УДК 004.89:664](045)

KOZAK O.S.
SINEGLAZOV V.M.

INTELLIGENT SYSTEM FOR DETERMINING THE HARMFULNESS OF FOOD COMPOSITION

У даній роботі обґрунтовано створення інтелектуальної системи визначення та аналізу складу продуктів для забезпечення користувачів автоматизованим інструментом визначення оцінки шкідливості складу продуктів, який буде інтелектуально визначати текстову інформацію складу. Заразом, було розроблено та запропоновано архітектурне рішення для побудови інтелектуальних систем, які обробляють текстові дані зображення, з можливістю оновлення інтелектуального програмного модуля та обміну даними з зовнішнім харним середовищем без постійного доступу до серверу. Розроблена архітектура включає в себе: безсерверне середовище, яке дозволяє матеріально ефективно обробляти невеликі навантаження; зовнішнє сховище даних для зберігання великої кількості неструктурованих даних; клієнт, який може тривалий час підтримувати функціональність, знаходячись у ізоляції (від решти системи, при відсутності мобільного зв'язку тощо); зовнішній репозиторій для моделей нейронних мереж, що дозволяє розповсюджувати оновлення моделі, не вимагаючи оновлення всього застосунку. Заразом, у статті було описано багатошарову структуру інтелектуальної системи персоналізованої оцінки складу продуктів харчування, а також обґрунтовано необхідність створення програмного забезпечення у вигляді бібліотеки на мові програмування Java для інтелектуального виявлення тексту на зображеннях. Створена бібліотека реалізує використання високопродуктивних алгоритмів та рішень, таких, як: алгоритм MSER для детекції тексту, алгоритм видалення нетекстових регіонів, алгоритм вирівнювання тексту на зображенні, алгоритм групування слів та рядків, згорткову нейронну мережу MobileNetV2 для розпізнавання символів, які дозволяють застосовувати її на пристроях з обмеженими обчислювальними ресурсами та містить підтримку обробки українського тексту. Результатом роботи є оцінка ефективності запропонованого програмного забезпечення.

КЛЮЧОВІ СЛОВА: автоматизована оцінка шкідливості складу продуктів; архітектурне рішення для інтелектуальної системи; інтелектуальна система; програмне забезпечення; розпізнавання тексту.

This paper substantiates the creation of an intelligent system for determining and analyzing the composition of products to provide users with an automated tool for determining the assessment of the harmfulness of the composition of products, which will intellectually determine the textual information of the composition. At the same time, an architectural solution was developed and proposed to build intelligent systems that process textual image data, with the ability to update the intelligent software module and exchange data with an external environment without constant access to the server. The developed architecture includes: a serverless environment that allows materially efficient handle small loads; external data storage for storing large amounts of unstructured data; a client that can maintain functionality for a long time, being isolated (from the rest of the system, in the absence of mobile communication, etc.); an external repository for neural network models, which allows you to distribute model updates without requiring updates to the entire application. At the same time, the article describes the multilayer structure of the intelligent system of personalized assessment of food composition, and also substantiates the need to create software in the form of a library in the Java programming language for intelligent text detection in images. The created library implements the use of high-performance algorithms and solutions, such as: MSER algorithm for text detection, algorithm for removing non-text regions, text skewing algorithm in the image, word and line grouping algorithm, MobileNetV2 convolutional neural network for character recognition, which allows to use it on devices with limited computing resources and includes support for Ukrainian text processing. The result is an assessment of the effectiveness of the proposed software.

KEYWORDS: automated assessment of product composition harmfulness; architectural solution for intelligent system; intelligent system; software; text recognition.

I. INRODUCTION

The purpose of the intelligent system for determining and analyzing the composition of products is to provide users with a tool that will intelligently determine the textual information of the product composition and automatically calculate the assessment of the harmfulness of the food composition.

The developed system implements a hybrid architecture that combines the advantages of client-server and client architecture and combats the disadvantages of each.

When using machine learning methods to process any type of information by mobile systems, the software module, which includes an information processing component is located mostly on the server part to increase performance. Despite the high speed, the use of client-server architecture in the development of mobile applications has many disadvantages, such as the problem of scalability, the problem of constant data access, privacy, lack of access to the system in the absence of a mobile connection [1]. Therefore, there is a need to transfer all basic computing operations to the client's side, thus struggling with the above problems. In other words, in the developed system of intelligent detection and assessment of the harmfulness of the composition of products, such a software module for detecting and recognizing text should be in the client part of the application. This solution raises a list of other problems: reducing the speed of text recognition in the image and its analysis due to lack of computing power; lack of reliable storage of user information about possible operations performed by him, etc.; the ability to update local database data only when updating the entire system.

II. PROPOSED SOLUTION

Our system allows users to enter a list of threats, restrictions, or allergies, i.e., essential user data that the user does not want to lose and re-enter. Therefore, there is a great need to save this information when logging out, etc. Also, another problem may be the subject area, the interaction with which relies on the use of vocabulary data that requires frequent updating. For example, in a developed system for assessing the harmfulness of the composition of products, information about the ingredients that are part of it should be regularly maintained. This problem can be solved by using additional external data storage.

But with the classic server approach, payment for calculated resources does not depend on the load. In our system, the load is low and not regular, so there is no need to pay for server time.

For the optimization of material expenses, it is necessary to use a serverless architecture for these purposes [2]. The combination of using the benefits of a client architecture for computing and a serverless architecture for data storage makes it possible to get the best of the two traditional architectures (Fig. 1).

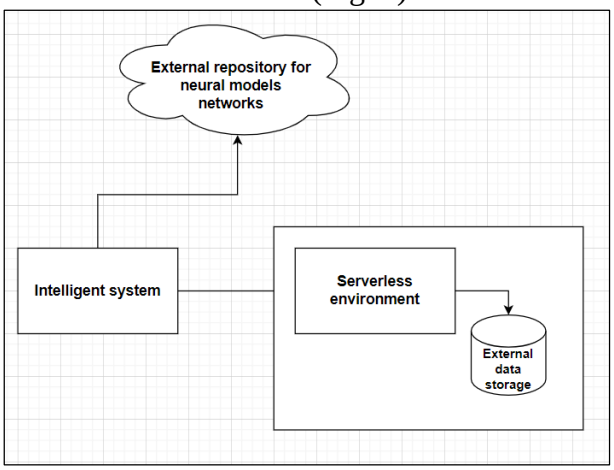

Figure 1 – Components of hybrid architecture

The developed hybrid architecture of the system has the following components:

• serverless environment that allows materially efficient handling of small loads [3].

• external data storage for storing large amounts of unstructured data.

• a client that can maintain functionality for a long time while isolated (from the rest of the system, in the absence of a mobile network, etc.).

• external repository for neural network models, which allows distributing model updates without requiring updates to the entire application.

III. THE STRUCTURE OF THE DEVELOPED INTELLIGENT SYSTEM

In the developed software the client is presented as an intelligent system for determination and the analysis of the food composition. The client part, in its turn, has a multi-layered architecture and consists of the five levels, which interact closely with each other. The first level – presentation level is a user interface; the business logic level is the implementation of

the subject area; the infrastructure level is software for intelligent text recognition of the composition; platform level represents the interaction of platform components, such as notification, camera, geolocation with system components; data level is a database and interaction with it.

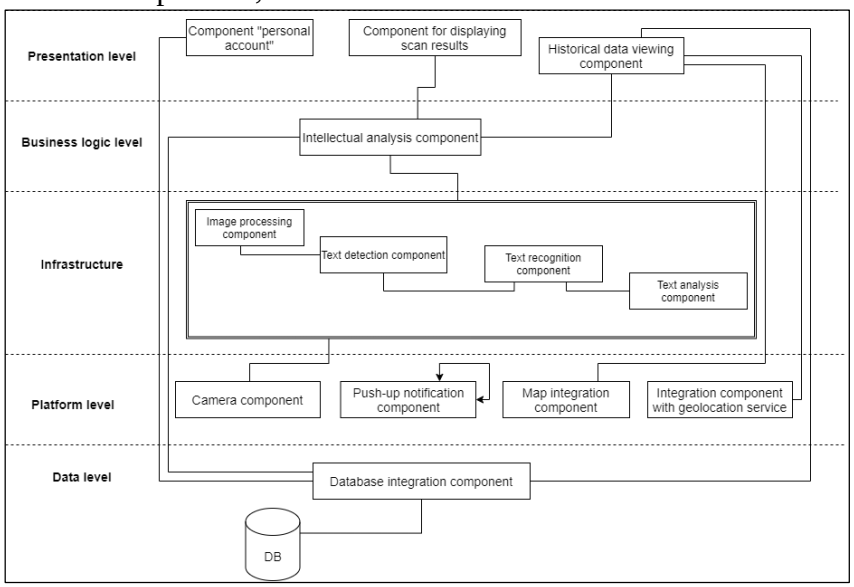

Figure 2 – Multilayer structure of the client part

The presentation level includes three components: the ‘personal account’ component, the component for displaying scan results, and the historical data view component. The personal account component allows the user to enter personal data such as restrictions, allergies, health problems, preferences, which will be taken into account when assessing the harmfulness of a food product. So, if there is an ingredient that is part of the user's restrictions, it will have the highest level of harm, and the system will inform the user about the presence of this ingredient in the food product. The component for displaying scan results is a presentation component that displays a product hazard assessment and a list of products with their characteristics. The historical data viewer component stores and presents information about previously scanned foods for quick access and resource savings for image and composition reprocessing. The ‘personal account’ component interacts with the database integration component to store information entered by the user. The component for displaying the results of scanning interacts with the component for analyzing the composition of products and displays the result of assessing the harmfulness of the product.

The business logic level includes a component of product composition analysis and represents the algorithmic support of the system – the decision-making algorithm. Based on the text, recognized from the image, the algorithm analyzes the ingredients and calculates the assessment of the harmfulness of the food product.

The platform level includes the interaction of levels with platform mechanisms for ease of use of the application. The image capture component interacts with the camera and uses the camera to capture images. The geolocation integration component makes it easy to determine the user's current location to store relevant data in the historical data view component. The data level includes a database integration component and gives other components access to data access mechanisms.

The infrastructure layer is software for processing and recognizing text in images with the composition of products. The text detection component finds the letters in the image and extracts the text of the product composition in the form of an array of images. The text recognition component receives a set of single-letter images from the previously described component and recognizes letters using a neural network classifier. The text analysis component deals with the division of the text into lines, composing letters into words. If it is necessary, the word is filled with missing letters based on the dictionary of the database.

The infrastructure layer of our system, which

is a component of text detection, is a developed reusable library for detecting and recognizing textual information in images. The implementation of the developed library allows you to reassign the implementation of the classifier (neural network) of symbols on the image matrix, which allows the developer to dynamically update the models, or use the built-in.

The criterion for assessing the accuracy of text recognition of the proposed software solution was the following ratio (represented by percentage): the sum of correctly recognized ingredients in relation to all available ingredients together with the sum of all misrecognized ingredients (false positive recognition) - "relative accuracy" and amount misrecognized - "absolute error". The result of the accuracy assessment is given in the table 1.

Table 1. The results of the accuracy assessment

Average relative accuracy	98.33 %
Minimum relative accuracy	92.14 %
Average absolute error	0.87
Maximum absolute error	1.30

CONCLUSION

The presented architectural solution helps to overcome the problem of scalability by abandoning the traditional client-server architecture and transferring the basic infrastructure logic to the client's side. In this case, for the possibility of storing valuable user data, a non-permanent serverless environment is used. The structure of an intelligent system presented as interconnected components helps gain constant access to the use of the application, and obtain an assessment of the composition of the food product.

During the implementation of the application, an independent reusable component, which can be supplied as a separate library for quick text recognition, and the component for analysis of product composition for the assessment of product harmfulness were developed.

This reusable program module in the form of a library differs from existing in high-speed text recognition due to the use of highly efficient mathematical algorithms, such as MSER text detection algorithm, non-text region deletion algorithm, word composition algorithm, and solutions - convolutional neural networks for character recognition.

References:

1. Sawan Gupta, Sahil Narang, Shubham Arora, Prof. Shruti Jindal. “A Brief Discussion About Client-Server System” in “International Journal of Advanced Research in Computer Engineering & Technology” (vol. 6, issue 3, march 2017). [Online]. Available: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-6-ISSUE-3-338-341.pdf

2. R. Arokia Paul Rajan, “Serverless Architecture - A Revolution in Cloud Computing” in Tenth International Conference on Advanced Computing (ICoAC), Dec. 2018, doi: 10.1109/ICoAC44903.2018.8939081

3. Garrett McGrath; Paul R. Brenner, “Serverless Computing: Design, Implementation, and Performance” in IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), June 2017, doi: 10.1109/ICDCSW.2017.36

УДК 683.519

Шелудько Д.М.,
Ковтунець О.В..

АНАЛІЗ ПЕРЕВАГ ТА НЕДОЛІКІВ ФРЕЙМВОРКІВ ORLEANS ТА AKKA.NET

В даній статті розглянуті основні переваги та недоліки двох фрейморків для платформи .NET. Також розглянуті сценарії використання та основні проблеми, що вирішують фреймворки. Детально описана математична модель та ідея, що лежить в основі інструментів.
КЛЮЧОВІ СЛОВА: .NET, ORLEANS, AKKA, ACTOR MODEL, DISTRIBUTED SYSTEMS