

ПРИМЕНЕНИЕ ИНСТРУМЕНТОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОМЫШЛЕННЫХ ПРОИЗВОДСТВ

А.Ю. Кайда, А.О. Савельев
Томский политехнический университет
ayk13@tpu.ru

Введение

На сегодняшний день цифровизация промышленных производств является актуальной задачей, требующей современных подходов. Автоматизация процесса в некоторых ключевых отраслях требует дополнительного применения таких технологий, как компьютерное зрение. В данной статье описывается идея и реализация приложения по распознаванию объекта на плоскости, которой, предположительно, служит конвейерная лента. Прототип приложения является самостоятельным модулем, предназначенным для работы в связке с программным обеспечением для робота KUKA в пищевой промышленности.

Концепция приложения

Потребность в использовании компьютерного зрения возникла и в отрасли пищевой промышленности, с использованием роботов KUKA [1]. Данный робот представляет собой подобие механической руки для захвата объекта, в данном случае, на конвейере. Задачей приложения является выявление требуемых объектов, движущихся в двух измерениях, внешний вид которых строго задан заранее пользователем. Конечным результатом должна стать прототип (proof of concept) REST-приложения, реализованный на языке программирования Python с использованием библиотеки OpenCV [2]. Приложение рассматривается как самостоятельно существующий модуль, не ориентированный на конкретную конфигурацию робота. Пользователь имеет возможность наблюдения за тем, как распознаются объекты определенного вида, выделяемые рамкой с центром объекта. Вместе с тем в качестве выходных данных передаются координаты объекта для последующего использования, чтобы задать движения манипулятора.

Текущая задача не предполагает использования алгоритмов аналогичных YOLO [3], способных распознавать любой объект определенной группы в силу того, что любой объект и его внешний вид – четко заданы, сам объект не вращается в трех измерениях, а расположение камеры и ее расстояние от поверхности конвейера являются фиксированными.

Было рассмотрено три алгоритма, потенциально применимых в качестве базовых для распознавания объектов:

- SIFT [4] – алгоритм масштабно-инвариантной трансформации признаков.
- SURF [5] – ускоренный алгоритм распознавания устойчивых признаков.
- ORB [6] – рекомбинация ранних дескрипторов BRIEF [7] и FAST [8].

В ходе рассмотрения данных алгоритмов, опираясь на предыдущие исследования [9], было отмечено, что SIFT имеет наивысший показатель сопоставимости точек двух изображений, в связи с чем было принято решение о его использовании.

SIFT выделяет локальные признаки изображений. Первоначально извлекаются ключевые точки (минимальное заданное значение – 10). Объект распознается путем сравнения каждого признака нового изображения с набором существующих признаков на основе евклидова расстояния между векторами признаков. Если говорить о принципиальном отличии на практике, то YOLO эффективно использовать в случае необходимости классифицировать случайный объект. YOLO классифицирует объекты по предварительно обученной модели, деля изображение на фрагменты, в то время как SIFT производит сравнение по точкам (признакам) на двух конкретных изображениях.

Данный алгоритм эффективен при вращении в двух плоскостях без отдаления, а также при отклонении на малые углы.

Для тестирования использовались камеры высокого разрешения 1920 x 1080 пикселей. Результаты распознавания представлены на рисунках 1 и 2.



Рис. 1. Пример распознавания объекта в потоковом видео

Анализ результатов показал, что распознавание объекта возможно на расстоянии до 0.9 м, что является достаточным согласно первичным требованиям. Первый прототип производит измерения с частотой 1 fps. Распознавание является возможным при движении объекта со скоростью не более 20 м/с, открытой поверхности объекта >70% и угле отклонения <25°.

Данное приложение поддерживает возможность загрузки нового изображения объекта пользователем, используемое для распознавания в потоковом видео. Таким образом, достаточно одного

изображения высокого разрешения.



Рис. 2. Пример распознавания объекта с изображением (объект частично скрыт)

Заключение

В настоящее время проводится тестирование на базе учебной лаборатории «Technobothnia» [10] при ассоциации университетов г.Вааса, Финляндия. Запланировано проведение тестов для различных объектов. Изменяемыми параметрами являются яркость освещения помещения, расстояние от камеры до объекта, угол отклонения и скорость перемещения. Первая демонстрация рабочего механизма прототипа без пользовательского интерфейса проведена успешно. Вторая и третья демонстрации запланированы до конца декабря 2018 г. Планируется внедрение приложения для дальнейшего использования в лаборатории «Technobothnia» для студенческих работ.

Список использованных источников

1. Промышленные роботы | KUKA AG [Электронный ресурс]. – Режим доступа: <https://www.kuka.com/ru-ru/>. – (Дата обращения: 21.11.2018).
2. OpenCV Library [Электронный ресурс]. – Режим доступа: <https://opencv.org/>. – (Дата обращения: 21.11.2018).
3. YOLO: Real-Time Object Detection [Электронный ресурс] – Режим доступа:

<https://pjreddie.com/darknet/yolo/>. – (Дата обращения: 21.11.2018).

4. Introduction to SIFT (Scale-Invariant Feature Transform) Open-CV Python Tutorials [Электронный ресурс] – Режим доступа: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html. – (Дата обращения: 21.11.2018).
5. Introduction to SURF (Speeded-Up Robust Features) Open-CV Python Tutorials [Электронный ресурс] – Режим доступа: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html – (Дата обращения: 21.11.2018).
6. ORB (Oriented FAST and Rotated BRIEF) Open-CV Python Tutorials [Электронный ресурс] – Режим доступа: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_orb/py_orb.html – (Дата обращения: 21.11.2018).
7. BRIEF (Binary Robust Independent Elementary Features) Open-CV Python Tutorials [Электронный ресурс] – Режим доступа: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_brief/py_brief.html – (Дата обращения: 21.11.2018).
8. FAST Algorithm for Corner Detection – Open-CV Python Tutorials [Электронный ресурс] – Режим доступа: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html – (Дата обращения: 21.11.2018).
9. E.Karami, S. Prasad, and M.d Shehata, Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images [Электронный ресурс] – Режим доступа: <https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf> – (Дата обращения: 21.11.2018).
10. Welcome to Technobothnia. [Электронный ресурс]. – [Электронный ресурс] – Режим доступа: <http://www.technobothnia.fi/home/> – (Дата обращения: 21.11.2018).