены теоретические основы

процесса аэрофотосъемки и работы с полученными данными [8]. Исследованы базовые положения теории нейронных сетей, в частности, теории сверточных нейронных сетей [9–11]. На основе [12–14] обобщены практики глубокого обучения и сведения, дающие необходимое представление о возможностях применения искусственного интеллекта для промышленного мониторинга.

Данные публикации дали возможность исследовать уязвимости в процессе работы с изображениями и тем самым уточнить цель. Кроме того, изучение технической платформы и решений уже проведенных проектных работ и представленных в публикациях отчетных данных [15–17] позволило обосновать в дальнейшем выбор средств и технологий для создания программного комплекса.

Разработанный с применением технологий искусственного интеллекта программный комплекс встраивается в текущие процессы компании для корректного детектирования промышленных объектов на ортофотопланах. Учет конкретных потребностей компании позволил создать востребованный программный продукт, имеющий потенциал для коммерциализации.

В качестве преимуществ проведенного исследования, подтверждающих его актуальность и результативность, могут быть выделены следующие:

- проанализированы способы автоматизации процесса обработки материалов аэрофотосъемки;
- выявлены возможные способы оптимизации процесса работы с ортофотопланами, создаваемыми по материалам аэрофотосъемки;
- рассмотрены инструменты, способствующие наиболее эффективному решению задачи автоматизации детектирования объектов на ортофотопланах, и обоснован их выбор, в частности, архитектуры нейронной сети;
- выявлены и сформулированы требования к программному комплексу;
- описан процесс обучения нейронной сети, приведена структура проекта и раскрыт способ хранения данных.

В перспективе полученные результаты могут быть протестированы, что повысит надежность программного комплекса и поможет сформулировать новые запросы к автоматизации процесса детектирования объектов ортофотопланов. В каждом конкретном случае при внедрении программного комплекса должны быть оценены характеристики его интеграции

в информационную архитектуру компании и проведена предварительная оценка экономической эффективности такого решения.

Последовательность выполнения исследования и обоснование выбора используемых методов

Формулирование требований к программному комплексу для детектирования объектов на изображении. Общая формулировка задачи разрабатываемого программного комплекса – автоматизировать процесс анализа изображений территории при проведении мониторинговых работ или подготовке картографических материалов. Цель проекта – сократить временные затраты на обработку пространственных данных и снизить риск повторной работы с ортофотопланами. Алгоритм работы нейросетевой архитектуры используемой нейронной сети должен обеспечивать наиболее точное обнаружение и локализацию объектов, необходимых для поиска. Программный комплекс должен быть ориентирован на возможность быстрого детектирования и выгрузки объектов, интересующих пользователя.

Необходимо, чтобы пользователь имел возможность указать тип объектов, которые он хочет обнаружить на подаваемых на вход ортофотопланах. Далее с помощью нейросети, обученной обнаруживать объекты конкретного типа, должен быть проведен анализ и обозначены необходимые объекты. Результаты могут быть выгружены в определенном формате (например, .geojson или .shp). Важная задача, которую также предстоит решить в ходе разработки программного комплекса, – переход от системы координат снимка к пространственной системе координат для получения географических координат детектированных объектов.

Можно выделить основные функции разрабатываемого программного комплекса:

- добавление новых объектов, доступных для обнаружения;
- удаление объектов из списка объектов, доступных для детектирования;
- детектирование выбранных объектов и выгрузка результатов детектирования в удобном формате (.shp, .geojson).

Чтобы реализовать подобный сервис, необходимо выполнить следующие задачи:

- спроектировать понятный и удобный для использования интерфейс взаимодействия с пользователем;

- собрать необходимые обучающие данные (набор данных) в нужном объеме;

- произвести обучение нейронной сети, при этом контролируя его качество посредством функции потерь модели.

Логично разбить программный комплекс на две составляющие: первая программа должна давать возможность редактировать перечень типов распознаваемых объектов, вторая – обеспечивать поиск необходимых объектов на входящих ортофотопланах.

Описание базовых условий работы с изображениями в проекте. Для выбора способа решения задачи детектирования объектов на изображении отдельного внимания заслуживает специфика собственно изображения. Подходы к зондированию территории и работе с изображениями сформированы давно и применяются на практике как своего рода золотой стандарт. Со временем менялись только техническая и технологическая основы.

В данном случае использованы наборы пространственных данных, полученных с применением технологии оперативной аэрофотосъемки. Она реализуется с применением высокотехнологичного общедоступного оборудования, например, *беспилотных летательных аппаратов* (БПЛА) или легких самолетов, что позволяет значительно снизить удельную стоимость работ. Появляется возможность оперативно получать результаты съемки на различных площадях (от 1 га до 100 кв. км) с высокой детализацией и точностью данных. Технология оперативной аэрофотосъемки базируется на использовании данных аэрофотосъемки сверхвысокого пространственного разрешения (размер пикселя на местности 5–10 см). Необходимая точность получаемых данных обеспечивается следующими параметрами:

- постоянная калибровка используемой фотокамеры;
- высокое разрешение снимков (20 мегапикселей);
- значительное перекрытие снимков (до 75 %);
- новейшие технологии попиксельной автоматической обработки изображений;
- использование контрольных наземных точек и спутниковых технологий местоопределения.

Такая точность является достаточной для высокой оценки надежности пространственных данных, что может быть подтверждено соответствующей математической моделью [18].

Данные могут поставляться либо отдельными файлами, либо в формате рисунка AutoCAD или другого специализированного ПО.

В качестве выходных материалов используются следующие наборы данных:

- ортофотоплан объекта (вплоть до масштаба 1:1000) в заданной системе координат с файлом координатной привязки;

- цифровая модель объекта с точностью 15–20 см по высоте в заданной системе координат с файлом координатной привязки.

Столь высокие требования к наборам данных были обусловлены необходимостью выполнять объективную качественную, а также количественную оценку и инвентаризацию объектов и территорий, создавать и обновлять топографические материалы до масштаба 1:1000 практически в режиме реального времени. Как следствие, качество выполненных съемочных топографических работ позволяет оптимизировать планирование работ, снижая риски повторной съемки. Это является узким местом для любой системы зондирования территории [19]. Но самое главное, что высокое качество топографических работ позволяет корректировать полученные полевые материалы непосредственно с ортофотопланов в камеральных условиях, поскольку перекрытие снимков составляет более 50 %.

В результате обработки изображений, полученных по данным аэрофотосъемки сверхвысокого пространственного разрешения, могут быть реализованы многие процессы, касающиеся промышленных объектов. Для представленного проекта наибольшую ценность имеет возможность реализации следующих процессов:

- поддержание в актуальном состоянии данных о наземных и подземных промышленных объектах, их коммуникациях;

- решение задач кадастрирования объектов промышленного мониторинга;

- мониторинг состояния местности нахождения объектов и их периферии;

- подготовка отчетной документации по результатам промышленного мониторинга;

- решение задач генерального плана и проектирования изменений объектов;

- съемка производимых в реальном времени изменений на объектах и контроль исполнительных съемок.

Несмотря на частые заблуждения относительно высокой стоимости аэрофотосъемки, получение топографических материалов обычно составляет не более 15 % от стоимости работы с объектом. Гораздо более существен-

ную часть затрат проекта составляет работа по детектированию объектов на изображении. Поэтому способы автоматизации даже отдельных подпроцессов этой работы весьма востребованы.

Представим технологическую схему процессов аэрофотосъемки и обработки данных. В целом это стандартная последовательность действий, отдельные отклонения от которой зависят от технических возможностей исполнителей [20]. Непосредственно после проектирования маршрута и выполнения аэрофотосъемки сырые, неформализованные данные подвергаются поэтапной обработке.

1. Обработка данных бортовых и наземных приемников *глобальной навигационной спутниковой системы* (ГНСС). Приемник ГНСС постоянно передает координаты центра каждого снимка при каждом нажатии затвора аэрофотоаппарата. При постобработке это позволяет восстановить точную траекторию полета по координатам центров фотографирования.

2. Цветовая коррекция снимков. Снимки приводятся к общему контрасту и яркости, удаляются тени и облака.

3. Накладной монтаж и контроль перекрытий и покрытия района работ. По координатам центров снимков определяется расположение объекта работ на локации. Контролируется покрытие всего объекта снимками с заданными продольными и поперечными перекрытиями с использованием QGIS.

4. Фотограмметрическая обработка. Производится точное взаимное уравнивание всех снимков с использованием предварительно найденных на них характерных точек. С помощью этого процесса снимки взаимно ориентируются.

5. Создание ортофотопланов – фотоизображений местности с учетом искажений, вносимых ее рельефом.

6. Создание картографического материала. По полученным на предыдущем этапе данным в QGIS выполняется построение контуров всех объектов, видимых на ортофотопланах. По тем же ортофотопланам при решении мониторинговых задач выполняются поиск необходимых объектов и оценка их состояния.

Следует отметить, что почти все перечисленные этапы технологического процесса выполняются в автоматическом режиме и только последний требует значительных временных затрат на ручной труд специалистов.

Мониторинг по материалам аэрофотосъемки и получаемым ортофотопланам, как и

картографирование, выполняется в ручном режиме: специалист визуально определяет и детектирует объекты. При создании картографического материала отмечаются все наземные объекты, видимые на ортофотоплане, – как площадные (леса, поля и т.д.), так и линейные (реки, ручьи, дорожная инфраструктура и т.д.) или точечные (столбы, мосты, опоры ЛЭП). Данный процесс занимает значительное время. Основные факторы, влияющие на трудоемкость детектирования, – размер обрабатываемой площади и насыщенность изучаемой местности объектами. В процессе мониторинга специалисту необходимо найти объекты определенного типа и провести оценку состояния территории вокруг них. Даже поиск конкретно заданных объектов на насыщенных различными структурами ортофотопланах является крайне трудоемкой задачей: необходимо просканировать сотни или тысячи квадратных километров, что равноценно визуальному просмотру нескольких десятков тысяч фотоизображений.

Очевидный недостаток технологической схемы процесса обработки данных аэрофотосъемки заключается в рисках качества работы сотрудников при визуальном анализе материалов аэрофотосъемки, ортофотопланов и в низкой производительности ручного труда. Автоматизация вышеописанных процессов позволит существенно сократить временные затраты, повысить эффективность работы отдельных специалистов и снизить риски реализации проектов промышленного мониторинга в целом.

С учетом сказанного в качестве технической задачи была заявлена разработка программного комплекса, автоматизирующего связку из двух последних этапов технологической схемы процесса обработки данных аэрофотосъемки и ортофотопланов. Автоматизированное решение даст возможность последовательно обрабатывать входные наборы данных и детектировать интересующие объекты.

Обоснование применения технологии сверточных нейронных сетей и метода детектирования объектов. Особенность создаваемого программного комплекса в том, что имеется потребность в обработке изображений высокого разрешения. Вариант с применением полносвязных нейронных сетей для работы с такими изображениями признан авторами данного исследования неоптимальным, поскольку в этом случае количество параметров нейронной сети будет расти очень быстро. Однако без

уменьшения размерности изображения и выделения важной специфической информации работа с изображениями будет затруднена. Именно под такие задачи была создана архитектура сверточных нейронных сетей, что и определило их использование в проекте.

Интерпретировать работу сверточной нейронной сети можно следующим образом: в ходе обработки входных данных происходит выделение абстрактных, малопонятных и трудных для интерпретации признаков. При этом сверточная нейронная сеть сама выбирает структуру и последовательность этих признаков, фильтруя малозначимые детали [21].

В типовой структуре сверточной нейронной сети большое количество слоев. После начального слоя (например, входного изображения) сигнал проходит серию сверточных слоев, в которых чередуются собственно свертка и слой субдискретизации. Чередование слоев позволяет составлять карты признаков из карт признаков, на каждом следующем слое карта уменьшается в размере, но увеличивается количество каналов. На практике это означает способность распознавания сложных иерархий признаков. Обычно после прохождения нескольких слоев карта признаков вырождается в вектор или даже скаляр, но таких карт признаков появляются сотни. На выходе сверточных слоев сети дополнительно устанавливают несколько слоев полносвязной нейронной сети (перцептрон), на вход которой подаются оконечные карты признаков. После прохождения сигнала через серию сверточных слоев и уплотнения с помощью субдискретизации система перестраивается от конкретной сетки пикселей с высоким разрешением к более абстрактным картам признаков. Как правило, на каждом следующем слое увеличивается число каналов и уменьшается размерность изображения в каждом из них. В конце концов остается большой набор каналов, хранящих небольшое число данных (и даже один параметр), которые интерпретируются как переход от конкретных особенностей изображения к более абстрактным деталям и далее к еще более абстрактным деталям вплоть до выделения понятий высокого уровня. Эти данные объединяются и передаются на обычные полносвязные слои. По сравнению с полносвязной нейронной сетью сверточная нейронная сеть имеет меньшее количество настраиваемых весов. Это позволяет уменьшать объем выполняемой работы путем обобщения информации, получаемой на вход слоя.

Обучение нейронных сетей, в том числе и сверточных, происходит с использованием метода обратного распространения ошибки. В течение процесса обучения нейронной сети необходимо на каждом этапе обучения производить корректировку весов связей между всеми элементами нейронной сети для уменьшения разницы (ошибки) между искомым значением признака входного тренировочного примера и значением признака, получаемым нейронной сетью. Делается это методом градиентного спуска. Градиент представляет собой вектор, координатами которого выступают частные производные функции по ее аргументам. Градиент указывает направление наиболее быстрого возрастания данной функции, а в некоторой точке – величину этого максимального роста. Смысл градиентного спуска состоит в том, чтобы последовательно вычитать из текущего значения параметра градиент функции в текущей точке, так как это является величиной наибольшего уменьшения целевой функции в данной точке. Определить параметр можно следующим образом: $W_i = W_{i_{prev}} - \alpha \frac{\partial E}{dW_{i_{prev}}}$, где

W_i – новое значение параметра; $W_{i_{prev}}$ – прошлое значение параметра; α – скорость обучения (определяется до начала обучения); $\frac{\partial E}{dW_{i_{rev}}}$ – частная производная функции ошибки от параметра.

Число объектов на изображении в исследуемой задаче детектирования объектов заведомо неизвестно. Поэтому метод классификации с локализацией в качестве решения задачи оказался неприменимым и предстояло сделать выбор в пользу одного из двух методов решения задачи детектирования объектов на изображении.

Первый из них – метод, основанный на регионе. Комбинируется он из двух этапов. На первом этапе отдельный слой нейронной сети выделяет регионы интереса. Как альтернатива возможен селективный поиск. Цель – выделить области, с высокой вероятностью содержащие внутри себя объекты. На втором этапе подключается классификатор, который определяет принадлежность объекта к исходным классам и регрессорам. Цель – уточнить местоположение ограничивающих рамок. Метод обеспечивает высокую точность, но его применение требует существенных временных затрат. Метод скользящего окна, при котором изображения анали-

зируются полностью, не рассматривался в качестве альтернативы, поскольку для него при рассмотрении каждого региона осуществляется попытка классификации группы пикселей, входящих в этот регион, что чрезмерно трудоемко.

Второй метод является одноэтапным. В нем локализация и классификация происходят одновременно и нет необходимости генерировать регионы интереса. Метод предсказывает координаты определенного количества ограничивающих рамок с результатами классификации и степенью уверенности, а в дальнейшем корректирует местоположение рамок.

Выбор метода зависит от ограничения объема вычислительной мощности, специфики предметной области и желаемой производительности разработанного инструмента. Чтобы в будущем масштабировать предложенные в исследовании решения поставленной задачи, были приняты усредненные по рынку значения объема вычислительной мощности [22], а в качестве базового критерия учтена оперативность получения данных по итогам детектирования объектов [23].

На основе этих допущений было решено использовать одноэтапный метод в качестве архитектурного решения для сверточной нейронной сети.

Углубляя и детализируя принятое решение о выборе метода детектирования объектов, необходимо коснуться также вопроса о выборе оптимального одноэтапного алгоритма. Среди одноэтапных алгоритмов можно выделить два наиболее популярных: YOLO и SSD [24]. Функциональные возможности YOLO ограничены предсказанием только одного класса для каждой ячейки, что делает невозможным обнаружение нескольких объектов в одной ячейке сетки. Поэтому был выбран алгоритм детектирования объектов SSD. Он включает в себя принципы YOLO, но при этом каждый слой свертки связан со слоем, обеспечивающим детекцию объектов.

Если в YOLO алгоритм детектирования работает с максимально сжатым изображением, то в SSD набор рамок, обрамляющих объекты, поступает на слой детектирования с разных уровней сжатия изображения и только после этого применяется подход Non-Maximum Suppression. Это означает, что рамки сортируются по вероятности включения в себя объекта определенного класса. Метрика для оценки сходства двух рамок в данном алгоритме – от-

ношение площади пересечения к площади объединения двух рамок. Таким образом, внутри класса выбирается рамка с наибольшей вероятностью и при сравнении ее с другими удаляются рамки с показателем сходства, превышающим пороговое, и с меньшей вероятностью содержания объекта. Данная архитектура позволяет детектировать и объекты малых размеров. Это также является преимуществом алгоритма SSD в сравнении с алгоритмом YOLO.

Обобщим главные моменты проведенного обоснования выбора применяемых в проекте технологий. Методология проекта будет основана на использовании сверточных нейронных сетей с одноэтапным прохождением по SSD-алгоритму.

Повторяя или масштабируя результаты проведенного исследования, рекомендуется учесть, что модель состоит из двух связанных частей. Первая часть – так называемая базовая модель. Обычно это предварительно обученная сеть классификации изображений, используемая в качестве средства извлечения свойств входного изображения, полезной информации, которая будет отражать характерные особенности поданного на вход ортофотоплана. Вторая часть – дополнительные слои, реализующие сам алгоритм детектирования необходимых промышленных объектов. Эта рекомендация особенно важна при выборе конкретной реализации нейронной сети для ее использования в процессе детектирования промышленных объектов на материалах аэрофотосъемки.

Выбор языка программирования для разработки программного комплекса и библиотеки при работе с нейронной сетью. Программный комплекс представляет собой интерфейс для взаимодействия с предобученной нейронной сетью. Среди языков, рассматриваемых в качестве вариантов, используемых для работы с нейронными сетями и методами машинного обучения, можно выделить трех лидеров – Java, C++, Python.

В данном случае для разработки программного комплекса определяющими факторами стали квалификация разработчика и потребность в постоянном обслуживании комплекса в период эксплуатации, а также в обновлении его структурных компонентов. Поэтому был выбран язык программирования Python [25].

Выбор языка программирования автоматически накладывает ограничения на использование библиотек для работы с нейронной сетью [26].

При выборе библиотеки для работы с нейронной сетью важно, чтобы она была про-

ста в использовании, совместима с разными платформами, операционными системами и имела подробную и хорошо оформленную документацию для ее изучения. Среди ведущих библиотек для работы с методами машинного обучения можно выделить Scikit-learn, Keras, PyTorch, TensorFlow [27].

В качестве инструмента, который в дальнейшем будет использоваться в разработке, выбрана популярная библиотека TensorFlow с открытым исходным кодом для машинного обучения. Поддерживаемая Google, TensorFlow обладает рядом преимуществ: бесперебойная работа, быстрые обновления и частые новые выпуски с новыми функциями, совместимость с различными платформами. Библиотека предлагает несколько уровней абстракции, что позволяет подобрать подходящий вариант в зависимости от нужд разработчика. Обучение и создание моделей происходят с помощью высокоуровневого Keras API, который упрощает начало работы с TensorFlow и машинным обучением. Быстрое выполнение кода обеспечивает немедленную итерацию и интуитивно понятную отладку. Для обучения больших моделей используется Distribution Strategy API для распределенного обучения на различных конфигурациях оборудования без изменения модели.

С учетом необходимости реализации программного комплекса в сжатые сроки проекта выбор данной библиотеки весьма обоснован. Высокий уровень ее абстракции облегчит взаимодействие с моделью нейронной сети, а совместимость с различными платформами позволит быстро интегрировать реализованный программный комплекс в информационную архитектуру организации.

Результаты

Процесс и результаты обучения модели сверточной нейронной сети. Применение TensorFlow Object Detection API. Создание точных моделей машинного обучения, способных локализовать и идентифицировать несколько объектов на одном изображении, остается основной задачей в области компьютерного зрения. Для локализации и идентификации объектов на изображении была выбрана платформа TensorFlow Object Detection API (TensorFlow OD API), распространяемая с открытым исходным кодом и построенная поверх TensorFlow. TensorFlow OD API упрощает создание, обучение и развертывание моделей об-

наружения объектов [28]. Это дает возможность внедрить модель в разрабатываемый программный комплекс и использовать ряд реализованных в API функций для работы с моделью сверточной нейронной сети.

В TensorFlow OD API реализовано контролируемое обучение (обучение с учителем). Это стало еще одним доводом в пользу выбора данной платформы для возможности применить метод обратного распространения ошибки. Необходимо, чтобы нейронная сеть, обучаясь на размеченном наборе данных, предсказывала ответы, которые далее используются для оценки точности алгоритма на обучающих данных. Так будет реализовано обучение с учителем – направление машинного обучения, объединяющее алгоритмы и методы построения моделей на основе множества примеров, содержащих пары «известный вход – известный выход» [29].

Чтобы все слои нейронной сети были должным образом натренированы, используем предобученные модели TensorFlow OD API через технику трансферного обучения (Transfer Learning), которая применяется к решению задачи знаний, извлеченных нейронной сетью при решении другой задачи. Глубокие нейронные сети требуют больших объемов данных для сходимости обучения. Поэтому часто встречается ситуация, когда для решаемой задачи недостаточно данных для того, чтобы хорошо натренировать все слои нейросети [30]. Это и позволяет сделать Transfer Learning.

Чаще всего Transfer Learning выглядит следующим образом: к натренированной на определенную задачу нейронной сети добавляются еще несколько скрытых слоев, которые позволяют использовать уже полученные знания для решения более конкретной задачи. Например, знания, полученные при обучении распознаванию различных предметов, могут применяться при решении задачи распознавания конкретных объектов [31].

TensorFlow Object Detection API предоставляет ряд предобученных моделей, которые могут использоваться для переобучения на новых исходных данных. Все модели были обучены на наборе данных Common Objects in Context (COCO) с 90 категориями объектов, и для каждой модели вычислены метрика эффективности (mAP) и средняя скорость обработки одного изображения фиксированного размера. Для поставленных задач проекта с учетом желаемого затрачиваемого времени и технических параметров имеющегося ПО в качестве

предобученной сверточной нейронной сети была выбрана модель `ssd_resnet50_v1_fpn_640x640_coco17_tpu8`. Она создана на архитектуре нейронной сети ResNet и может найти применение в задачах компьютерного зрения, таких как классификация изображений, локализация объектов, обнаружение объектов. Выбор данной модели происходил в рамках рекомендаций для реализации процесса аналитической иерархии, которая включает три этапа принятия решения – выбор платформы, архитектуры и метода [30].

Модель имеет 50 слоев глубокой сверточной сети в качестве базовой модели. Обучение нейронной сети осуществлялось методом обучения с учителем. В контролируемом машинном обучении важна маркировка изображений. Поскольку размеченные данные будут использоваться для обучения модели, она сможет учиться и давать результаты, основанные на предоставленных качественных данных. Такой подход полностью соответствует требованиям к оценке качества и надежности пространственных данных, а значит, позволит опираться на результаты обучения модели при принятии решений о полноте детектирования объектов на ортофотопланах.

В качестве примера промышленных объектов, которые будут детектироваться в программном комплексе, выбраны станок-качалка (скважина) – один из типов наземных приводов скважинных штанговых насосов при эксплуатации нефтяных скважин, и строительные площадки. Для разметки использовалась программа LabelImg – инструмент для графической аннотации изображений, написанный на языке Python. Интерфейс инструмента представлен на рисунке 1.



Рис. 1. Интерфейс инструмента LabelImg

Fig. 1. The LabelImg tool interface

Разметка сохраняется в виде xml-файла:

```
<annotation>
<folder>test</folder>
```



```
<filename>Almet_2000_3.jpgimage11.
jpg</filename>
<path>C:\Users\erida\OneDrive\
Рабочий стол\test\Almet_2000_3.
jpgimage11.jpg</path>
<source>
  <database>Unknown</database>
</source>
<size>
  <width>1561</width>
  <height>1561</height>
  <depth>3</depth>
</size>
<segmented>0</segmented>
<object>
  <name>Pumpjack</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  <bndbox>
    <xmin>9</xmin>
    <ymin>180</ymin>
    <xmax>119</xmax>
    <ymin>265</ymin>
  </bndbox>
</object>
</annotation>
```

Для станка-качалки было аннотировано 52 изображения, при этом 41 изображение выбрано в качестве тренировочной выборки, а 11 – в качестве тестовой. Для строительной площадки было аннотировано 30 снимков, из которых 25 использовались для обучения, а 5 для тестирования.

Параметры модели хранятся в файле расширения .config. Были выделены два важных настраиваемых параметра:

- batch_Size, обозначающий размер частей, на которые дробится дата-сет; при тренировке параметру присвоено значение 25;

- num_Steps, обозначающий количество итераций по всем частям, на которые разделен набор данных; при обучении параметру присвоено значение 600.

Отслеживать процесс обучения помогает встроенный в библиотеку TensorFlow дашборд TensorBoard (рис. 2). Это удобное средство визуализации, позволяющее контролировать метрики модели в течение обучения [32].

Функция потерь уменьшается по мере увеличения количества шагов. Модель проходит процесс обучения и лучше понимает данные. Очевидно, что при увеличении параметра steps функция потерь практически не изменяется и использование этого параметра с большим значением может привести к переобучению модели.

После обучения модель была опробована на ранее неизвестных изображениях (рис. 3).

Обученная модель обвела в рамку детектированный объект и утверждает, что найденный объект можно со 100 %-ной уверенностью отнести к типу «скважина».

Структура программного комплекса и реализация задачи перехода от системы координат снимка к пространственной системе координат. Рекомендуется следующая структура программного комплекса:

- models
 - scripts/preprocessing
 - generate_tfrecord.py
 - workspace/training_demo
 - annotations
 - exported-models
 - images
 - models

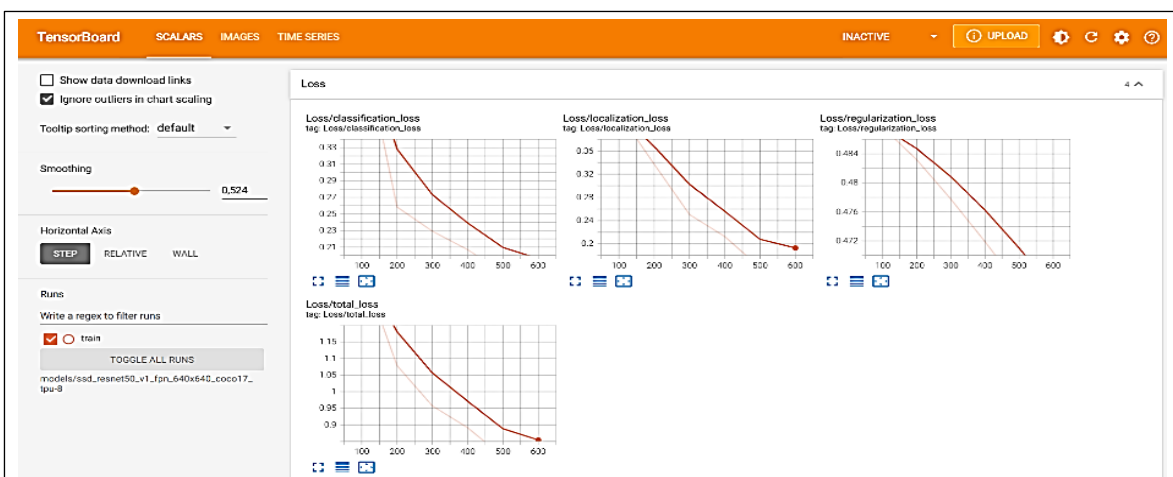


Рис. 2. Результат обработки данных в TensorBoard

Fig. 2. The data processing result in TensorBoard

- pre-trained-models
 - detector.py
 - exporter_main_v2.py
 - layer_manager.py
 - model_main_tf2.py

Основываясь на требованиях к программному комплексу, необходимо обеспечить хранение изображений и данных их разметки, предобученных моделей и самих компонентов программного комплекса.

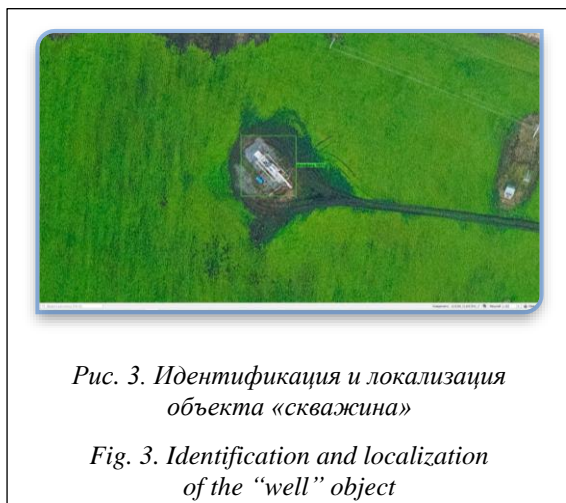


Рис. 3. Идентификация и локализация объекта «скважина»

Fig. 3. Identification and localization of the "well" object

Директория `scripts` содержит субдиректорию `subprocessing`, в которой размещены скрипты для предобработки данных. В директории `workspace` содержится директория `training_demo`, в которой есть все необходимые данные. В субдиректории `annotations` находятся информация о размеченных снимках и файл с закодированными типами детектируемых объектов, в `exported-models` – уже обученные модели, а в директории `models` – файлы конфигурации модели. Изображения для обучения и тестирования хранятся в субдиректории `images`.

Программы `exporter_main_v2` и `model_main_tf2.py` отвечают за экспорт обученной модели и обучение модели соответственно. `Layer_manager` управляет удалением и добавлением типов детектируемых объектов, а `detector` запускает процесс детектирования выбранных объектов на входных снимках.

Быстрый доступ и удобный формат данных обеспечат высокую работоспособность программного комплекса. Для решения задачи хранения были выделены два типа данных, доступ к которым критически важен.

- Файл с закодированными типами объектов. Такой файл называется `labelmap` и имеет расширение `.pbtxt`. Он помогает при поиске

класса объекта и является источником классов для аннотирования изображений. Элементы кодируются в json-подобном формате.

- Информация о снимках, в данный момент используемых в качестве тестовых и тренировочных данных. Непосредственно после разметки имеются только пары, включающие в себя файл со снимком и xml-файл с разметкой.

Однако по мере увеличения объема данных процесс сбора информации становится весьма трудоемким. Именно для этого в библиотеке TensorFlow существует формат TFRecord, позволяющий хранить все данные последовательно в одном файле, а не в разных местах, не тратить много времени на сбор этих данных и не нагружать память устройства [33]. Этот тип данных хорошо интегрирован в API TensorFlow и обеспечивает потоковую передачу данных. Единицей данных в TFRecord-файле является так называемый Example. Для создания TFRecord файла необходимо выполнить три шага:

- пройти по всем файлам разметки и собрать информацию о координатах размеченных рамок на каждом снимке;

- создать таблицу с указанием названия снимка и параметрами рамки, для чего удобнее использовать тип данных DataFrame (табличная структура данных) из библиотеки Pandas;

- итерировать полученную табличную структуру данных, ставя в соответствие каждой строке необходимый снимок и создавая Example с нужными параметрами.

Этот процесс можно схематично описать соответствующим кодом:

```
import glob
import pandas as pd
import xml.etree.ElementTree as ET
path='../Path to markup files'
for xml_file in glob.glob(path +
'/*.xml'):
    tree = ET.parse(xml_file)
    root = tree.getroot()
    filename = root.find('filename').text
    width = int(root.find('size').
find('width').text)
    height = int(root.find('size').
find('height').text)
    for member in root.findall('object'):
        bndbox = member.find('bndbox')
        value = (filename,
                width,
                height,
                member.find('name').text,
                int(bndbox.find('xmin').
text),
                int(bndbox.find('ymin').
text),
                int(bndbox.find('xmax').
text),
```

```

        int(bndbox.find('ymax') .
text),
        xml_list.append(value)
        column_name = ['filename',
'width', 'height', 'class', 'xmin',
'ymin', 'xmax', 'ymax']
        xml_df = pd.DataFrame(xml_list,
columns=column_name)

```

Таким образом, на выходе получается таблица для восьми параметров с информацией о названии снимка, длине и ширине окаймляющей рамки, координатах вершин и классе размеченного объекта.

При детектировании объекта на изображении обученная нейронная сеть обводит его в рамку. На выходе передаются четыре координаты – значение оси абсцисс и ординат левого верхнего угла и правого нижнего угла рамки в системе координат снимка. Эти координаты приведены в нормализованной форме. Однако при создании файла с пространственными данными требуется перейти из системы координат снимка в пространственную систему координат. Этому содействуют файлы координаты привязки, соответствующие каждому ортофотоплану. Файл состоит из шести строк, первая из которых обозначает размер пикселя в метрах, а последние две – центр самого левого пикселя в системе координат UTM. Для перехода из одной системы координат в другую были проделаны следующие шаги:

- прочтение построчного файла привязки;
- вычисление в системе координат UTM координат левого верхнего угла левого крайнего пикселя снимка;
- перевод нормализованных координат рамки, обводящей детектированный объект, в стандартный вид;
- перевод координат рамки из координат снимка в систему координат UTM, отталкиваясь от координат левого верхнего угла снимка.

Представим схематичное описание процесса перехода из системы координат снимка в пространственную систему координат в виде кода на языке Python:

```

with open(file_jgw) as f_jgw: # opening
the binding file
    lines=f_jgw.readlines() # all lines
of the file are read
    px_size=float(lines[0]) # pixel size
    half_px_size=float(px_size/2) # half
the pixel size
    x_center=float(lines[4]) # x of the
center of the upper-left pixel
    y_center=float(lines[5]) # y of the
center of the upper-left pixel
    x_up_left=float(lines[4]) - half_px_
size
    y_up_left=float(lines[5]) + half_px_
size

```

```

ymin = (int(boxes[0,0]*height)) # boxes
contains information about the number of
detected objects and data on the size of
the frames
xmin = (int(boxes[0,1]*width))
ymax = (int(boxes[0,2]*height))
xmax = (int(boxes[0,3]*width))
y_max_ref = y_up_left - ymin * px_size #
transition from pixel coordinates to
spatial coordinate system
x_min_ref = x_up_left + xmin * px_size
y_min_ref = y_up_left - ymax * px_size
x_max_ref = x_up_left + xmax * px_size

```

В результате пройденных этапов рамка вокруг объекта на изображении приобретает пространственные координаты. Следовательно, проектную задачу следует считать выполненной.

Заключение

В данной работе представлен анализ технологического процесса обработки материалов промышленной аэрофотосъемки и детектирования объектов на ортофотопланах. Определено, что ошибки, допущенные при детектировании объектов, являются следствием камеральной (ручной) обработки большого числа изображений. В качестве решения проблемы, следствием которой является повторное проведение дорогостоящей аэрофотосъемки и детектирования объектов, предложено автоматизировать процесс построения контуров промышленных объектов при их детектировании на ортофотоплане.

Для разработки программного комплекса, предназначенного для точного детектирования промышленных объектов на ортофотопланах, сформулированы функциональные требования, а также проведен анализ технологий и инструментов современных средств разработки. В качестве архитектуры нейронной сети выбрана сверточная архитектура, так как именно она позволяет работать с множеством изображений, а в качестве алгоритма детектирования выбран одноэтапный алгоритм Single Shot Detection. Обоснован выбор предобученной модели сверточной нейронной сети с использованием техники Transfer Learning. В качестве языка программирования выбран Python, а для работы с нейронными сетями его фреймворк TensorFlow.

Приведена структура программного комплекса, описаны процесс обучения нейронной сети методом контролируемого обучения, а также типы и способы хранения данных программного комплекса. Показана реализация процесса перехода от системы координат

снимка к пространственной системе координат с использованием файлов привязки ортофотопланов.

Применение разработанного программного комплекса востребовано промышленными

компаниями, которые используют материалы аэрофотосъемки для оценки состояния своей инфраструктуры и актуализации собственных карт и планов местности с учетом разветвленной сети промышленных объектов.

Литература

1. Wang R., Zhang S., Chen B., Xu J., Zhong L. Unmanned aerial vehicle (UAV) vision-based detection and location of power towers for transmission line maintenance. Proc. IET Conf. Publ., 2020, pp. 1937–1941.
2. Astudillo A., Al-Kaff A., Madridano A., Garcia F. Reducing the breach between simulated and real data for top view images. Proc. IEEE IV Symposium, 2021, pp. 510–515. DOI: 10.1109/IV48863.2021.9575845.
3. Pasquen J., Chaumont M., Subsol G., Derras M. An efficient multi-resolution SVM network approach for object detection in aerial images. Proc. IEEE Int. XXV Workshop on MLSP, 2015, pp. 1–6. DOI: 10.1109/MLSP.2015.7324329.
4. Jian L., Li Z., Yang X. et al. Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads. IEEE Consumer Electronics Magazine, 2019, vol. 8, no. 3, pp. 81–86. DOI: 10.1109/MCE.2019.2892286.
5. Рыльский И.А., Еремченко Е.Н., Котова Т.В. Устранение теней от облаков на материалах авиационных съемок в видимом диапазоне. ИнтерКарто. ИнтерГИС: матер. Междунар. конф., 2020. Т. 26. № 2. С. 286–297. DOI: 10.35595/2414-9179-2020-2-26-286-297.
6. Jaud M., Passot S., Le Bivic R. et al. Assessing the accuracy of high resolution digital surface models computed by PhotoScan® and MicMac® in sub-optimal survey conditions. Remote Sensing, 2016, vol. 8, no. 6, art. 465. DOI: 10.3390/rs8060465.
7. Muliady M., Sartika E.M., Lesmana C., Wianto E. UAV photogrammetry for generating 3D campus model. AIP Conf. Proc., 2019, vol. 2097, art. 030106. DOI: 10.1063/1.5098281.
8. Tahar K.N. Aerial terrain mapping using unmanned aerial vehicle approach. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., 2012, vol. XXXIX-B7, pp. 493–498. DOI: 10.5194/isprsarchives-XXXIX-B7-493-2012.
9. Zhao Z.-Q., Zheng P., Xu S.-T., Wu X. Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 2019, vol. 30, no. 11, pp. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.
10. Aziz L., Salam Md.S.B.H., Sheikh U.U., Ayub S. Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. IEEE Access, 2020, vol. 8, pp. 170461–170495. DOI: 10.1109/ACCESS.2020.3021508.
11. Jiao L., Zhang F., Liu F. et al. A survey of deep learning-based object detection. IEEE Access, 2019, vol. 7, pp. 128837–128868. DOI: 10.1109/ACCESS.2019.2939201.
12. Zuo J., Wang Z., Zhang Y. et al. Research on image super-resolution algorithm based on mixed deep convolutional networks. Computers and Electrical Engineering, 2021, vol. 95, art. 107422. DOI: 10.1016/j.compeleceng.2021.107422.
13. Öztürk M., Çavuş E. Vehicle detection in aerial imaginary using a miniature CNN architecture. Proc. Int. Conf. INISTA, 2021, pp. 1–6. DOI: 10.1109/INISTA52262.2021.9548348.
14. Jiang S., Zhang J. Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. Computer-Aided Civil and Infrastructure Engineering, 2020, vol. 35, no. 6, pp. 549–564. DOI: 10.1111/mice.12519.
15. Zhou Z., Irizarry J., Lu Y. A multidimensional framework for unmanned aerial system applications in construction project management. J. of Management in Engineering, 2018, vol. 34, no. 3, art. 04018004. DOI: 10.1061/(ASCE)ME.1943-5479.0000597.
16. Robinson C., Chugg B., Anderson B., Ferres J.M.L., Ho D.E. Mapping industrial poultry operations at scale with deep learning and aerial imagery. IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing, 2022, vol. 15, pp. 7458–7471. DOI: 10.1109/JSTARS.2022.3191544.
17. Filkin T., Sliusar N., Huber-Humer M., Ritzkowski M., Korotaev V. Estimation of dump and landfill waste volumes using unmanned aerial systems. Waste Management, 2022, vol. 139, pp. 301–308. DOI: 10.1016/j.wasman.2021.12.029.
18. Hoskova-Mayerova S., Talhofer V., Hoffmann A., Kubicek P. Mathematical model used in decision-making process with respect to the reliability of geodatabase. Studies in Computational Intelligence, 2013, pp. 143–162.

19. Colomina I., Molina P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.*, 2014, vol. 92, pp. 79–97. DOI: 10.1016/j.isprs.2014.02.013.
20. Crawford A., Mueller D., Joyal G. Surveying drifting icebergs and ice islands: Deterioration detection and mass estimation with aerial photogrammetry and laser scanning. *Remote Sensing*, 2018, vol. 10, no. 4, art. 575. DOI: 10.3390/rs10040575.
21. Dash T., Chitlangia S., Ahuja A. et al. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Sci. Reports*, 2022, vol. 12, art. 1040. DOI: 10.1038/s41598-021-04590-0.
22. Araujo G.F.M. The Importance of (Exponentially More) Computing Power. Brasília, DF Publ., 2021, 51 p. URL: https://bdm.unb.br/bitstream/10483/30713/1/2021_GabrielFilipeMansoAraujo_tcc.pdf (дата обращения: 10.09.2022).
23. Tan M., Pang R., Le Q.V. EfficientDet: Scalable and efficient object detection. *Proc. IEEE Conf. CVPR*, 2020, pp. 10778–10787. DOI: 10.1109/cvpr42600.2020.01079.
24. Liu W., Anguelov D., Erhan D., Szegedy C. et al. SSD: Single shot MultiBox detector. In: *Computer Vision – ECCV 2016*, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2.
25. TIOBE Index for December 2022. URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 10.09.2022).
26. Nguyen G., Dlugolinsky S., Bobák M. et al. Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*, 2019, vol. 52, no. 1, pp. 77–124. DOI: 10.1007/s10462-018-09679-z.
27. Python Developers Survey 2021 Results. URL: <https://lp.jetbrains.com/python-developers-survey-2021/> (дата обращения: 10.09.2022).
28. Morgunov A. TensorFlow object detection API: Best practices to training, evaluation & deployment. *MLOps Blog*, 2022. URL: <https://neptune.ai/blog/tensorflow-object-detection-api-best-practices-to-training-evaluation-deployment> (дата обращения: 10.09.2022).
29. Barba-Guaman L., Naranjo J.E., Ortiz A. Object detection in rural roads using Tensorflow API. *Proc. Incodtrin*, 2020, pp. 84–88. DOI: 10.1109/Incodtrin51881.2020.00028.
30. Ayar M., Dalkiran A., Kale U., Nagy A., Karakoc T.H. Image processing methods decision mechanism for surveillance applications with UAVs. *Aircraft Engineering and Aerospace Technology*, 2022. DOI: 10.1108/AEAT-05-2022-0121.
31. Ujilings J.R.R., Popov S., Ferrari V. Revisiting knowledge transfer for training object class detectors. *Proc. IEEE Conf. CVPR*, 2018, pp. 1101–1110. DOI: 10.1109/CVPR.2018.00121.
32. Khandelwal R. TensorBoard – a visualization suite for TensorFlow models. *Medium*, 2020. URL: <https://towardsdatascience.com/tensorboard-a-visualization-suite-for-tensorflow-models-c484dd0f16cf> (дата обращения: 10.09.2022).
33. Moroney L. Chapter 4. Using public datasets with TensorFlow datasets. In: *AI and Machine Learning for Coders*, 2020. URL: <https://www.oreilly.com/library/view/ai-and-machine/9781492078180/ch04.html> (дата обращения: 10.09.2022).

Automated solution of the problem of detecting industrial objects on a orthomosaic using a neural network

T.R. Nazarov¹, Graduate Student, timnazarovya@gmail.com

N.A. Mamedova¹, Ph.D. (Economics), Associate Professor, Leading Researcher of the Basic Department of Digital Economics, mamedova.na@rea.ru

¹ Plekhanov Russian University of Economics, Moscow, 117997, Russian Federation

Abstract. The task of object detection is common for any image type; only the detection quality indicators and the process technological support of differ. The basis for this study was the materials of aerial photography of industrial facilities. Office processing of aerial photography images by overlapping the original photographs to obtain orthophotomaps of the captured area is a complex and costly process, its automation is fragmentary.

This article contains a solution for automating the stage of constructing the industrial object contours as a part of the process of their detection on an orthophotomap. As an approach that provided automation, the authors used modeling of a trained convolutional neural network with a one-stage pass through the SSD algorithm

and based on the backpropagation method. The result was the development of a software package capable of identifying and classifying several objects on an orthomosaic.

For optimal work with the software package, there are described types and methods of storing generated data, as well as the implementation of the process of a transition from an image coordinate system to a spatial coordinate system using orthomosaic mapping files.

The practical significance of the results is in the fact that all steps in the development of the software package are described: there is the given rationale for choosing technologies and algorithms, there are identified and formulated requirements for the software package; there is a described process of training a neural network and the project structure. This allows not only reproducing the proposed solution of the automation problem, but also scaling it considering the input parameters for detecting industrial objects on orthomosaics.

Keywords: image object detection, industrial object, orthophotography, convolutional neural network, software package, automation.

References

1. Wang R., Zhang S., Chen B., Xu J., Zhong L. Unmanned aerial vehicle (UAV) vision-based detection and location of power towers for transmission line maintenance. *Proc. IET Conf. Publ.*, 2020, pp. 1937–1941.
2. Astudillo A., Al-Kaff A., Madridano A., Garcia F. Reducing the breach between simulated and real data for top view images. *Proc. IEEE IV Symposium*, 2021, pp. 510–515. DOI: 10.1109/IV48863.2021.9575845.
3. Pasquen J., Chaumont M., Subsol G., Derras M. An efficient multi-resolution SVM network approach for object detection in aerial images. *Proc. IEEE Int. XXV Workshop on MLSP*, 2015, pp. 1–6. DOI: 10.1109/MLSP.2015.7324329.
4. Jian L., Li Z., Yang X. et al. Combining unmanned aerial vehicles with artificial-intelligence technology for traffic-congestion recognition: Electronic eyes in the skies to spot clogged roads. *IEEE Consumer Electronics Magazine*, 2019, vol. 8, no. 3, pp. 81–86. DOI: 10.1109/MCE.2019.2892286.
5. Rylskiy I.A., Eremchenko E.N., Kotova T.V. Elimination of cloud shadows on materials of aviation shooting in the visible range. *Proc. InterCarto. InterGIS*, 2020, vol. 26, no. 2, pp. 286–297. DOI: 10.35595/2414-9179-2020-2-26-286-297 (in Russ.).
6. Jaud M., Passot S., Le Bivic R. et al. Assessing the accuracy of high resolution digital surface models computed by PhotoScan® and MicMac® in sub-optimal survey conditions. *Remote Sensing*, 2016, vol. 8, no. 6, art. 465. DOI: 10.3390/rs8060465.
7. Muliady M., Sartika E.M., Lesmana C., Wianto E. UAV photogrammetry for generating 3D campus model. *AIP Conf. Proc.*, 2019, vol. 2097, art. 030106. DOI: 10.1063/1.5098281.
8. Tahar K.N. Aerial terrain mapping using unmanned aerial vehicle approach. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, 2012, vol. XXXIX-B7, pp. 493–498. DOI: 10.5194/isprsarchives-XXXIX-B7-493-2012.
9. Zhao Z.-Q., Zheng P., Xu S.-T., Wu X. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 2019, vol. 30, no. 11, pp. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.
10. Aziz L., Salam Md.S.B.H., Sheikh U.U., Ayub S. Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review. *IEEE Access*, 2020, vol. 8, pp. 170461–170495. DOI: 10.1109/ACCESS.2020.3021508.
11. Jiao L., Zhang F., Liu F. et al. A survey of deep learning-based object detection. *IEEE Access*, 2019, vol. 7, pp. 128837–128868. DOI: 10.1109/ACCESS.2019.2939201.
12. Zuo J., Wang Z., Zhang Y. et al. Research on image super-resolution algorithm based on mixed deep convolutional networks. *Computers and Electrical Engineering*, 2021, vol. 95, art. 107422. DOI: 10.1016/j.compeleceng.2021.107422.
13. Öztürk M., Çavuş E. Vehicle detection in aerial imaginary using a miniature CNN architecture. *Proc. Int. Conf. INISTA*, 2021, pp. 1–6. DOI: 10.1109/INISTA52262.2021.9548348.
14. Jiang S., Zhang J. Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. *Computer-Aided Civil and Infrastructure Engineering*, 2020, vol. 35, no. 6, pp. 549–564. DOI: 10.1111/mice.12519.
15. Zhou Z., Irizarry J., Lu Y. A multidimensional framework for unmanned aerial system applications in construction project management. *J. of Management in Engineering*, 2018, vol. 34, no. 3, art. 04018004. DOI: 10.1061/(ASCE)ME.1943-5479.0000597.
16. Robinson C., Chugg B., Anderson B., Ferres J.M.L., Ho D.E. Mapping industrial poultry operations at scale with deep learning and aerial imagery. *IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022, vol. 15, pp. 7458–7471. DOI: 10.1109/JSTARS.2022.3191544.

17. Filkin T., Sliusar N., Huber-Humer M., Ritzkowski M., Korotaev V. Estimation of dump and landfill waste volumes using unmanned aerial systems. *Waste Management*, 2022, vol. 139, pp. 301–308. DOI: 10.1016/j.wasman.2021.12.029.
18. Hoskova-Mayerova S., Talhofer V., Hoffmann A., Kubicek P. Mathematical model used in decision-making process with respect to the reliability of geodatabase. *Studies in Computational Intelligence*, 2013, pp. 143–162.
19. Colomina I., Molina P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.*, 2014, vol. 92, pp. 79–97. DOI: 10.1016/j.isprsjprs.2014.02.013.
20. Crawford A., Mueller D., Joyal G. Surveying drifting icebergs and ice islands: Deterioration detection and mass estimation with aerial photogrammetry and laser scanning. *Remote Sensing*, 2018, vol. 10, no. 4, art. 575. DOI: 10.3390/rs10040575.
21. Dash T., Chitlangia S., Ahuja A. et al. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Sci. Reports*, 2022, vol. 12, art. 1040. DOI: 10.1038/s41598-021-04590-0.
22. Araujo G.F.M. *The Importance of (Exponentially More) Computing Power*. Brasília, DF Publ., 2021, 51 p. Available at: https://bdm.unb.br/bitstream/10483/30713/1/2021_GabrielFilipeMansoAraujo_tcc.pdf (accessed September 10, 2022).
23. Tan M., Pang R., Le Q.V. EfficientDet: Scalable and efficient object detection. *Proc. IEEE Conf. CVPR*, 2020, pp. 10778–10787. DOI: 10.1109/cvpr42600.2020.01079.
24. Liu W., Anguelov D., Erhan D., Szegedy C. et al. SSD: Single shot MultiBox detector. In: *Computer Vision – ECCV 2016*, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2.
25. *TIOBE Index for December 2022*. Available at: <https://www.tiobe.com/tiobe-index/> (accessed September 10, 2022).
26. Nguyen G., Dlugolinsky S., Bobák M. et al. Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey. *Artificial Intelligence Review*, 2019, vol. 52, no. 1, pp. 77–124. DOI: 10.1007/s10462-018-09679-z.
27. *Python Developers Survey 2021 Results*. Available at: <https://lp.jetbrains.com/python-developers-survey-2021/> (accessed September 10, 2022).
28. Morgunov A. TensorFlow object detection API: Best practices to training, evaluation & deployment. *MLOps Blog*, 2022. Available at: <https://neptune.ai/blog/tensorflow-object-detection-api-best-practices-to-training-evaluation-deployment> (accessed September 10, 2022).
29. Barba-Guaman L., Naranjo J.E., Ortiz A. Object detection in rural roads using Tensorflow API. *Proc. Incodtrin*, 2020, pp. 84–88. DOI: 10.1109/Incodtrin51881.2020.00028.
30. Ayar M., Dalkiran A., Kale U., Nagy A., Karakoc T.H. Image processing methods decision mechanism for surveillance applications with UAVs. *Aircraft Engineering and Aerospace Technology*, 2022. DOI: 10.1108/AEAT-05-2022-0121.
31. Uijlings J.R.R., Popov S., Ferrari V. Revisiting knowledge transfer for training object class detectors. *Proc. IEEE Conf. CVPR*, 2018, pp. 1101–1110. DOI: 10.1109/CVPR.2018.00121.
32. Khandelwal R. TensorBoard – a visualization suite for TensorFlow models. *Medium*, 2020. Available at: <https://towardsdatascience.com/tensorboard-a-visualization-suite-for-tensorflow-models-c484dd0f16cf> (accessed September 10, 2022).
33. Moroney L. Chapter 4. Using public datasets with TensorFlow datasets. In: *AI and Machine Learning for Coders*, 2020. Available at: <https://www.oreilly.com/library/view/ai-and-machine/9781492078180/ch04.html> (accessed September 10, 2022).

Для цитирования

Назаров Т.Р., Мамедова Н.А. Автоматизированное решение задачи детектирования промышленных объектов на ортофотоплане с помощью нейронной сети // Программные продукты и системы. 2023. Т. 36. № 1. С. 144–158. DOI: 10.15827/0236-235X.141.144-158.

For citation

Nazarov T.R., Mamedova N.A. Automated solution of the problem of detecting industrial objects on a orthomosaic using a neural network. *Software & Systems*, 2023, vol. 36, no. 1, pp. 144–158 (in Russ.). DOI: 10.15827/0236-235X.141.144-158.