

Построение системы технического зрения для выравнивания содержимого упаковок дельта-манипулятором на пищевом производстве

**О.В. Толстель
А.Е. Ширкин
А.Л. Калабин**

Ссылка для цитирования

Толстель О.В., Ширкин А.Е., Калабин А.Л. Построение системы технического зрения для выравнивания содержимого упаковок дельта-манипулятором на пищевом производстве // Программные продукты и системы. 2023. Т. 36. № 2. С. 334–341. doi: 10.15827/0236-235X.142.334-341

Информация о статье

Поступила в редакцию: 02.02.2023

После доработки: 21.03.23

Принята к публикации: 22.03.2023

Аннотация. Статья посвящена использованию компьютерного зрения для автоматизации процесса выравнивания на конвейерной линии упаковок с пищевым содержимым в картонных коробках. Данный процесс необходим для предотвращения появления неплотно закрытых коробок вследствие неравномерного распределения их содержимого. Объектом исследования является комбинация стереосистемы и алгоритмов искусственного интеллекта, решающих задачу детектирования и позиционирования объекта по 2D-изображению для выравнивания дельта-манипулятором. Такой манипулятор создан специально для автоматизации процесса выравнивания содержимого упаковок. Для задачи позиционирования был реализован алгоритм детектирования объектов на изображении, работающий в режиме реального времени. Обнаруженные 2D-участки изображения транслируются на стереосистему, формируя частичные карты глубин. Результатом является быстрый и точный алгоритм получения 3D-координаты объекта. Для определения текущего состояния содержимого упаковки и оптимизации количества операций выравнивания был реализован бинарный классификатор, который сообщает системе два типа статуса: объект выровнен, объект не выровнен. Для решения задачи собран специальный набор данных, состоящий из последовательности изображений частотой 30 кадров в секунду и протяженностью 1 час. В этом наборе на каждом изображении были размечены прямоугольными рамками упаковки с пищевым содержимым и отмечено их состояние – упаковка выровнена или нет. Для разметки использовалась программа с открытым исходным кодом LabelImg, предоставляющая графический интерфейс для разметки изображений, которая используется в дальнейшем обучении. Кроме того, создана программа, реализующая предложенный алгоритм на языке Python 3.6 с использованием интегрированной среды Jupyter Lab для операционной системы Ubuntu 18.04. Приведены результаты эксперимента по использованию предложенного алгоритма для оценки 2D-позиции объекта и текущего состояния детектируемого объекта. Для оценки качества алгоритма использованы метрика общей средней точности обнаружения объектов, а также метрики классификации – точность и полнота для задачи определения статуса содержимого упаковки.

Ключевые слова: искусственные нейронные сети, детектирование объектов, бинарная классификация, контроль качества, оценка 3D-позиции объекта, захват, робототехническая манипуляция

Автоматизация производственных процессов в пищевой промышленности в некоторых аспектах является более затруднительной, чем в других областях индустрии.

Сложность автоматизации заключается в процессах первичной упаковки пищевых продуктов в картонную коробку или в другую тару. Объекты на конвейерной ленте в основном имеют однородную форму, объем, массу или текстуру, что позволяет достаточно легко воздействовать на них при помощи манипуляторов в составе робототехнических комплексов. Перед этими комплексами ставится сложная задача автоматизации процесса воздействия на объекты, различающиеся по размеру, форме, качеству и весу, при этом сохраняя требуемую эффективность и поддерживая качество и целостность продукта.

Рассмотрим одну из задач, решение которой необходимо в цехе упаковки на пищевом производстве, – выравнивание содержимого коробок с целью их дальнейшего плотного закрытия на другом участке. В данной статье показано, как была решена составляющая компьютерного зрения, позволяющая специально созданному под эту задачу дельта-манипулятору совершать манипуляции с упаковками на конвейерной ленте.

Имеется производственная лента, по которой на скорости в среднем 80 штук в минуту перемещаются упаковки с пищевым содержимым, например, куриные отбивные в панировочных сухарях (далее – нагетсы). До попадания на ленту на другом участке производства нагетсы в пакетах укладываются в коробку манипулятором с вакуумным захватом. Пакет

приобретает выпуклую форму, что осложняет дальнейшее заклеивание коробки, так как крышка плотно не закрывается, в результате коробка или некачественно формируется, или остается незакрытой. Продукт не имеет товарного вида, и поэтому на участке упаковочного цеха работают два человека, которые выполняют эту задачу.

С целью автоматизации данной операции был разработан робототехнический комплекс с системой компьютерного зрения, способной обнаружить и локализовать положение коробки с нагетсами по 2D-изображению.

Для решения задачи обнаружения объектов были применены искусственные нейронные сети, что предполагает подготовку набора данных. Для этого на снимках или кадрах видео вручную обводятся прямоугольными рамками те объекты, которые необходимо обнаружить. Таким образом, одной из задач автоматизации участка конвейерной ленты была разметка данных при помощи утилиты LabelImg.

Обзор существующих методов

Подход, описываемый в данной статье, использует современные наработки в области нейронных сетей с некоторым учетом апробации подходов в бизнесе и производстве в задачах контроля качества продукции [1]. В работах [1, 2] описаны схожие методы оценки качества продукции бинарным классификатором в соответствии с парадигмой Индустрии 4.0. Этот метод используется и в рассматриваемой задаче, но с дополнением детектирования и локализации, что решает в реальном времени сразу две проблемы – локализации и контроля качества.

Используемые архитектуры нейронных сетей выбраны с точки зрения соотношения качества и скорости. Анализ используемых архитектур подробно выполнен в [3, 4], где на основании сравнения на больших наборах данных выявлено, что архитектура *ssd* до сих пор является оптимальной по точности и скорости обработки.

Набор данных

Реализация специального набора данных осуществляется в процессе видеозаписи на целевом пространстве, где будет работать алгоритм.

Сбор данных проводился на камеру машинного зрения компании Basler модели *acA1920*.

В процессе записи был получен 1 час видеоматериала с различных ракурсов работы производственной ленты. Видео имеет разрешение 1920×1080 , частота – 30 кадров в секунду. Видеоматериалы были преобразованы в набор кадров, каждый из них сохранен в формат JPEG. В итоге получено 108 000 изображений с ракурсом на производственную ленту, на которой перемещаются коробки с нагетсами.

Изображения, полученные с видео частотой 30 кадров в секунду, незначительно отличаются друг от друга, если брать каждый кадр последовательно, поэтому для разметки был взят только каждый пятый кадр, чтобы на изображении было больше уникальной информации. Таким образом, для разметки (рис. 1) были отобраны 21 600 изображений.

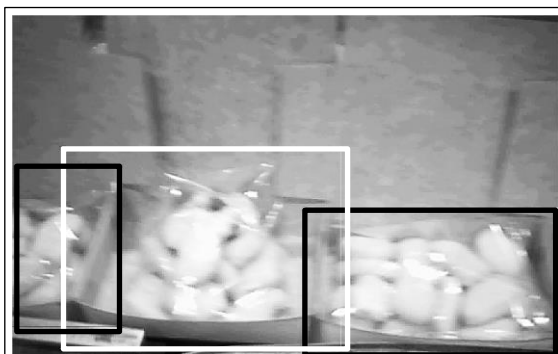


Рис. 1. Пример разметки набора данных: темный цвет прямоугольной рамки означает выровненное содержимое коробки, светлый – невыровненное

Fig. 1. An example of dataset markup, where a dark box color means aligned box contents, and a light color indicates unaligned box content

Чтобы использовать алгоритм, детектирующий на изображении коробки с пищевым содержимым, необходимо сформировать аннотации ко всем изображениям. Каждая аннотация описывает класс – наименование объекта, который требуется распознать, и ограничивающую рамку – прямоугольную рамку, заданную в виде вектора из четырех значений $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$, в которую заключен объект распознавания.

Аннотации для каждого изображения составляются при помощи утилиты разметки данных LabelImg (рис. 1). Она сохраняет ограничивающие рамки в формате Pascal VOC [5]; формат аннотации, в котором записан один из эталонных наборов данных, – Pascal VOC.

У каждого объекта в выборке есть два класса – выровнен и не выровнен, показываю-

щие, в каком состоянии находится объект и нужно ли применять к нему операцию выравнивания.

Полученный набор данных и аннотации использовались в дальнейшем для обучения алгоритмов машинного обучения, позволяющих решать поставленную задачу.

Материалы и методы

Для оценки положения объекта нужно решить три задачи: обнаружение объекта на изображении, бинарная классификация (выровненные и невыровненные коробки), локализация объекта в пространстве.

Для решения каждой из них были подобраны методы, удовлетворяющие как метрике точности, так и скоростным характеристикам, поскольку используемые алгоритмы должны стать частью робототехнического комплекса с высокой скоростью отклика.

Стоит отметить, что практические аспекты реализации решения поставленной задачи сильно влияют на выбор методов и топологий нейронных сетей. Так как робототехнический комплекс должен иметь достаточно точную и быструю систему локализации при конкурентоспособной стоимости компьютера, исполняющего алгоритм, необходимо исключить из конфигурации наиболее популярные в глубоком обучении графические процессоры. Популярность обусловлена их большим количеством на видеокартах, сопутствующим этому параллелизмом и библиотекой CUDA компании Nvidia, позволяющей эффективно распределять матричные операции, связанные с нейронными сетями.

Для получения наиболее оптимизированного решения, использующего для вычислений мощности центральный процессор (CPU), а не графические процессоры в видеокартах, была выбрана библиотека OpenVino toolkit компании Intel. Данный набор инструментов предназначен для оптимизированного исполнения моделей нейронных сетей на процессорах компании Intel, что позволяет удешевить конфигурацию компьютера, исполняющего алгоритмы. Важно, что обучение моделей нейронных сетей происходит также при помощи графических процессоров, то есть библиотека OpenVino обеспечивает оптимальное исполнение моделей, уже обученных при помощи других библиотек, таких как Tensorflow, Pytorch, Caffe, Mxnet и т.п. Поэтому для обучения моделей используется отдельный системный блок, уком-

плектованный видеокартой, а для исполнения применяется другой.

Обнаружение объекта на изображении

Большинство моделей нейронных сетей, поддерживаемых в OpenVino, имеются в стеке Tensorflow Object Detection API, поэтому для удобства конвертации моделей в формат OpenVino топологии нейронных сетей выбирались из стека Tensorflow Model Zoo, где размещены COCO, предобученные на наборе данных, состоящем из 90 классов [6]. Каждая модель имеет два параметра: средняя точность обнаружения объектов на изображении (mAP) и задержка в миллисекундах для обработки изображения из набора данных COCO.

В таблице 1 представлен список моделей, имеющих приближенную к реальному времени скорость обработки изображения.

Таблица 1

Топологии нейронных сетей из Tensorflow Model Zoo

Table 1

Neural network topologies from Tensorflow Model Zoo

Модель	Задержка	Средняя точность
ssd_mobilenet_v1_coco	30	21
ssd_mobilenet_v2_coco	31	22
ssd_inception_v2_coco	42	24
ssd_resnet_50_fpn_coco	76	35
centernet_resnet_50_coco	27	29.5
centernet_mobilenet_v2_coco	6	23.4

Данный список отображает соотношение скорости и качества моделей из поддерживаемых набором инструментов OpenVino. Модели указаны именами двух архитектур: нейронная сеть обнаружения объектов и экстрактор визуальных признаков объектов, на основании которого модель строит более абстрактное описание объекта для последующей классификации или регрессии. Экстрактор визуальных признаков является модулем для модели обнаружения объектов, и в зависимости от того, какое соотношение качества и скорости нужно получить, выбираются различные сочетания архитектур детектирования и извлечения признаков.

Известная архитектура *Single Shot detector* (SSD) [7] часто применяется на практике как имеющая высокие показатели скорости. Актуальная архитектура нейронных сетей CenterNet решает задачу обнаружения объектов в реаль-

ном времени [8]. По качеству и скорости она уступает только архитектуре YOLO v4 [9] – лидеру в упомянутом классе алгоритмов. Однако в данной архитектуре имеются реализации только под модели с большим количеством параметров, скорость работы которых не подходит под рассматриваемую задачу. Среди экстракторов визуальных признаков делается выбор между MobileNet v1 [10], MobileNet v2 [11], inception или GoogleNet v2 [12], resnet50 [13], resnet50-fpn [14].

Таким образом, по соотношению скорости и качества наиболее быстрой и точной оказалась архитектура centernet с экстрактором визуальных признаков resnet-50.

Для детектора был подготовлен ранее размеченный набор данных и разделен на две выборки, тренировочную и тестовую, в соотношении 0.8:0.2, то есть 17 280 изображений на обучение и 4 520 изображений для проверки на тесте. Каждый объект в выборке считается одним классом, так как на данном этапе главная задача – найти на изображении все коробки с наггетами.

В качестве оборудования для обучения нейронной сети выступают графический процессор Nvidia Geforce 1660 с 6 Гб видеопамати и центральный процессор Intel Core i7 9750 2.6 Гц, оперативная память объемом 16 Гб.

Поскольку за основу была взята модель, предобученная на большом наборе данных COCO, содержащем разнообразные отличающиеся друг от друга объекты, в качестве инициализации нейронной сети использованы веса готовой модели.

Для предотвращения переобучения к данным был применен набор из пяти описанных далее аугментаций – операций по различному искажению изображений, чтобы нейронная сеть не строила зависимости от цвета, на который влияет освещение, не привязывалась к определенному положению на изображении, к определенной гамме, яркости и т.п.

- *Случайное нормальное размытие.* На изображении случайным образом добавляется эффект размытости, чтобы научить работать нейронную сеть в условиях нечеткого отображения.

- *Случайные черные перекрытия.* Черными прямоугольниками заданного размера перекрываются случайные участки объектов целевого класса, чтобы научить сеть обнаруживать объект по частичной видимости, так как даже при его обнаружении отслеживание продолжается до момента завершения операции

захвата, где объект будет перекрываться манипулятором.

- *Случайное выбрасывание цветowych каналов.* Из цветовой палитры RGB случайным образом зануляется один из каналов, тем самым искажая цвет картинки. Данная аугментация позволяет нейронной сети абстрагироваться от цветowych признаков целевого признака и обращать внимание только на его фактуру и форму, иначе кодировка цвета может сильно изменяться при другом освещении.

- *Случайная яркость, насыщенность, оттенков, гамма.* Эти параметры могут отличаться при разных настройках камеры и условиях окружающего пространства. Необходимо абстрагировать от них нейронную сеть, позволяя обнаруживать объекты при различных параметрах гаммы, яркости, насыщенности и оттенка.

- *Случайное смещение ограничивающих рамок.* Рамки, которыми отмечены объекты на изображении, смещаются в случайном направлении на заданное расстояние, чтобы нейронная сеть строила признаки не на композиции всего объекта, а на его отдельных частях. Таким образом, свертки при каждом слое будут обучены искать свой собственный элемент изображения целевого класса.

Размер входа нейронной сети был установлен в 300×300, так как при анализе выборки не были обнаружены достаточно удаленные маленькие объекты, которые тяжело было бы различить при таком малом разрешении. Это позволяет увеличить скорость работы нейронной сети без особых потерь качества.

Размер пакета (входной набор картинок из тренировочной выборки, на основании которых высчитывается градиент) был установлен на значение 4, что является максимумом по вместимости в видеопамати с используемой конфигурацией оборудования.

В качестве функции оптимизации использован метод Adam [15] как один из самых эффективных для оптимизации глубоких нейронных сетей.

Обучение проводилось на протяжении 20 000 итераций.

Для шага градиента был установлен параметр задержки, который уменьшает градиентный шаг в 0.95 раза каждые 5 000 итераций.

Таким образом, была получена модель нейронной сети, способной на входе получать изображение, а на выходе предоставлять координаты целевых объектов – коробок с наггетами.

Обучение классификатора

При полученном участке изображения целевого класса – коробки с наггетсами нужно определить, в каком положении находится содержимое и подлежит ли оно выравниванию.

Это обусловлено особенностями производственного процесса: содержимое попадает в упаковку в скомканном виде, образуя неправильную форму. Неплотно закрытая крышка не позволит на следующих участках конвейерной ленты автоматически закрыть и заклеить коробку. Таким образом, на основе полученных данных и разметок на классы «выровнен» и «не выровнен» была обучена модель.

Данная задача является бинарной классификацией, и для ее решения необходима выборка, содержащая изображения двух классов объектов.

Поскольку для предыдущей задачи детектирования уже были выделены объекты на изображении, необходимо использовать только вырезки из изображений, то есть части изображений, содержащие целевой класс. Эти изображения будут меньшего размера, чем исходные, тем самым позволяя создать нейронную сеть с более маленьким размером входного слоя, что ускорит работу дополнительной нейронной сети. В качестве архитектуры использовалась EfficientNet B0 [16]. Согласно [16], семейство архитектур EfficientNet является одним из самых эффективных решений для задач классификации изображений. Это достигнуто путем нахождения оптимальной архитектуры нейронной сети при помощи другой нейронной сети, которая занимается перебором различных топологий для достижения максимального результата. Чтобы детектор и классификатор работали на максимальной скорости, выбрана самая быстрая версия EfficientNet – B0. Стоит отметить, что выбор аргументирован требованием высокой скорости выполнения поставленной задачи выравнивания, так как объекты двигаются в среднем по 80 коробок в минуту и это требует максимально быстрого отклика на входные данные.

Если модель ошибется и посчитает выровненную коробку невыровненной, это не приведет к фатальным последствиям, просто робот выполнит на одну операцию больше. Поэтому в данной постановке задачи требуются высокая скорость работы и высокая метрика качества – полнота, отвечающая за количество найденных объектов того или иного класса.

Нейронная сеть обучалась на тренировочной выборке размером 22 320 изображений и тестировалась на 13 480 изображениях.

Метод оптимизации – Adam. Размер входного вектора – 224×224. В качестве инициализации весов использованы предобученные веса на изображениях ImageNet, содержащих 1 000 классов. Размер пакета составил 8 изображений. Процесс обучения проходил на протяжении 10 000 итераций. Аугментация для данной задачи – получение 3D-позиции объекта. Так как целевые объекты находятся в горизонтальной плоскости и положение фиксировано вдоль отдельной зоны изображения – конвейерной ленты, для получения 3D-позиции можно воспользоваться стереопарой, где будут просчитываться карты глубины только для отдельных участков изображения. То есть, используя калибровочные параметры стереопары для получения 3D-облака точек, можно получить 3D-положения коробок с наггетсами, зная их 2D-положение на изображении.

Результаты

Были получены два решения поставленной задачи – модель обнаружения коробок с наггетсами на изображении и классификатор выровненных и невыровненных коробок. Для каждой задачи сформулированы решения с использованием искусственных нейронных сетей и выбрана оптимальная архитектура нейронной сети на основе пакета программ OpenVino Toolkit.

Для задачи обнаружения выбрана метрика mAP как наиболее часто используемая для оценки качества в современных эталонных наборах данных [5, 6].

Для оценки классификации подсчитаны метрики качества «точность» и «полнота». Точность отвечает за определение доли правильно распознанных объектов среди всех объектов, отнесенных к данному классу, а полнота – за выделение доли объектов данного класса среди всех распознанных объектов.

В случае определения принадлежности к классу «выровнен» и «не выровнен» наибольший приоритет имеет полнота, так как более серьезной ошибкой будет допустить невыровненную коробку с наггетсами, чем ошибочно причислить выровненную к невыровненной.

Результаты полученных метрик качества приведены в таблице 2.

Полученное решение визуально отображено на рисунке 2. В процессе работы алго-

Таблица 2

Метрики оценки качества для обнаружения и классификации коробок с наггетсами

Table 2

Quality assessment metrics for nugget box detection and classification

Задача	Архитектура	Метрика	Значение метрики, %
Обнаружение коробок с наггетсами на изображении	CenterNet + MobileNetV2	mAP	95
Классификация выровненных и невыровненных коробок на изображении	EfficientNet B0	Точность	89
		Полнота	61

ритма вычисляются координаты прямоугольной рамки, в которой находится объект, его класс, что в дальнейшем добавляет в очередь команду для манипулятора – выровнять его или пропустить, и расстояние до него от конечного органа манипулятора, что дает возможность алгоритмам обратной кинематики вычислить положение данного объекта и выполнить выравнивание.



Рис. 2. Результат распознавания коробки с наггетсами в лабораторных условиях

Fig. 2. The result of recognizing a nugget box in the laboratory

Рассуждения

Построенное решение задачи позволяет эффективно распознавать объекты на конвейерной ленте для последующих вертикальных поступательно-вращательных движений по отношению к ним. Постановка задачи и ее выполнение с использованием искусственных

нейронных сетей уникальны в своем роде. Однако с точки зрения экономической эффективности решение может быть улучшено путем построения алгоритма детектирования 3D-позиции при помощи только одной камеры, что позволит существенно снизить общую стоимость системы технического зрения. На данный момент система способна обрабатывать объект только одного класса – наггетсы, что делает ее узкоспециализированной и не позволяет масштабировать. На основе методологии из [17] данное решение будет расширено для разных объектов с регулированием, чтобы обеспечить его масштабируемость при сохранении характеристик качества.

Описанный метод будет служить фундаментом для решения более общей задачи – обнаружение объектов с нефиксированным количеством классов [17, 18].

Заключение

В данной статье представлено решение уникальной задачи на производстве с использованием промышленных камер компьютерного зрения и современных архитектур искусственных нейронных сетей.

Представленный подход позволяет автоматизировать или сильно упростить рутинный ручной труд на производстве, снизить операционные издержки.

Решение подобной задачи показывает, как в рамках Индустрии 4.0 можно оптимизировать современные производства.

Список литературы

1. Ketelaere B.D., Wouters N., Kalfas I., Belleghem R.V., Saeys W. A fresh look at computer vision for industrial quality control. *Quality Eng.*, 2022, vol. 34, pp. 152–158. doi: 10.1080/08982112.2021.2001828.
2. Hernández S.A., Romero H.J., Morales A.D. A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. *IOP Conf. Ser.: Mater. Sci. Eng.*, 2020, vol. 844, pp. 012–024. doi: 10.1088/1757-899X/844/1/012024.
3. Javaid M., Haleem A., Singh P.R., Rab Sh., Suman R. Exploring impact and features of machine vision for progressive industry 4.0 culture. *Sensors Int.*, 2022, vol. 3, art. 100132. doi: 10.1016/j.sintl.2021.100132.
4. Srivastava Sh., Divekar A.V., Anilkumar Ch., Naik I., Kulkarni V., Pattabiraman V. Comparative analysis of deep learning image detection algorithms. *J. of Big Data*, 2021, vol. 8, art. 66. doi: 10.21203/rs.3.rs-132774/v1.

5. Everingham M., Van Gool L., Williams C.K.I., Winn J., Zisserman A. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2009, vol. 88, pp. 303–338.
6. Lin T.-Y., Maire M., Belongie S., Bourdev L., Girshick R. et al. Microsoft COCO: Common Objects in Context. In: *LNIP. Proc. ECCV*, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.
7. Liu W., Anguelov D., Erhan D. et al. SSD: Single shot multibox detector. In: *LNIP. Proc. ECCV*, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
8. Zhou X., Wang D., Krähenbühl P. Objects as points. *ArXiv*, 2019, art. 1904.07850. URL: <https://arxiv.org/abs/1904.07850> (дата обращения: 03.02.2022).
9. Bochkovskiy A., Wang C.-Y., Mark Liao H.-Y. YOLOv4: Optimal speed and accuracy of object detection. *ArXiv*, 2020, art. 2004.10934. URL: <https://arxiv.org/abs/2004.10934> (дата обращения: 03.02.2022).
10. Howard A.G., Zhu M., Chen B., Kalenichenko D. et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, 2017, art. 1704.04861. URL: <https://arxiv.org/abs/1704.04861> (дата обращения: 03.02.2022).
11. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.-C. MobileNetV2: Inverted residuals and linear bottlenecks. *Proc. CVPR*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
12. Szegedy C., Liu W., Jia J., Sermanet P., Reed S. et al. Going deeper with convolutions. *Proc. CVPR*, 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
13. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition. *Proc. CVPR*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
14. Lin T.-Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Feature pyramid networks for object detection. *Proc. CVPR*, 2017, pp. 2117–2125. doi: 10.1109/CVPR.2017.106.
15. Kingma D.P., Ba J. Adam: A method for stochastic optimization. *ArXiv*, 2017, art. 1412.6980. URL: <https://arxiv.org/abs/1412.6980> (дата обращения: 03.02.2022).
16. Tan M., Le Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. *PMLR*, 2020, pp. 1–10. URL: <https://proceedings.mlr.press/v97/tan19a/tan19a.pdf> (дата обращения: 03.02.2022).
17. Толстель О.В., Ширкин А.Е., Калабин А.Л. Проецирование технических объектов на изображении в метрическое пространство при помощи глубоких нейронных сетей для задачи детектирования // Программные продукты и системы. 2020. Т. 33. № 3. С. 464–470. doi: 10.15827/0236-235X.131.464-470.
18. Тарачков М.В., Ширкин А.Е., Перминов И.К. Разработка элементов системы управления антропоморфным роботом AP-601 // Гибридные и синергетические интеллектуальные системы: матер. IV Всерос. конф. 2018. С. 368–374.

Software & Systems

doi: 10.15827/0236-235X.142.334-341

2023, vol. 36, no. 2, pp. 334–341

**Constructing a computer vision system for aligning the contents of packages
by a delta manipulator in food production**

**Oleg V. Tolstel
Alexandr E. Shirkin
Alexandr L. Kalabin**

For citation

Tolstel, O.V., Shirkin, A.E., Kalabin, A.L. (2023) ‘Constructing a computer vision system for aligning the contents of packages by a delta manipulator in food production’, *Software & Systems*, 36(2), pp. 334–341 (in Russ.). doi: 10.15827/0236-235X.142.334-341

Article info

Received: 02.02.2023

After revision: 21.03.2023

Accepted: 22.03.2023

Abstract. This article focuses on using computer vision to automate the process of aligning the contents of food packages in cartons on a conveyor line. This process is necessary to prevent the appearance of loosely closed packages due to uneven distribution of the contents inside the package. The research object is a combination of a stereo vision system and artificial intelligence algorithms that solve the problem of detecting and positioning an object in a 2D image for alignment with a delta manipulator designed specifically for the task of automating the process of aligning the package contents. The authors have obtained the following results: 1) For the positioning problem, a real-time object detection algorithm was implemented in the image; it broadcasts the detected 2D image sections to the stereo system forming partial depth maps. As a result, a fast and accurate algorithm for obtaining the 3D coordinates of an object was obtained. 2) To determine the current state of the package contents and optimize the number of alignment operations, the authors have implemented a binary classifier that informs the system by two types of statuses: “an object is aligned”, “an object is not aligned”. 3) For this task, the authors have collected a special data set consisting of a sequence of images with a frequency of 30 frames per second during 1 hour. Food contents were marked with rectangular boxes in each image and their state was noted, whether the package was aligned or not. Labeling was done using the open source LabelImg program, which provides a graphical interface for labeling images used in further training. 4) There was a program developed that implements the proposed algorithm in Python 3.6 using the Jupyter Lab integrated environment in Ubuntu 18.04 operating system. The paper presents the results of an experiment on using the proposed algorithm for estimating the 2D position of an object and the current state of the detected object. To evaluate the algorithm quality, the authors used the metric of the mean average precision (mAP) of

object detection, as well as classification metrics - accuracy and recall for the task of determining the status of the package contents.

Keywords: artificial neural networks, object detection, binary classification, quality control, 3D pose estimation, grasping, robotic manipulators

Reference List

1. Ketelaere, B.D., Wouters, N., Kalfas, I., Belleghem, R.V., Saeys, W. (2022) 'A fresh look at computer vision for industrial quality control', *Quality Eng.*, 34, pp. 152–158. doi: 10.1080/08982112.2021.2001828.
2. Hernández, S.A., Romero, H.J., Morales, A.D. (2020) 'A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework', *IOP Conf. Ser.: Mater. Sci. Eng.*, 844, pp. 012–024. doi: 10.1088/1757-899X/844/1/012024.
3. Javaid, M., Haleem, A., Singh, P.R., Rab, Sh., Suman, R. (2022) 'Exploring impact and features of machine vision for progressive industry 4.0 culture', *Sensors Int.*, 3, art. 100132. doi: 10.1016/j.sintl.2021.100132.
4. Srivastava, Sh., Divekar, A.V., Anilkumar, Ch., Naik, I., Kulkarni, V., Pattabiraman, V. (2021) 'Comparative analysis of deep learning image detection algorithms', *J. of Big Data*, 8, art. 66. doi: 10.21203/rs.3.rs-132774/v1.
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A. (2009) 'The PASCAL Visual Object Classes (VOC) Challenge', *IJCV*, 88, pp. 303–3388.
6. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R. et al. (2014) 'Microsoft COCO: Common Objects in Context', in *LNIP. Proc. ECCV*, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.
7. Liu, W., Anguelov, D., Erhan, D. et al. (2016) 'SSD: Single shot multibox detector', in *LNIP. Proc. ECCV*, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
8. Zhou, X., Wang, D., Krähenbühl, P. (2019) 'Objects as points', *ArXiv*, art. 1904.07850, available at: <https://arxiv.org/abs/1904.07850> (accessed February 03, 2022).
9. Bochkovskiy, A., Wang, C.-Y., Mark Liao, H.-Y. (2020) 'YOLOv4: Optimal speed and accuracy of object detection', *ArXiv*, art. 2004.10934, available at: <https://arxiv.org/abs/2004.10934> (accessed February 03, 2022).
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D. et al. (2017) 'MobileNets: Efficient convolutional neural networks for mobile vision applications', *ArXiv*, art. 1704.04861, available at: <https://arxiv.org/abs/1704.04861> (accessed February 03, 2022).
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018) 'MobileNetV2: Inverted residuals and linear bottlenecks', *Proc. CVPR*, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.
12. Szegedy, C., Liu, W., Jia, J., Sermanet, P., Reed, S. et al. (2015) 'Going deeper with convolutions', *Proc. CVPR*, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.
13. He, K., Zhang, X., Ren, S., Sun, J. (2016) 'Deep residual learning for image recognition', *Proc. CVPR*, pp. 770–778. doi: 10.1109/CVPR.2016.90.
14. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie S. (2017) 'Feature pyramid networks for object detection', *Proc. CVPR*, pp. 2117–2125. doi: 10.1109/CVPR.2017.106.
15. Kingma, D.P., Ba, J. (2017) 'Adam: A method for stochastic optimization', *ArXiv*, art. 1412.6980, available at: <https://arxiv.org/abs/1412.6980> (accessed February 03, 2022).
16. Tan, M., Le, Q.V. (2020) 'EfficientNet: Rethinking model scaling for convolutional neural networks', *PMLR*, pp. 1–10, available at: <https://proceedings.mlr.press/v97/tan19a/tan19a.pdf> (accessed February 03, 2022).
17. Tolstel, O.V., Shirkin, A.E., Kalabin, A.L. (2020) 'Technical object projection in the image into metric space using deep neural networks for the detection problem', *Software & Systems*, 33(3), pp. 464–470. doi: 10.15827/0236-235X.131.464-470 (in Russ.).
18. Tarachkov, M.V., Shirkin, A.E., Perminov, I.K. (2018) 'Development of elements of the control system anthropomorphic robot AR-601', *Proc. IV Conf. Hybrid and Synergetic Intelligence Systems*, pp. 368–374 (in Russ.).

Авторы

Толстель Олег Владимирович¹,
к.т.н., доцент, tolstel.oleg@mail.ru
Ширкин Александр Евгеньевич¹,
ассистент, shirkin39@mail.ru
Калабин Александр Леонидович²,
д.ф.-м.н., профессор,
akalabin@yandex.ru

¹ Балтийский федеральный университет им. Иммануила Канта, Институт физико-математических наук и информационных технологий, г. Калининград, 236041, Россия

² Тверской государственный технический университет, г. Тверь, 170026, Россия

Authors

Oleg V. Tolstel¹, Ph.D. (Engineering),
Associate Professor, tolstel.oleg@mail.ru
Alexandr E. Shirkin¹, Assistant,
shirkin39@mail.ru
Alexandr L. Kalabin²,
Dr.Sc. (Physics and Mathematics),
Professor, akalabin@yandex.ru

¹ Baltic Federal University. Immanuel Kant, Institute of Physical and Mathematical Sciences and Information Technologies, Kaliningrad, 236041, Russian Federation

² Tver State Technical University, Tver, 170026, Russian Federation