**«Санкт-Петербургский государственный электротехнический университет**

**«ЛЭТИ» им. В.И.Ульянова (Ленина)»**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | 09.04.04 – Программная инженерия | |
| **Программа** | Разработка распределенных программных систем | |
| **Факультет** | КТИ | |
| **Кафедра** | МО ЭВМ | |
| *К защите допустить* |  | |
| Зав. кафедрой |  | Кринкин К. В. |

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

магистра

Тема: Разработка алгоритма распознавания объекта и определения его координат по изображению

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Левшин П. И. |
|  |  |  |  |  |
| Руководитель |  |  |  | Чернокульский В. В. |
|  |  |  |  |  |
| Консультанты | к.э.н., доцент |  |  | Ичкитидзе Ю. Р. |
|  |  |  |  |  |
|  | к.т.н., доцент |  |  | Родионов С. В. |
|  |  |  |  |  |

Санкт-Петербург

2019**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой МО ЭВМ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Кринкин К. В. |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | Левшин П. И. | | | |  | Группа | 3303 | |
| Тема работы: Разработка алгоритма распознавания объекта и определения его координат по изображению | | | | | | | | |
| Место выполнения ВКР: СПБГЭТУ «ЛЭТИ», кафедра МО ЭВМ | | | | | | | | |
| Исходные данные (технические требования): Ubuntu, python, OpenCV | | | | | | | | |
| Содержание ВКР: аналитическая часть, моделирование компонентов разрабатываемого ПО, результаты разработки, составление бизнес-плана по коммерциализации результатов НИР магистранта | | | | | | | | |
| Перечень отчетных материалов: пояснительная записка, иллюстративный материал | | | | | | | | |
| Дополнительные разделы: Составление бизнес-плана по коммерциализации результатов НИР магистранта | | | | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | | | | |
| «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | | | |
| Студент | |  | | Левшин П. И. | | | |
| Руководитель | |  | | Чернокульский В. В. | | | |
|  | |  | |  | | | |
| Консультанты к.э.н., доцент | |  | | Ичкитидзе Ю. Р. | | | |
|  | |  | |  | | | |
| к.т.н., доцент | |  | | Родионов С. В. | | | |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой МО ЭВМ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Кринкин К. В. |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | Левшин П. И. |  | Группа | 3303 |
| Тема работы: Разработка алгоритма распознавания объекта и определения его координат по изображению | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 06.02 – 07.03 |
| 2 | Аналитическая часть | 07.03 – 21.03 |
| 3 | Моделирование компонентов разрабатываемого ПО | 07.03 – 04.04 |
| 4 | Результаты разработки | 04.04 – 18.04 |
| 5 | Составление бизнес-плана по коммерциализации результатов НИР магистранта | 18.04 – 02.05 |
| 6 | Оформление пояснительной записки | 02.05 – 18.05 |
| 7 | Оформление иллюстративного материала | 18.05 – 22.05 |

|  |  |  |
| --- | --- | --- |
| Студент |  | Левшин П. И. |
| Руководитель |  | Чернокульский В. В. |
|  |  |  |

**РЕФЕРАТ**

**ABSTRACT**

**СОДЕРЖАНИЕ**

[Определения, обозначения и сокращения 9](#_Toc8772295)

[Введение 10](#_Toc8772296)

[1.1. Мотивация к разработке программного обеспечения 11](#_Toc8772297)

[1.2. Требования предъявляемые к программному обеспечению 12](#_Toc8772298)

[1.3. Анализ требований 13](#_Toc8772299)

[1.3.1. Язык программирования 13](#_Toc8772300)

[1.3.2. Структура программного обеспечения 14](#_Toc8772301)

[1.3.3. Выявления характеристик продукции и его окружения на изображении для задачи поиска 15](#_Toc8772302)

[1.3.2. Выявление характеристик продукции для задачи классифицирования 17](#_Toc8772303)

[1.4. Исследование возможностей контурного анализа в OpenCV 18](#_Toc8772304)

[1.4.1 Подготовка изображения 19](#_Toc8772305)

[1.4.2 Применение функции findContours и поиск прямоугольников минимальной площади 20](#_Toc8772306)

[1.4.3 Получение прямоугольных областей с исходного изображения 23](#_Toc8772307)

[1.4.4. Алгоритм поиска прямоугольных объектов с помощью библиотеки OpenCV 24](#_Toc8772308)

[1.5. Целесообразность использования нейронных сетей в контексте рассматриваемой задачи 25](#_Toc8772309)

[1.6. Алгоритм классификации объектов 26](#_Toc8772310)

[1.6.1. Распознавание штрихкода на изображении 28](#_Toc8772311)

[1.6.2. Вычисление цветового признака изображения 32](#_Toc8772312)

[1.6.3. Сравнение вычисленного цветового признака с эталонными значениями классов 33](#_Toc8772313)

[1.6.4. Сравнение признака размера с эталонными значениями классов 34](#_Toc8772314)

[1.7. Окончательная структура программного обеспечения 35](#_Toc8772315)

[1.8. Выводы по разделу 36](#_Toc8772316)

[2. моделирование компонентов разрабатываемого ПО 37](#_Toc8772317)

[2.1. Построение модели алгоритма поиска объектов на изображении 37](#_Toc8772318)

[2.2. Построение модели алгоритма выделения прямоугольной области с изображения 38](#_Toc8772319)

[2.3. Построение моделей алгоритмов вычисления признаков по изображению 38](#_Toc8772320)

[2.5. Построение модели алгоритма классификации объектов 38](#_Toc8772321)

[2.6. Построение модели алгоритма создания и загрузки конфигурации программного обеспечения 38](#_Toc8772322)

[2.7. Выводы по разделу 38](#_Toc8772323)

[3. Результаты разработки 39](#_Toc8772324)

[4. Составление бизнес-плана по коммерциализации результатов НИР 40](#_Toc8772325)

[4.1. Резюме 40](#_Toc8772326)

[4.2. Описание продукции 40](#_Toc8772327)

[4.2.1. Характеристики продукции 41](#_Toc8772328)

[4.2.2. Потребительские свойства 42](#_Toc8772329)

[4.2.3. Конкурентные преимущества продукции 43](#_Toc8772330)

[4.3. Анализ рынка сбыта 43](#_Toc8772331)

[4.3.1. Основные потребители 43](#_Toc8772332)

[4.4. План производства 43](#_Toc8772333)

[4.4.1. Расчет расходов на оплату труда 43](#_Toc8772334)

[4.4.2. Расчет накладных расходов 46](#_Toc8772335)

[4.4.3. Расходы по статье «Материалы» 46](#_Toc8772336)

[4.4.4. Издержки на амортизацию ПК и оргтехники 47](#_Toc8772337)

[4.4.5. Прочие прямые расходы 48](#_Toc8772338)

[4.4.6. Себестоимость выполнения ВКР 49](#_Toc8772339)

[4.4.7. Объем продаж 49](#_Toc8772340)

[4.5 Финансовый план 49](#_Toc8772341)

[4.5.1. План прибылей и убытков 49](#_Toc8772342)

[4.6. Выводы по разделу 51](#_Toc8772343)

[Заключение 52](#_Toc8772344)

[Список использованных источников 53](#_Toc8772345)

Определения, обозначения и сокращения

В настоящей пояснительной записке применяют следующие термины с соответствующими определениями:

Вендинг (англ. Vending) — это продажа товаров и услуг с помощью автоматизированных систем (торговых автоматов).

ROS (Robot Operating System) – это гибкая основа для написания программного обеспечения робота. Это набор инструментов, библиотек и соглашений, направленных на упрощение задачи создания сложного и надежного поведения роботов на самых разных роботизированных платформах. [1]

Снек – это в англоязычных странах общее название лёгких блюд, предназначенных для утоления голода между основными приёмами пищи.

EAN (European Article Number) — европейский стандарт штрихкода, предназначенный для кодирования идентификатора товара и производителя

Введение

выв1. Аналитический раздел

1.1. Мотивация к разработке программного обеспечения

Создаваемое программное обеспечение разрабатывается по запросу компании, занимающейся обслуживанием вендинговых автоматов.

В обслуживание входит: контроль за количеством продукции, загрузка новой продукции и выгрузка непроданной продукции из вендинговых автоматов.

В качестве продукции выступают снеки или шоколадные батончики.

После выгрузки непроданной продукции из вендингового автомата, ее необходимо сортировать по наименованию продукции. В данный момент для сортировки продукции используется ручной труд. Компания планирует провести автоматизацию данного процесса, заменив ручной труд на конвейер с роботом-манипулятором с фотокамерой.

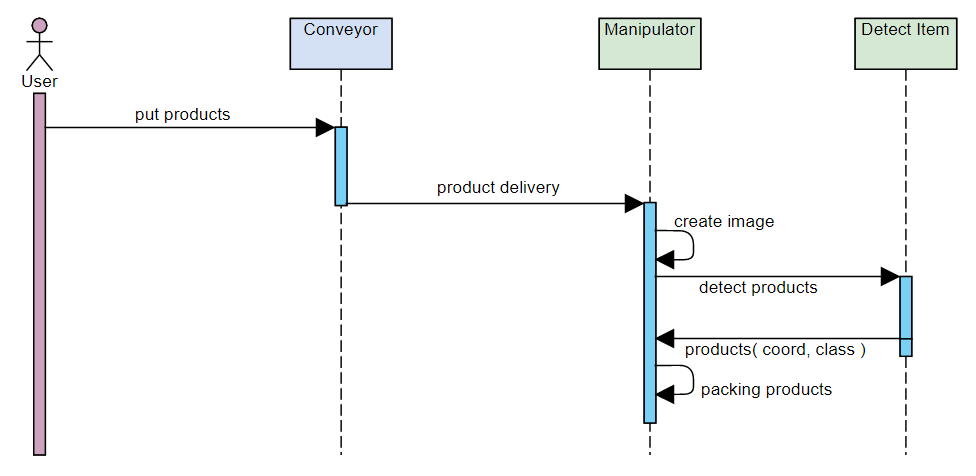
На рис. 1.1 изображена диаграмма последовательности автоматизированного процесса сортировки продукции. 

Рисунок 1.1 – Диаграмма последовательности автоматизированного процесса сортировки продукции

Рабочий загружает продукцию на конвейер, который доставляет продукцию к роботу-манипулятору. Робот-манипулятор, оснащенный фотокамерой, получает изображение конвейера с расположенными на нем продуктами. Далее изображение должно быть обработано для получения координат продуктов и их классификацию для дальнейшей расфасовки их роботом-манипулятором.

Под классами подразумеваются конкретные наименования продуктов, “Snickers Super”, “Bounty Trio”, “Kitkat” и так далее.

В данной работе описывается разработка алгоритма детектирования продукции на изображении и их классификация.

1.2. Требования предъявляемые к программному обеспечению

К разрабатываемому программному продукту предъявлены следующие требования:

* Программное обеспечение должно быть реализовано на одном из языков программирования, поддерживаемом операционной системой для управления роботами ROS.
* Обработка изображения должна быть произведена с помощью библиотеки OpenCV.
* Программное обеспечение должно быть оформлено в виде библиотеки, содержащей следующие основные модули:
  + модуль определения координат продукта на изображении;
  + модуль классификации найденных на изображении продуктов.
* Проанализировать продукты и выявить признаки, определяющие тот или иной продут.
* Программное обеспечение должно иметь возможность добавления новых классов продуктов, не требующую длительной подготовки для анализа.
* Программный продукт должен иметь возможность логирования хода обработки изображения и сохранения промежуточных этапов обработки на файловую систему.
* Программное обеспечение должно иметь следующие возможности настройки:
  + должна быть возможность использовать каждый модуль отдельно от других или совместно;
  + при классификации продуктов должна быть возможность указать какие признаки анализировать, а какие нет;
  + должна быть возможность включения/отключения логирования хода обработки изображения.
* Программа должна тщательно быть задокументирована.

1.3. Анализ требований

В данном разделе будет проведен анализ требований к программному обеспечению.

1.3.1. Язык программирования

Основными языками программирования разработки под ROS являются python и C++. Для них написано большое количество модулей, а также они хорошо задокументированы.

В табл. 1.1 сведены плюсы и минусы использования языков python и C++. [2]

На основе изложенных выше плюсов и минусов был выбран язык python, так как удобство и скорость разработки предпочтительнее.

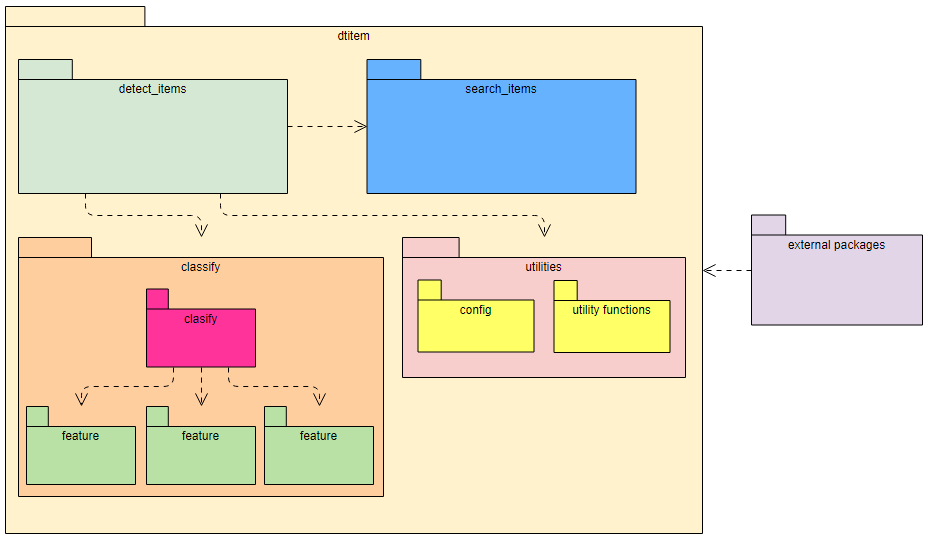
Таблица 1.1. – Анализ языков программирования ROS

|  |  |  |
| --- | --- | --- |
| **Метод** | **Языки** | |
| **python** | **C++** |
| Плюсы | * Быстрое построение прототипа * Быстрое изучение * В среднем программы короче * Проще прочитать и понять исходный код | * Работает намного быстрее * Нахождение ошибок на этапе компиляции |
| Общее | Оба имеют большое количество библиотек | |
| Минусы | * Работает медленнее * Нахождение ошибок на этапе выполнения | * Намного сложнее в изучении * Сложнее понять код |

1.3.2. Структура программного обеспечения

В соответствии с требованиями программное обеспечение должно быть оформлено в виде библиотеки, поделенной на модули.

Так как был выбран язык python, то программное обеспечение будет оформлено в виде пакета. На рис. 1.2 представлена внутренняя структура пакета.

  
Рисунок 1.2 – Внутренняя структура пакета

На представленной выше диаграмме пакетов можно увидеть четыре основные модуля:

* search\_items – модуль выделения продуктов на изображении;
* classify – модуль классификации продуктов. Модуль содержит подмодуль classify, который производит классификацию продуктов на основе признаков, выявленных с помощью подмодулей feature;
* utilities – модуль, содержащий подмодуль config, отвечающий за возможность настройки библиотеки и модуль utility\_functions, который содержит вспомогательные функции;
* detect\_items – модуль, позволяющий запустить процесс обработки изображения с возможность настройки.

1.3.3. Выявления характеристик продукции и его окружения на изображении для задачи поиска

В качестве продуктов выступают снеки. Основное свойство, объединяющее их от других объектов – это их прямоугольная форма, которая удобна для упаковки данной продукции в коробки для транспортировки.

На рис. 1.3 представлен пример распознаваемого продукта.

Рисунок 1.3 – Пример распознаваемого продукта

Так как робот-манипулятор фотографирует продукцию на конвейере, то фон изображению будет однотонным и прямоугольные объекты на таком фоне будут хорошо различимы, что позволит их легче отделить от основного изображения.

Также важно понимать, что в данном случаем ось фотокамеры всегда перпендикулярна плоскости конвейера, что не влечет за собой искажение формы объектов.

На рис 1.4 представлен пример обрабатываемого изображения.

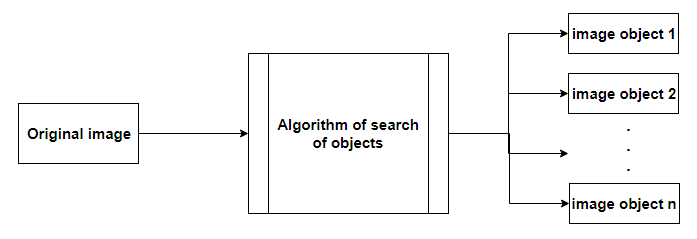
Таким образом, можно сделать вывод, что задача поиска объектов на изображении сводится к поиску прямоугольников, т.к. объекты других форм на изображения попасть не могут.

Рисунок 1.4 – Внутренняя структура пакета

Исходя из того, что задача сводится к поиску объектов определенной формы, то целесообразно использовать методы контурного анализа.

Методы контурного анализа будут рассмотрены далее.

На рис. 1.5 представлена схема, определяющая входные и выходные данные алгоритма поиска объектов на изображении. Входом для алгоритма является изображение с объектами, а выходом – прямоугольные фрагменты минимальной площади, полученные из исходного изображения, каждый из которых полностью вмещает в себя каждый найденный объект.

Рисунок 1.5 – входные и выходные данные алгоритма поиска объектов

1.3.2. Выявление характеристик продукции для задачи классифицирования

В данной задаче классификации классами выступают конкретные наименования продуктов.

Для задачи классификации объектов необходимо выделить признаки, которые отличают друг от друга.

Основной признак каждого продукта это штрихкод, который представлен в формате EAN13, представляющем собой тринадцатиразрядную цифровую последовательность. [3] Штрихкод можно назвать идентификатором того или иного продукта и четко сопоставляется с тем или иным классом (наименованием).

Следующие два признака, отличающие продукты друг относительно друга имеют визуальных характер – это цвет и размер (ширина, длина).

Каждый класс, которым может быть отмечен объект, должен иметь в себе эталонную информацию о штрихкоде, цвете, размерах.

В свою очередь, для каждого объекта, который должен быть классифицирован, необходимо вычислить их значения признаков.

Таким образом задача классификации сводится к сравнению вычисленных признаков объекта с эталонными признаками каждого класса.

На рис. 1.6 представлена схема, определяющая входные и выходные данные алгоритма классификации.

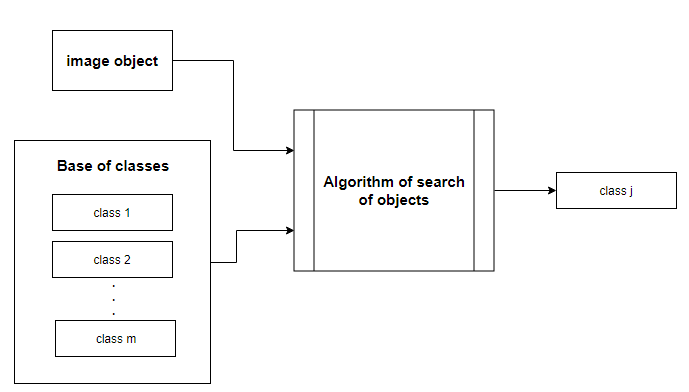


Рисунок 1.6 – входные и выходные данные алгоритма классификации объектов

1.4. Исследование возможностей контурного анализа в OpenCV

Согласно анализу обрабатываемых изображений, рассматриваемому в разделе 1.3.3., было выявлено, что искомые объекты имеют прямоугольную форму и располагаются на однотонном фоне, что позволяет применить алгоритмы контурного анализа.

Контурный анализ заключается в поиске контуров объектов на изображении.

Контурный анализ является совокупностью методов выделения, описания и преобразования контуров изображений. Форма полностью определяется контуром объекта на изображения. Контур содержит всю необходимую информацию для распознавания изображений по их форме. [4]

OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях — распространяется в условиях лицензии BSD.

Библиотека OpenCV имеет в своем арсенале функцию findContours. [5] Функция принимает на вход три основных аргумента: изображение, режим группировки и метод упаковки.

Так как алгоритм также должен вычислить координаты объектов, то на вход алгоритму также передается коэффициент масштабирования, определяющий соотношение длин на изображении с реальными длинами объектов. В качестве координат объекта будет пониматься центр этого объекта, что в точности соответствует центру найденного прямоугольника.

Обобщенный ход алгоритма при применении данной функции:

1. Подготовить изображение.
2. Применить функцию findContours.
3. Для каждого найденного контура определить прямоугольник минимальной площади, который полностью его вмещает.
4. Вырезать из исходного изображения области, определенные найденными прямоугольниками.
5. Вычисление реальных размеров и центра объекта умножением размеров вырезанных областей на коэффициент масштаба.

1.4.1 Подготовка изображения

Для применения функции findContours, необходимо подготовить изображение, так как функция принимает на вход монохромное изображение, которое содержит в себе информацию о границах объектов на изображении, которые функция в последствии объединит в контуры объектов.

Границы – это линии на изображении, в которых происходит резкое изменение яркости.

Чтобы получить монохромное изображения необходимо использовать оператор Кэнни.

Оператор Кэнни в дисциплине компьютерного зрения — оператор обнаружения границ изображения. Был разработан в 1986 году Джоном Кэнни и использует многоступенчатый алгоритм для обнаружения широкого спектра границ в изображениях.

Подготовка изображения:

1. Сжать изображение для уменьшения количества обрабатываемой информации.
2. Преобразовать изображение их трехканальной цветовой модели RGB в одноканальную.
3. Применить оператор Кэнни к одноканальному изображению для получения монохромного изображения, содержащее информацию о границах объектов.

1.4.2 Применение функции findContours и поиск прямоугольников минимальной площади

Помимо изображения функция принимает на вход еще два аргумента: режим группировки, метод упаковки.

Режим группировки, отвечает за то, какие границы искать и то, как найденные границы группируются друг относительно друга. Этот параметр может принимать следующие значения:

* CV\_RETR\_LIST — выдаёт все контуры без группировки;
* CV\_RETR\_EXTERNAL — выдаёт только крайние внешние контуры. Например, если в кадре будет пончик, то функция вернет его внешнюю границу без дырки;
* CV\_RETR\_CCOMP — группирует контуры в двухуровневую иерархию. На верхнем уровне — внешние контуры объекта. На втором уровне — контуры отверстий, если таковые имеются. Все остальные контуры попадают на верхний уровень;
* CV\_RETR\_TREE — группирует контуры в многоуровневую иерархию.

Учитывая специфику задачи, необходимо выбрать значение CV\_RETR\_EXTERNAL, так как важны только внешние границы искомых объектов и не важна их внутренняя структура.

Метод упаковки, отвечает за то, как контуры хранятся в памяти, хранится ли каждая точка контура или происходит упаковка точек для уменьшения количестао памяти, занимаемой контуром. Этот параметр может принимать следующие значения:

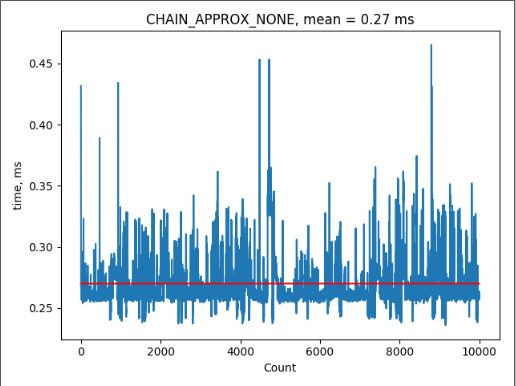
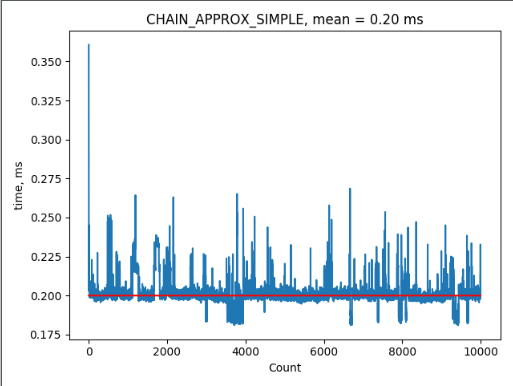
* CV\_CHAIN\_APPROX\_NONE — упаковка отсутствует и все контуры хранятся в виде отрезков, состоящих из двух пикселей;
* CV\_CHAIN\_APPROX\_SIMPLE — склеивает все горизонтальные, вертикальные и диагональные контуры;
* CV\_CHAIN\_APPROX\_TC89\_L1, CV\_CHAIN\_APPROX\_TC89\_KCOS — применяет к контурам метод упаковки Teh-Chin.

Данный параметр напрямую влияет на поиск минимального прямоугольника, так как на этом шаге идет перебор всех точек контура. С одной стороны значения для третьего параметра CV\_CHAIN\_APPROX\_TC89\_L1, CV\_CHAIN\_APPROX\_TC89\_KCOS позволят получить контуры с минимальным количество точек, но в тоже время требуют значительных преобразования исходных контуров, что может повлиять на производительность.

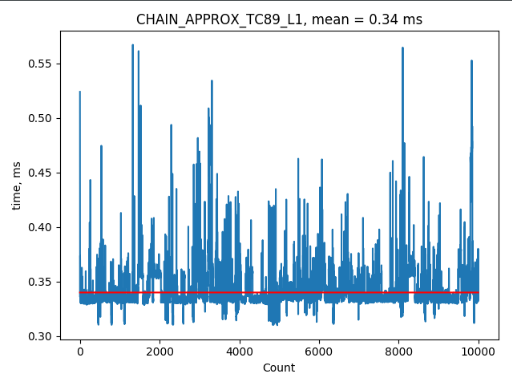
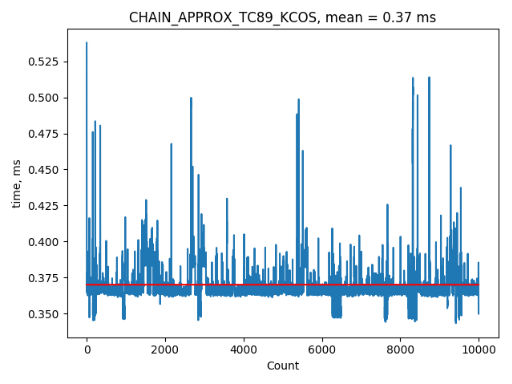
Для определения оптимального значения параметра был проведен эксперимент. Эксперимент состоял из:

* Подготовка изображения к поиску контуров согласно разделу 1.4.1.
* Последовательное применение функций findContours и minAreaRect, так же входящей в состав библиотеки OpenCV. Были проведены замеры времени работы этих функций для каждого значения параметра метода упаковки.

Результаты эксперимента показаны на рис. 1.5. Согласно результатам эксперимента самым оптимальным является значение CV\_CHAIN\_APPROX\_SIMPLE. Результаты объясняются тем, что при значении CV\_CHAIN\_APPROX\_NONE функция findContours не упаковывает полученные контуры и из-за этого функция minAreaRect перебирает большое количество точек, в тоже время при значениях параметра CV\_CHAIN\_APPROX\_TC89\_L1, CV\_CHAIN\_APPROX\_TC89\_KCOS функция тратит большое количество времени на преобразование полученных контуров.

(a) (b)

(c) (d)

Рисунок 1.7 – Результаты эксперимента по выбору оптимального параметра упаковки контуров: (a) CHAIN\_APPROX\_NONE,

(b) CHAIN\_APPROX\_SIMPLE, (c) CV\_CHAIN\_APPROX\_TC89\_L1,

(d) CV\_CHAIN\_APPROX\_TC89\_KCOS

1.4.3 Получение прямоугольных областей с исходного изображения

Одним из последних этапов поиска объектов на изображении является выделение прямоугольных областей, полученных на предыдущем шаге.

Так как объекты на изображении могут быть расположены не параллельно сторонам изображение, то необходимо применить к исходному изображению аффинные преобразования.

Аффинное преобразование – отображение плоскости или пространства в себя, при котором параллельные прямые переходят в параллельные прямые, пересекающиеся — в пересекающиеся, скрещивающиеся — в скрещивающиеся.

Необходимые нам аффинные преобразования: параллельный перенос и поворот вокруг точки.

С предыдущего шага алгоритма нам известен центр прямоугольника, а также угол поворота относительно горизонтальной оси.

Пусть центр прямоугольника, высота и ширина прямоугольника соответственно, а также угол поворота относительно горизонтальной оси.

Пусть ширина и высота изображения соответственно.

Для применения аффинных преобразований стоит рассмотреть так называемые однородные точки. Если есть точка , то ее однородная это . Для удобства .

Первый шаг – это параллельный перенос. Необходимо перенести центр прямоугольника в начало координат на вектор . Тогда параллельный перенос любой точки в однородных координатах производится по формуле (1.1)

(1.1)

Следующий шаг – поворот на заданный угол. Необходимо повернуть изображения относительно начала координат на угол . Поворот любой точки относительно начала координат в однородных координатах производится по формуле (1.2).

(1.2)

Общую формулу преобразования можно получить, объединив формулы (1.1) и (1.2). Общая формула с дополнительным переносом в центр координат показана в формуле (1.3).

(1.3)

Тогда точке в однородных координатах соответствует точка, определенная формулой (1.4).

(1.4)

Произведя соответствующие преобразование остается только вырезать прямоугольную область размера с центром в точке .

1.4.4. Алгоритм поиска прямоугольных объектов с помощью библиотеки OpenCV

Исходя из разделов 1.4.1 и 1.4.2 был сформирован алгоритм поиска прямоугольных объектов:

1. Сжать изображение для уменьшения количества обрабатываемой информации.
2. Преобразовать исходное изображение из трехканальной цветовой модели RGB в одноканальную цветовую модель.
3. Применение оператора Кэнни.
4. Применение функции findContours c параметрами CV\_RETR\_EXTERNAL и CV\_CHAIN\_APPROX\_SIMPLE.
5. Применение функции minAreaRect для нахождения прямоугольников минимальной площади, вмещающих в себя найденные контуры.
6. Вырезать из исходного изображения области, определенные найденными прямоугольниками.
7. Вычисление реальных размеров и центра объекта умножением размеров вырезанных областей на коэффициент масштаба.

1.5. Целесообразность использования нейронных сетей в контексте рассматриваемой задачи

Искусственная нейронная сеть — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

На данный момент это один из самых популярных подходов в области обработки изображений. Нейронные сети позволяют качественно классифицировать объекты на изображении по заранее обученным классам.

В данной задаче, поставленной фирмой вендинговых автоматов, количество классов на которые надо будет натренировать нейронную сеть боле двадцати, что ведет к огромным затратам на подготовку данных и обучение.

Также важным моментом является то, что часто могут появляется новые продукты для автоматов, что ведет к появлению новых классов и соответственно требует заново обучать нейронную сеть.

Таким образ выявлено, что использование нейронных сетей нецелесообразно.

1.6. Алгоритм классификации объектов

В соответствии с анализом характеристик классифицируемых объектов, как показано в разделе 1.3.2, основными признаками, определяющими класс, являются: штрихкод, цвет, размер.

Данные признаки не равносильны, так как штрихкод является признаком, четко определяющим тот или иной класс продукта, но из-за человеческого фактора объект может быть расположен на конвейере штрихкодом вниз, что не позволит его распознать.

Для таких случаев необходимо провести сравнения цвета и размеров объекта, которые в совокупности являются хорошими признаки, отличающими класс от другого, но следует учитывать, что цвет является более важным, чем размер.

Чтобы продемонстрировать, что признаки цвета и размера не равносильны, был проведен эксперимент. Были взяты объекты, являющиеся представителями самых популярных марок и проведены вычисление их размера и цвета. Затем были вычислены коэффициенты вариации для обоих признаков.

Вариация — различие значений какого-либо признака у разных единиц совокупности (набора).

Коэффициент вариации — это отношение среднеквадратического отклонения к среднеарифметическому, рассчитывается в процентах. Необходим для сравнения вариаций наборов, измеряемых разными шкалами. [6]

Вычисление коэффициентов вариации производится по формуле (1.5).

, (1.5)

где – коэффициент вариации признака ;

– среднеквадратичное отклонение признака;

– среднее значение признака .

Результаты эксперимента сведены в таблиц 1.2.

Среднеквадратичные отклонения цвета и размера вычислены по формуле (1.6).

, (1.6)

где – среднеквадратичное отклонение признака;

– количество объектов в наборе;

– значение признака объекта ;

– среднее значение признака .

Средние значения признаков цвета и размера вычислены по формуле (1.7).

, (1.7)

где – среднее значение признака .

– количество объектов в наборе;

– значение признака объекта ;

По результатам эксперимента можно выявить, что коэффициент вариации цвета выше, чем коэффициент вариации размера, что показывает, что разброс цвета исследуемых продуктов выше, чем разброс размера объектов.

Таблица 1.2. – Результаты эксперимента по вычислению коэффициента

|  |  |  |
| --- | --- | --- |
| **Продукт** | **Признаки** | |
| **Цвет (R, G, B)** | **Размер (ширина, длина), см** |
| Snickers Super | (88; 92; 107) | (18,2; 4,5) |
| Twix | (52; 75; 111) | (16; 8) |
| Twix Max | (75; 97; 131) | (18,5; 8) |

Продолжение таблицы 1.2

|  |  |  |
| --- | --- | --- |
| KitKat | (91; 92; 134) | (14; 4) |
| KitKat Maxi | (98; 101; 136) | (18; 4) |
| Bounty | (131; 129; 126) | (16; 4) |
| Mars Max | (102; 116; 127) | (18,5; 4,5) |
|  | (91; 100; 124) | (17,03; 5,29) |
|  | 64,34 | 2,37 |
|  | 34,71 | 13,39 |

Таким образом между признаками были расставлены приоритеты: Штрихкод, цвет, размер.

Алгоритм сводится к последовательному вычислению признаков и сравнению их с эталонными значениями.

Так как штрихкод является идентификатором продукта, то при его успешном распознавании вычисление остальных признаков нецелесообразно. Если штрихкод распознать не удалось, то следует учитывать цвет и размер.

1.6.1. Распознавание штрихкода на изображении

Алгоритм распознавания штрихкода основан на форме штрихкода. Все продукты маркируются штрихкодов стандарта EAN13, который кодируется 13 цифрами.

Данный штрих код является линейным, состоящим из параллельных линий разной толщины с разным промежутками между ними. На рис. 1.8 представлен пример штрихкода одного из исследуемых продуктов.

Преобразование самого штрихкода в цифровую последовательность будет производиться с помощью библиотеки pyzbar, одной из самых популярных библиотек по распознаванию штрихкодов различных типов.



Рисунок 1.8 – Штрихкод снека “SNICKERS SUPER”

На рис. 1.9 представлен пример изображения, поступающего на обработку.

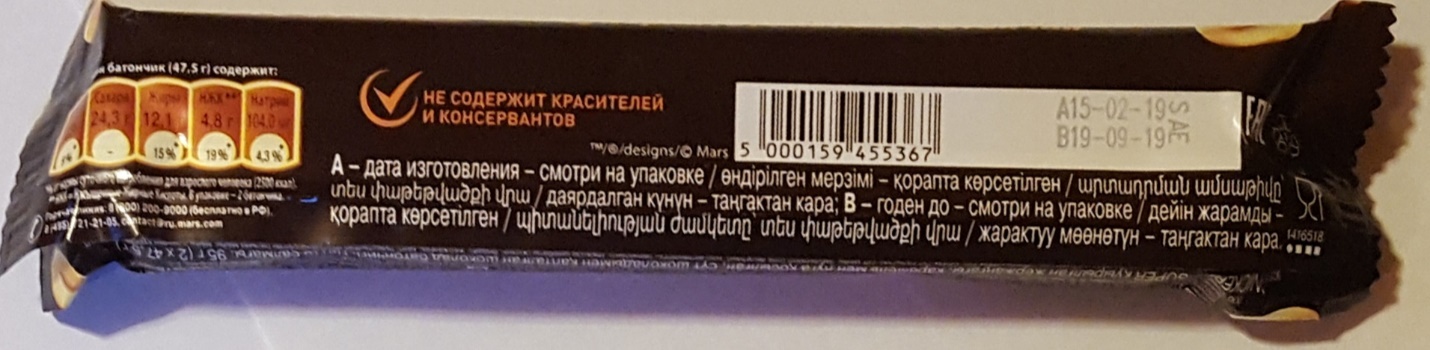


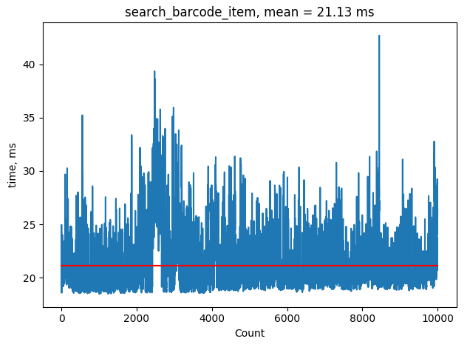
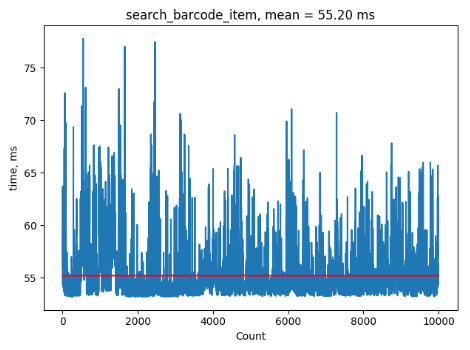
Рисунок 1.9 – Штрихкод снека “SNICKERS SUPER”

Для уменьшения времени распознавания штрихкода библиотекой pyzbar, его необходимо выделить с изображения. Данное решение принято на основе эксперимента, сравнения производительности распознавания по исходному изображению и по выделенному изображению. Результаты эксперимента отображены на рис. 1.10.

Проанализировав множество изображений, полученных после алгоритма поиска, был сделан вывод, что штрихкод всегда располагается параллельно одной из осей. Что позволяет применить оператор Собеля.

Оператор Собеля — дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения.

Данный оператор необходимо применить к одноканальному изображению, поэтому исходное изображение необходимо преобразовать из трехканальной цветовой модели RGB в одноканальную модель.

(a) (b)

Рисунок 1.10 – Результаты сравнительного эксперимента

(а) c выделением штрихкода, (b) без выделения штрихкода

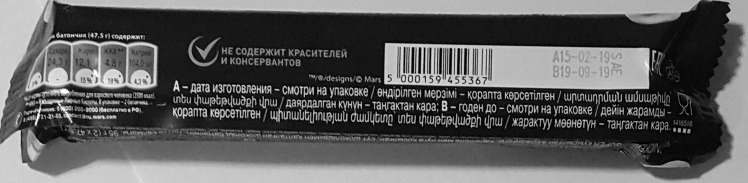
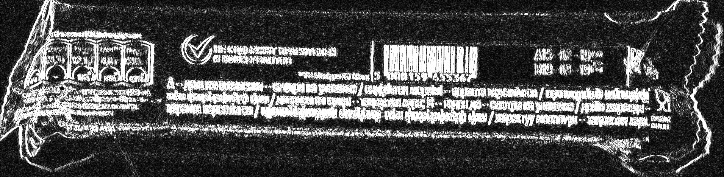
Данный оператор необходимо применить к одноканальному изображению, поэтому исходное изображение необходимо преобразовать из трехканальной цветовой модели RGB в одноканальную модель.

Далее необходимо убрать шум на изображении с помощью функции threshold, которая обнуляет точки изображения, значение которых меньше некоторого заданного значения.

На рис. 1.11 представлены преобразования цветовой модели и применение оператора Собеля над изображением с рис. 1.9.

Следующим шагом необходимо убрать лишние тонкие элементы с изображения. Для этого выполняются морфологические преобразования:

* Закрытие. Позволяет объединить близко стоящие фрагменты белого цвета, что позволит объединить полосы, из которых состоит штрихкод, в один большой общий блок
* Эрозия. Позволяет уменьшить фрагменты белого цвета, в результате чего большинство мелких элементов пропадет с изображения.
* Дилатация. Позволяет увеличить фрагменты белого цвета, в результате чего область штрихкода вернет свои изначальные размера, которые были до операции эрозии.



(a) (b)

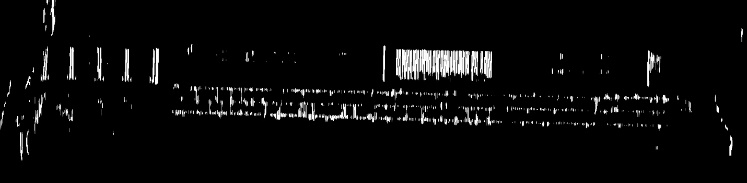
Рисунок 1.11 – Преобразования изображения:

(a) в одноканальную модель, (b) применение оператора Собеля

Данные преобразования позволят убрать тонкие элементы с изображения, в тоже время оставив область штрихкода неизменной.

Последним шагом необходимо выполнить поиск контуров и прямоугольников минимальной площади, аналогично разделу 1.4. На изображении могут остаться несколько таких прямоугольников. Необходимо найти прямоугольник с максимальной площадью.

Пример преобразований над изображением с рис. 1.11(b) представлен на рис. 1.12.

(a) (b)

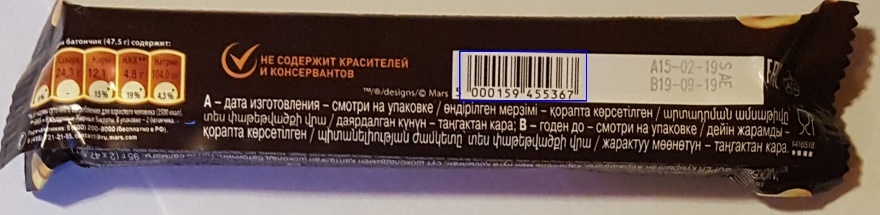
(c)

Рисунок 1.12 – Преобразования изображения:

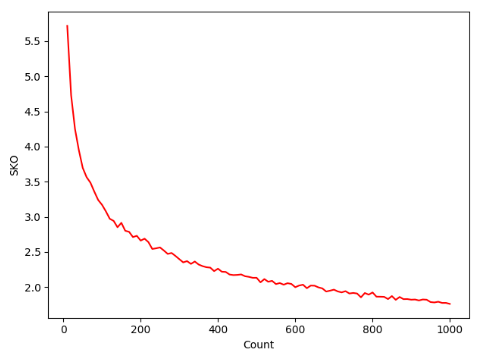
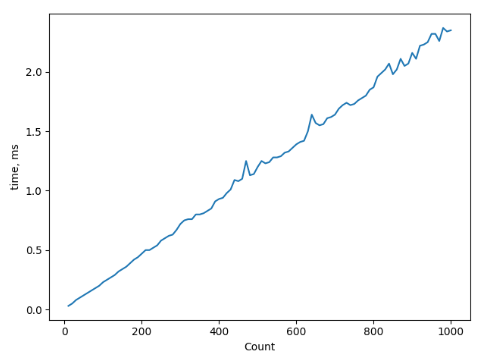
(a) фильтрация шума, (b) морфологические преобразования, (c) результат

1.6.2. Вычисление цветового признака изображения

Вычисление цветового признака производится на основе вычисления среднего значения цветов точек, взятых с изображения случайным образом с равномерным распределением.

Для получения количества точек, которых необходимо подсчитать для достижения достаточной точности и производительности вычислений был поставлен эксперимент.

В эксперименте были проведены расчеты времени вычисления и дисперсии получаемых цветов для количества точек от 10 до 10000. Дисперсия цветового признака рассчитана по формуле (1.6). Результаты эксперимента продемонстрированы на рис. 1.13.

(a) (b)

Рисунок 1.13 – Подбор количества точек в расчете цвета:

(a) зависимость среднеквадратичного отклонения от количества точек

(b) зависимость времени выполнения вычисления от количества точек

Эксперимент показал, что время вычислений растет линейно, а дисперсия падет по гиперболе, что означает что нет смысла в количестве большем 1000.

1.6.3. Сравнение вычисленного цветового признака с эталонными значениями классов

Рассмотрев значение цветового признака, как точку в трехмерной пространстве, введем меру близости цветов. Мера близости рассчитывается, как декартово расстояние по формуле (1.8).

(1.8)

где – мера близости;

– значение соответствующего канал цвета.

Пусть имеется набор классов , каждый из которых имеет свое эталонное значение цветового параметра .

Пусть имеется вычисленное по изображению объекта значение цветового признака.

Тогда номер класса для объекта на изображении вычисляется по формуле (1.9)

(1.9)

Таким образом производится классификация по цветовому признаку.

Также стоит учитывать, что могут быть классы, которые имеют близкие значения цветового признака, например для продуктов “Kitkat” и “Kitkat Super”. При таком случае необходима дополнительная проверка признака размера. Для того чтобы точно ли объект соответствует классу с минимальным значением меры близости, необходимо вычислить коэффициенты похожести вычисленные значения для каждого класса.

Имея множество декартовых расстояний полученных для каждого класса по формуле (1.8), можно получить коэффициенты масштабирования, показывающие во сколько раз мера близости того или иного класса выше минимальной меры близости, соответствующей классу , посчитанному по формуле (1.9). Коэффициент масштабирования рассчитывается по формуле (1.10).

, (1.10)

где – коэффициент масштабирования;

– значение меры близости класса ;

– минимальное значение меры близости, соответствующее классу .

Заметим, что для класса значение коэффициента масштабирования равно 1;

Имея множество коэффициентов масштабирования для каждого класса, можно вычислить коэффициенты похожести решив систему уравнений (1.11):

(1.11)

Решение уравнения (1.11) представлено в системе (1.12).

(1.12)

Таким образом были вычислены коэффициенты похожести цветового признака относительно эталонных цветовых признаков каждого класса .

Наибольший коэффициент похожести соответствует классу с наименьшей мерой близости.

Взяв отношение двух самых больших коэффициентов, можно принять решение, необходима ли проверка признака размера, сравнив его с заранее заданным пороговым значением.

1.6.4. Сравнение признака размера с эталонными значениями классов

Для признака размера аналогично цветовому признаку, рассмотренному в разделе 1.5.3, вводится мера близости, которая рассчитывается в двумерном евклидовом пространстве по формуле (1.13).

(1.13)

где – мера близости;

– ширина, высота объекта.

Пусть – эталонный признак размера для класса, а значение признака класса, вычисленного по изображению.

Тогда номер лучшего класса определяется по формуле (1.9), а коэффициента похожести по формулам (1.10) и (1.12).

1.7. Окончательная **структура программного обеспечения**

В соответствии с требованиями, предъявленными к программному обеспечению и анализом задач поиска и классификации, была доработана структура программного обеспечения и разработан алгоритм использования данного ПО. Структура программного обеспечения показана на рис. 1.14.

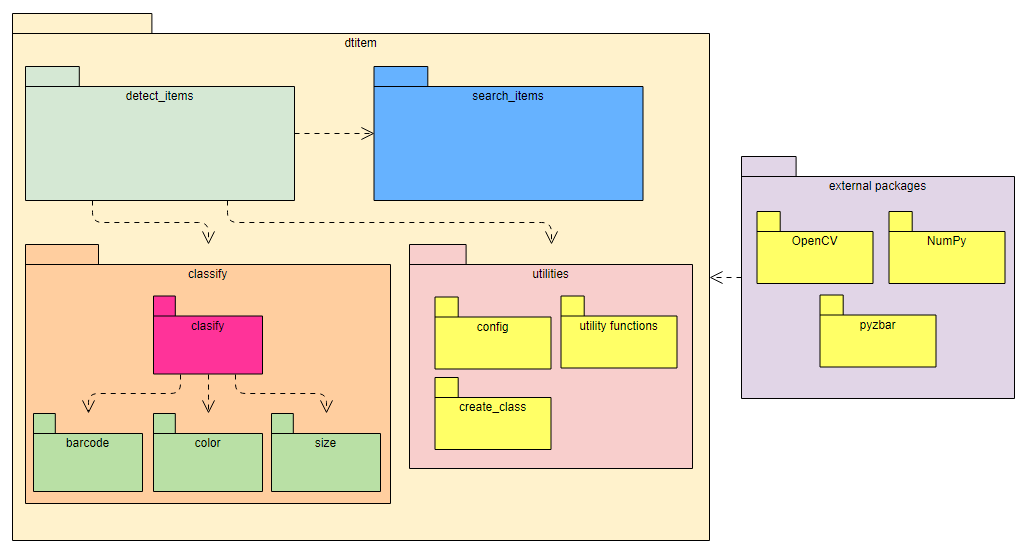


Рисунок 1.14 – Структура программного обеспечения

Алгоритм использование разрабатываемого программного обеспечения:

1. Подготовка стенда. Заключается в установке фотокамеры в месте, в котором она будет использоваться.
2. Подготовка классов продуктов. Заключается в создании базы классов, каждый из которых содержит: наименование, штрихкод, цвет, размеры.
3. Использование по назначению, описанному в разделе 1.1

1.8. Выводы по разделу

В данном разделе были проанализированы технические требования к разрабатываемому программному обеспечению.

Был проведен анализ изображений, поступающих на обработку, из которого был сделан вывод, что оптимальным решением задачи поиска объектов на изображениях, будет контурный анализ, основанный на форме искомых объектов.

Также были проанализированы исходные объекты для выявления признаков, отличающих их друг от друга для задачи классификации. Были выявлены следующие признаки: штрихкод, цвет, размер. Для каждого из признаков были разработаны алгоритмы их вычисления.

В итоге была разработана структура разрабатываемого программного обеспечения и алгоритм его использования.

2. моделирование компонентов разрабатываемого ПО

2.1. Построение модели алгоритма поиска объектов на изображении

На основе анализа, проведенного в разделе 1.4, была разработана модель действия алгоритма поиска объектов прямоугольной формы на изображении. На рис. 2.1 представлена диаграмма действий алгоритма поиска объектов на изображении.

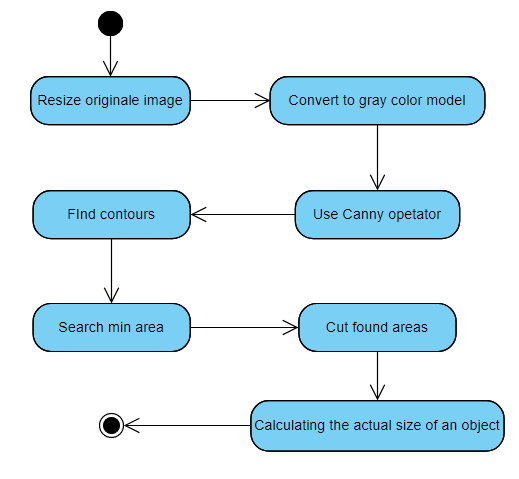


Рисунок 2.1 – Диаграмма действий алгоритма поиска объектов на изображении

Алгоритм принимает на вход:

* исходное изображение для анализа;
* параметры, зависящие от фотокамеры:
  + коэффициент сжатия;
  + минимальное и максимальное пороговые значения оператора Кэнни;
  + минимальный порог площади прямоугольника;

2.2. Построение модели алгоритма выделения прямоугольной области с изображения

2.3. Построение моделей алгоритмов вычисления признаков по изображению

2.5. Построение модели алгоритма классификации объектов

На основе анализа, проведенного в разделе 1.6, была разработана модель действия алгоритма классификации объектов по трем признакам: штрихкод, цвет, размер. На рис. 2.2 представлена диаграмма действий алгоритма классификации объектов.

НАРИСОВАТЬ ЕБУЧУЮ ДИАГРАММУ

Рисунок 2.2 – Диаграмма действий алгоритма классификации объектов

2.6. Построение модели алгоритма создания и загрузки конфигурации программного обеспечения

2.7. Выводы по разделу

3. Результаты разработки

# 4. Составление бизнес-плана по коммерциализации результатов НИР

4.1. Резюме

Выпускная квалификационная работа описывает анализ и разработку программного продукта позволяющего определять местоположения объекта на изображении и идентификации продукта питания (снеки, батончики).

Бизнес-план предназначен для экономического обоснования разработанного программного обеспечения:

Были рассчитаны полные затраты на проект и прибыль за первой год.

Стоимость единицы продукции – 60000 руб.

В первом квартале продажи продукции не производится. В нем идет только разработка программного продукта.

В следующие 3 квартала производятся продажа продукции.

4.2. Описание продукции

В качестве продукции выступает библиотека для языка python включающая в себя модули, позволяющие определять местоположения объекта на изображении и идентифицировать продукт питания (снеки, батончики).

Согласно ГОСТ 19781 – 90:

Прикладная программа - Программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации. [23]

Таким образом продукция является прикладным программным обеспечением, нацеленным на автоматизацию обработки изображений различных продуктов питания.

4.2.1. Характеристики продукции

Согласно ГОСТ Р ИСО/МЭК 9126 – 93 программная продукция должна иметь следующие характеристики [24]:

1. Функциональные требования

Данная продукция имеет следующие функциональные возможности:

* Определение координат центра масс продукта и его угол поворота относительно оси абсцисс.
* Классификация продукта методом сравнение размеров продукта
* Классификация продукта методом сравнения цвета продукта
* Классификация продукта методом поиска штрих-кода
* Классификация продукта с использованием обученной нейросети.

1. Практичность

Разработанная программное обеспечение является практичным за счет ее деление на модули, в соответствии с функциональными возможностями).

Каждый модуль может использоваться отдельно от других или в совокупности для достижения более точных результатов.

Каждый модуль имеет настройки по умолчанию, с возможностью внесения изменений с помощью речного ввода настроек или с помощью конфигурационного файла.

1. Надежность

Разработанное программное обеспечение было протестировано с помощью:

* Функциональное тестирование (метод черного ящика). Согласно проведенному тестированию выявлено, что все модули соответствуют своему функциональному требованию с учетом их ограничений.

Ограничения модулей описаны в разделе 1.3.

* Тестирование производительности. Результаты тестирования производительности приведены в разделе 3.5.1.
* Модульное тестирование. Результаты модульного тестирование приведены в разделе 3.5.2.

1. Эффективность

Программное обеспечение разработано для быстрого и удобного поиска продуктов питания (Снеком, Батончиков) с возможностью настройки конфигурации его модулей.

1. Сопровождаемость

Сопровождаемость достигается за счет деления библиотеки на модули, что позволяет упростить разработку новых модулей при появлении новых функциональных требований, а также за счет наличия документации.

1. Мобильность

Программный продукт разработан на языке python, что позволяет использовать на любых операционных системах, для которых реализован интерпретатор python

4.2.2. Потребительские свойства

Потребитель при использовании разработанного программного продукта уменьшает затраты на выпуск своего ПО из-за отсутствия необходимости разрабатывать сове решение с нуля и возможности настройки конфигурации модулей разработанного программного продукта.

4.2.3. Конкурентные преимущества продукции

На данный момент нет реализованных комплексных программных продуктов, которые имели бы тот же функционал и возможность комбинации результатов модулей.

4.3. Анализ рынка сбыта

4.3.1. Основные потребители

Основными потребителя являются компании, которым по тем или иным причинам проводить анализ изображения для определения типа продукции. Например, компании обслуживающие автоматы с батончиками, для которых необходимо проводить сортировка батончиков или снеков, идущих по конвейеру.

4.4. План производства

В данном разделе производится расчёт себестоимости продукции и объема продаж

4.4.1. Расчет расходов на оплату труда

На основе трудоемкости выполнения работ по разработке системы рассчитываются издержки на оплату труда ее исполнителей, являющиеся одной из основных статей калькуляции себестоимости разработки.

В процессе разработки организационного плана работ определяется перечень мероприятий по реализации проекта, оценка трудоемкости работ в человеко-днях. Ставка заработной платы в день определяется как отношение средней заработной в месяц по данным к числу рабочих дней в месяце (21 день):

Сайт russia.trud.com проанализировал состояние заработных плат у стажеров-программистов. По состоянию на 20.04.19, по профессии Программист стажер в России открыто 983 вакансий. Для 49.4% открытых вакансий, работодатели указали заработную плату в размере 0 - 22 000 руб. 35% объявлений с зарплатой 22 000 - 44 000 руб., 9.9% с зарплатой 44 000 - 66 000 руб., и 5.6% с зарплатой более 88 000 руб.

Возьмем среднюю зарплату, как взвешенную сумму зарплат по процентам открытых вакансий (4.1):

Возьмем, что у стажера заработная плата изначальна будет составлять 25200 руб. Тогда ставка заработной платы в день рассчитывается по формуле (4.2).

(4.2)

На основе приложения 1 к приказу от 03.05.2017 №1372 «Размеры должностных окладов ППС с учетом повышающих квалификационных коэффициентов», размер заработной платы доцента с наличием ученой степени кандидата наук составляет 22 200 рублей [9].

Таким образом, ставка заработной платы в день рассчитывается по формуле (4.3):

(4.3)

Планы работ разработчика и руководителя представлены в табл. 4.1. и табл. 4.2. соответственно.

Таблица 4.1. – План работ разработчика

|  |  |  |
| --- | --- | --- |
| **Этапы и содержание выполняемых работ** | **Трудоемкость, чел./день.** | **Ставка, руб./день** |
| Разработка ТЗ | 4 | 1302,71 |
| Анализ предметной области | 5 | 1302,71 |
| Разработка методов выделения объекта на изображении и их классифицирования | 10 | 1302,71 |
| Разработка приложения | 50 | 1302,71 |
| Отладка и тестирование программного обеспечения | 10 | 1302,71 |
| Согласование с руководителем и уточнение недочетов | 3 | 1302,71 |
| Создание технико-экономического обоснования | 4 | 1302,71 |
| Оформление пояснительной записки | 10 | 1302,71 |
| Сдача проекта | 1 | 1302,71 |

Таблица 4.2. – План работ руководителя

|  |  |  |
| --- | --- | --- |
| **Этапы и содержание выполняемых работ** | **Трудоемкость, чел./день** | **Ставка, руб./день** |
| Разработка ТЗ | 2 | 1057,14 |
| Анализ предметной области | - | 1057,14 |
| Разработка методов выделения объекта на изображении и их классифицирования | 2 | 1057,14 |
| Согласование с руководителем и уточнение недочетов | 6 | 1057,14 |
| Сдача проекта | 1 | 1057,14 |

В основе определения заработной платы непосредственных исполнителей является определение трудоемкости работ (табл. 4.1. и табл. 4.2.), связанных с разработкой ВКР.

Основная и дополнительная заработная плата исполнителей разработки, т.е. руководителя и разработчика, рассчитываются на основании следующих данных:

• Трудоемкость выполнения работ разработчика по табл. 4.1.

• Трудоемкость выполнения работ руководителя по табл. 4.2.

• Ставка разработчика и ставка руководителя указаны в табл. 4.1. и табл. 4.2. соответственно

• Процент дополнительной заработной платы 12%

• Процент отчислений на социальные нужды 30,2%

Основная заработная плата исполнителей рассчитывается по формуле (4.4).

(4.4)

Дополнительная заработная плата составляет 12% от основной заработной платы рассчитывается по формуле (4.5):

(4.5)

Отчисления на социальные нужды составляют 30,2% от суммы основной и дополнительной заработной платы вычисляется по формуле (4.6).

(4.6)

4.4.2. Расчет накладных расходов

В рамках данной работы накладные расходы устанавливаются в размере 40% от суммы основной и дополнительной заработной платы рассчитывается по формуле (4.7).

(4.7)

4.4.3. Расходы по статье «Материалы»

Расчет количества и стоимости материалов с учетом транспортно-заготовительных расходов: на статью «Материалы» относятся расходы на основные и вспомогательные материалы и комплектующие изделия, которые могут понадобиться при выполнении разработки. Калькуляция расходов по статье «Материалы» с учетом транспортно-заготовительных расходов приведена в табл. 4.3.

Стоимость материалов определена по данным, находящимся на сайте ulmart.ru.

Таблица 4.3. – расходы по статье «Материалы»

|  |  |  |  |
| --- | --- | --- | --- |
| **Материалы** | **Кол-во** | **Цена, руб.** | **Сумма, руб.** |
| Бумага для печати, пачка | 1 | 210 | 210 |
| Картриджи для принтера  Canon G3400 (зеленый, синий, красный, черный) | 1 | 1960 | 1960 |
| Канцелярские товары |  |  | 300 |
| ИТОГО: | | | 2470 |
| Транспортные расходы (15%) | | | 370.5 |
| ВСЕГО: | | | 2840,5 |

4.4.4. Издержки на амортизацию ПК и оргтехники

Амортизация оборудования (в данном случае это ПК и принтер) определяется линейным методом [10]. За год сумма амортизации вычисляется по формуле (4.8).

(4.8)

где стоимость оборудования;

норма амортизации.

Стоимость используемого ноутбука 70 000 руб.

Стоимость принтера Canon G3400 11 720 руб.

Данная стоимость принтера была взята с сайта sebevdom.ru, а стоимость используемых ПК (ноутбук и компьютер) с сайта ulmart.ru.

Рассчитаем норму амортизации. Согласно постановлению правительства РФ от 01.01.2002 N 1 (ред. от 07.07.2016) "О Классификации основных средств, включаемых в амортизационные группы" персональные компьютеры и печатающие устройства, относятся к второй группе, то есть имуществу со сроком полезного действия от двух до трех лет [11]. Считая, что оборудование надежное, выбран срок три года.

Тогда норма амортизации рассчитывается по формуле (4.9).

(4.9)

Рассчитаем годовую сумму амортизации при условии, что использовались ноутбук, ПК и один принтер:

За рабочий день сумма амортизации рассчитывается по формуле (4.10).

(4.10)

где количество рабочих дней в 2019 году, равное 247 дней

Амортизация оборудования за время проектирования ВКР рассчитывается по формуле (4.11):

(4.11)

4.4.5. Прочие прямые расходы

Смета по статье “прочие прямые расходы”, включающая затраты, связанные с приобретением специальной научно-технической литературы, а также платежи за использование средств связи и коммуникации приведены в табл. 4.4.

Таблица 4.4. – прочие прямые расходы

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование** | **Кол-во** | **Цена, руб.** | **Сумма, руб.** |
| Доступ в интернет, месяц | 3 | 650 | 1950 |
| ИТОГО: | | | 1950 |

4.4.6. Себестоимость выполнения ВКР

Себестоимость выполнения ВКР в целом представлена в табл. 4.5.

Таблица 4.5. – Смета затрат на ВКР

|  |  |
| --- | --- |
| **Статья затрат** | **Сумма, руб.** |
| Материалы | 2 840,5 |
| Расходы на оплату труда | 137 991,41 |
| Дополнительная заработная плата | 16 558,96 |
| Отчисления на социальные нужды | 46 674,21 |
| Издержки на амортизацию ПК и оргтехники. | 10 696,42 |
| Прочие прямые расходы | 1950 |
| Накладные расходы | 61 820,15 |
| ИТОГО: | 278 531,65 |

4.4.7. Объем продаж

Программный продукт было решено продавать по квартальной подписке стоимостью 60 000 р. Чтобы получить первую прибыль во втором квартале необходимо, с учетом себестоимости продукции, продавать 5 единиц продукции в квартал.

4.5 Финансовый план

4.5.1. План прибылей и убытков

План прибылей и убытков представляет собой финансовый документ, в котором отражаются доходы, расходы и финансовые результаты за период реализации проекта. Основная задача формирования данного документа состоит в том, чтобы рассчитать величину и структуру себестоимости продукции, соотношение затрат и результатов деятельности за определенный период, по которым можно будет судить о рентабельности проекта, его окупаемости.

План прибылей и убытков составлен в соответствии с формулами (4.12), (4.13), (4.14), (4.15), (4.16).

* Выручка-нетто (без учета НДС) от реализации:

(4.12)

где V – объем реализованной продукции;

P – цена за единицу продукции.

* Валовая прибыль ()

(4.13)

где В – выручка-нетто;

С – себестоимость продукции.

* Прибыль от продаж

(4.14)

где валовая прибыль;

– коммерческие расходы;

управленческие расходы.

* Прибыль до налогообложения

(4.15)

где – прибыль от продаж;

ДО и РО - доходы и расходы от обычных видов деятельности;

прочие доходы и расходы.

* Чистая прибыль

(4.16)

где – прибыль до налогообложения;

Н – налог на прибыль.

Таблица 4.6. – План прибылей и убытков

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Показатели** | **Квартал** | | | | **Всего** |
| **I** | **II** | **III** | **IV** |
| Объем продаж | 0 | 5 | 7 | 10 | 22 |
| Стоимость продукции | 0 | 60 000 | 60 000 | 60 000 | 180 000 |
| Выручка-нетто (без учета НДС) от реализации, руб. | 0 | 254 237,28 | 355932,20 | 508474,57 | 1118644,06 |
| Валовая прибыль, руб. | (278 531,65) | 254 237,28 | 355932,20 | 508474,57 | 840112,41 |
| Прибыль от продаж, руб. | (278 531,65) | 254 237,28 | 355932,20 | 508474,57 | 840112,41 |
| Прибыль до налогообложения, руб. | (278 531,65) | 254 237,28 | 355932,20 | 508474,57 | 840112,41 |
| Налог на прибыль | 0 | 17796,61 | 24915,25 | 35593,22 | 78305,08 |
| Чистая прибыль, руб. | (278 531,65) | 236440,67 | 331016,93 | 472881,35 | 1040338,97 |

Налог на прибыль взят 7%, т.к. форма организации подразумевается ИП.

4.6. Выводы по разделу

В результате работ по экономическому обоснованию ВКР была определена сумма, необходимая для разработки программного продукта, а также был рассчитан план прибылей и убытков.

Себестоимость продукции составляет 278 531,65 руб.

Чистая прибыль за первый год 1040338,97 руб.

Заключение

Список использованных источников

1. About ROS. [Электронный ресурс] // https://www.ros.org/ официальный сайт Open Source Robotics Foundation: https://www.ros.org/about-ros/ (дата обращения: 07.02.2019).
2. Should I learn ROS with Python or with C++? [Электронный ресурс]// http://www.theconstructsim.com/ сайт обучения и прикладной разработки под ROS: http://www.theconstructsim.com/learn-ros-python-or-cpp/ (дата обращения: 07.02.2019).
3. ГОСТ ISO/IEC 15420-2010. Межгосударственный стандарт. Автоматическая идентификация. Кодирование штриховое. Спецификация символики штрихового кода EAN/UPC – Москва: Изд-во стандартов, 2011. – 56 с.
4. Сакович И.О., Белов Ю.С. Обзор основных методов контурного анализа

для выделения контуров движущихся объектов. Инженерный журнал: наука и

инновации, 2014, вып. 12. URL: http://engjournal.ru/catalog/it/hidden/1280.html

1. Structural Analysis and Shape Descriptors [Электронный ресурс] // https://docs.opencv.org/ официальный сайт документации библиотеки компьютерного зрения OpenCV. (дата обращения: 09.02.2019).
2. Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику. — М. : Издательство ЛКИ, 2010.
3. ГОСТ 19781 – 90. Обеспечение систем обработки информации программное. Термины и определения. – Москва: Изд-во стандартов, 1992. – 45 с.
4. ГОСТ Р ИСО/МЭК 9126 – 93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению. – Москва: Изд-во стандартов, 1994. – 17 с.
5. Размеры должностных окладов ППС с учетом повышающих квалификационных коэффициентов: Приложение 1 к приказу № 1372 от 03.05.2017: о повышении уровня оплаты труда отдельных категорий работников университета
6. Налоговый кодекс Российской Федерации (часть вторая) от 05.08.2000 № 117-ФЗ (ред. от 03.04.2017) // Собрание законодательства Российской Федерации. 07.08.2000. N 32. Ст. 259.1. С. 309 – 310.
7. О Классификации основных средств, включаемых в амортизационные группы: постановление Правительства Рос. Федерации от 01.01.2002 N 1 (ред. от 07.07.2016) // Собрание законодательства Российской Федерации. 07.07.2016. N 640. С. 4