А.В. Перунова, А.В. Глазкова

Тюменский государственный университет, г.Тюмень

УДК 004.932

РАЗРАБОТКА СЕРВИСА ДЛЯ ПРОВЕРКИ ПОЛНОТЫ ИНФОРМАЦИИ НА ЭТИКЕТКАХ ПИЩЕВЫХ ПРОДУКТОВ

Аннотация. В данной статье представлено описание работы над приложением, которое позволяет оценить полноту данных на фотографии этикетки, с помощью распознавания текста с изображения и шаблонов для извлечения данных.

Ключевые слова: Tesseract, Yargy, Quasar framework, распознавание текста, пищевая продукция в части ее маркировки.

Введение

В настоящее время на рынке представлено большое количество продуктов питания. С каждым годом их становится все больше. Для этикеток установлены нормы [1], которых должны придерживаться производители товаров. Так как фирмы производят большое количество товаров, появляется необходимость автоматически проверять этикетки на соответствие установленным нормам. Автоматизация данного процесса позволит сократить время фирмам-производителям, проверяющим лицам, а также рядовым покупателям, которые хотели бы проверить производителя товара на добросовестность.

Работы в данном направлении представлены:

- В статье «Recognition of Nutrition Facts Labels from Mobile Images» [2] описывается система обработки для определения

позиций на этикетках продуктов питания на фотографиях с мобильного телефона.

- В статье «Image Processing for the Extraction of Nutritional Information from Food Labels» [3] предоставляется информация о программном интерфейсе для анализа изображений, который считывает и отображает информацию, представленную на этикетках продуктов питания.

Оба решения для распознавания текста с фотографии используют Tesseract.

Целью работы является разработка приложения для проверки полноты информации на этикетках пищевых продуктов.

Задачу проверки текста этикетки на полноту можно разделить на следующие части:

- распознать текст с изображения;
- проверить текст на полноту, т. е. на наличие следующих пунктов:
    - состав;
    - количество (масса или объем);
    - дата изготовления;
    - срок годности;
    - пищевая ценность:
        - энергетическая ценность (ккал/Дж)

- белки;
- жиры;
- углеводы.

Архитектура системы

Продукт представлен в виде системы. Как показано на рис. 1, система получает фотографию от пользователя и выводит заключение о полноте данных на этикетке.

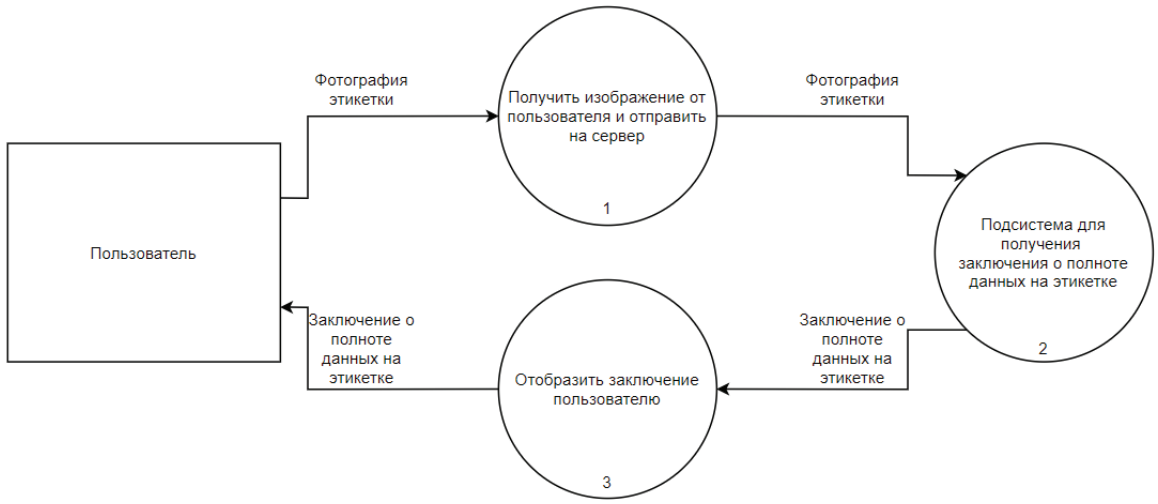

Рис. 1. Описание архитектуры на глобальном уровне

Система состоит из следующих компонентов:

- Пользовательский интерфейс – сайт и мобильное приложение, с помощью которого пользователь отправляет фотографию и получает заключение.
- Подсистема, получения заключения о полноте данных на этикетке – подсистема для распознавания текста с фотографии и его анализа (рис. 2)

Рис 2. Детализация подсистемы, получения заключения о полноте данных на этикетке.

Распознавание текста с изображения

Перед непосредственным распознаванием текста с изображения требуется его предобработка. Существует большое число методов и их комбинации для предобработки изображения. В этой работе использовался метод бинаризации.

Бинаризация

Бинаризация – это преобразование изображения в черно-белое. В случае бинаризации фотографии с текстом: белое – фон, черное – текст.

Для преобразования изображения необходимо определить порог, по которому будет определяться каким является данный пиксель – белым или черным.

Самым эффективным, как по быстродействию, так и по качеству, считается метод Оцу [4].

Метод Оцу [5] избегает необходимости выбирать порог и определяет его автоматически. Алгоритм Оцу пытается найти пороговое значение (t), которое минимизирует дисперсию внутри класса, определяемую соотношением:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

где веса q_j – это вероятности двух классов, разделенных порогом t:

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

, σ_j^2 – дисперсия этих классов:

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=1}^{t} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Пример применения метода Оцу представлен на рис. 3.

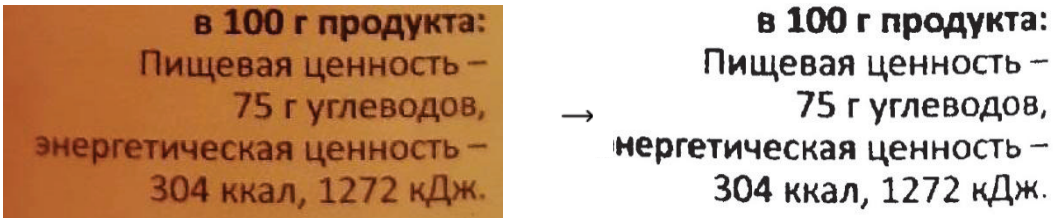

Рис. 3. Пример бинаризации изображения.

Tesseract

Для распознавания текста с изображения используется библиотека Tesseract OCR [6]. Tesseract – механизм распознавания текста с открытым исходным кодом. Tesseract предоставляет механизм распознавания текста на основе нейронной сети (LSTM), который ориентирован на распознавание линий.

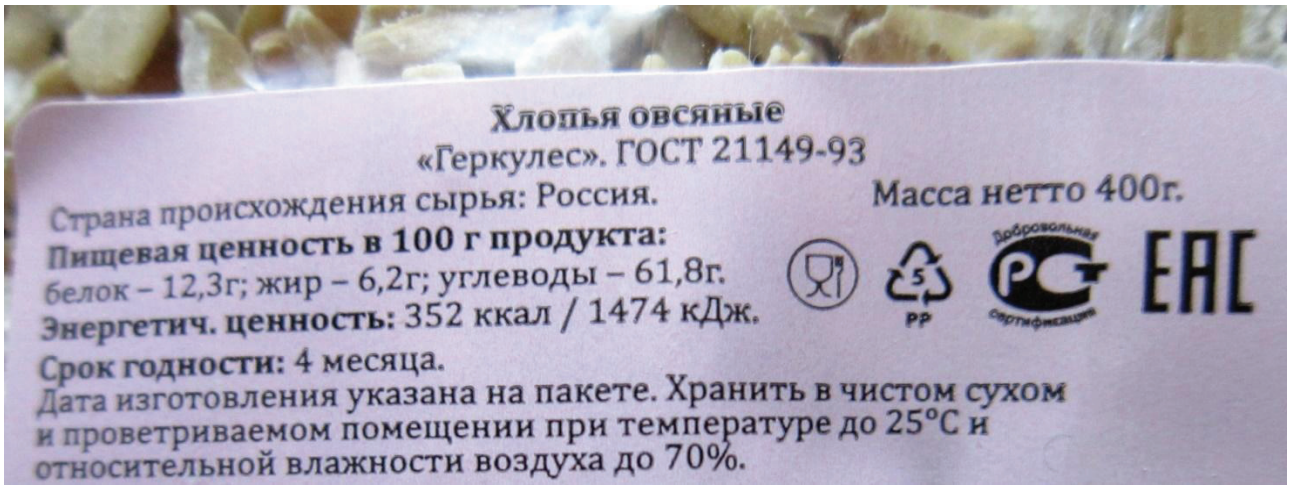

Рис. 4. Изображение с текстом.

Пример распознавания текста с рис. 4:

«Хлопья овсяные Геркулес. ГОСТ 21149 93 страна происхождения сырья: Россия. Масса нетто 400г. Пищевая ценность в 100 г продукта: белок 12,3г жир 6,2г углеводы 61,8г. Энергетич. ценность: 352 ккал 1474 кДж. Срок годности: 4 месяца. Дата изготовления указана на пакете. Хранить в чистом сухом и проветриваемом помещении при температуре до 25 С и относительной влажности воздуха до 70.»

Проверка текста на полноту

Для проверки текста, распознанного с фотографии, на соответствие нормам используется Yargy. Yargy [7] – библиотека для извлечения структурированной информации из текстов на русском языке. Правила описываются контекстно-свободными грамматиками и словарями ключевых слов. Для работы с морфологией используется Pymorphy2 [8].

Правила извлечения необходимых пунктов представлены на рис. 5.

Рис. 5. Схематическое представление шаблонов для извлечения данных.

Пример извлеченных данных для рис. 4:

```
{
    "Состав": null,
    "Количество": {
        word: "масса нетто",
        value: { value: "400", unit: "г" }
    },
    "Дата изготовления": null
    "Энергетическая ценность": {
        word: "энергетич. ценность",
        c_value: { value: "352", unit: "ккал" },
```

```
        j_value: { value: "1474", unit: "кдж" }
    },
    "Белки": {
        word: "белок",
        value: { value: "12,3", unit: "г" }
    }
    "Жиры": {
        word: "жир",
        value: { value: "6,2", unit: "г" }
    },
    Углеводы: {
        word: "углевод"
        value: { value: "61,8", unit: "г" },
    }
}
```

Интерфейс

Для работы с пользователем написано веб-приложение, интерфейс которого представлен на рис. 6.

Рис. 6. Интерфейс приложения.

Для разработки приложения используется Quasar [9]. Quasar – это MIT-платформа с открытым исходным кодом, основанная на Vue.js, которая позволяет создавать адаптивные веб-сайты/приложения.

Заключение

Разработанное приложение позволяет пользователю оценить полноту данных на этикетке по выбранным пунктам.

На данном этапе выявлено несколько проблем, связанных с распознаванием текста с фотографии этикетки:

- Искажение поверхности из-за формы упаковки продукта. Для улучшения предсказания необходимо разработать алгоритм, позволяющий устранить искаженное.

- Исправление выходного текста, т.е. удаление «мусора», когда Tesseract распознал шум, как букву и исправление опечаток.

В последующей работе над приложением планируется добавить извлечение информации об условиях хранения, местоположении производства и проверки его на существование, а также, единого знака обращения продукции на рынке государств-членов Евразийского экономического союза (EAC).

СПИСОК ЛИТЕРАТУРЫ

1. ТР ТС 022/2011 Технический регламент Таможенного союза "Пищевая продукция в части ее маркировки" (с изменениями на 14 сентября 2018 года)
2. O. Grubert, L. Gao. Recognition of Nutrition Facts Labels from Mobile Images // Steford, 2017. URL: https://stacks.stanford.edu/file/druid:bf950qp8995/Grubert_Gao.pdf - проверено 08.04.2020
3. N. Matsunaga, R. Sullivan. Image Processing for the Extraction of Nutritional Information from Food Labels // Scholar Commons, 2015. URL: https://scholarcommons.scu.edu/cgi/viewcontent.cgi?article=1041&context=cseng_senior - проверено 08.04.2020
4. Федоров А. Бинаризация черно-белых изображений: состояние и перспективы развития. URL: http://it-claim.ru/Library/Books/ITS/wwwbook/ist4b/its4/fyodorov.htm#Table1 - проверено 12.03.2020
5. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979
6. Tesseract OCR, https://github.com/tesseract-ocr/tesseract - проверено 05.04.2020

7. Yargy, https://yargy.readthedocs.io/ru/latest/ - проверено 10.04.2020
8. Pymorphy2, https://pymorphy2.readthedocs.io/en/latest/ - проверено 12.04.2020
9. Quasar, https://quasar.dev/introduction-to-quasar - проверено 11.04.2020