BELARUSIAN STATE UNIVERSITY OF INFORMATICS AND RADIOELECTRONICS

DEPARTMENT OF CROSS-CULTURAL PROSESSIONAL COMMUNICATION

COMPUTER VISION ALGORITHMS FOR FOOD RECOGNITION

Anzhalika V. Dziarkach
Master Degree Student
Speciality: *Computer engineering*

Scientific Supervisor
Associate Professor, PhD in Philology
Department of Cross-cultural
Professional Communication
Nina N. Pristupa

Minsk 2024

CONTENTS

ABSTRACT

The essay provides an overview of modern methods and algorithms of computer vision used in the field of food recognition. Given the increasing need for automation in sales processes and inventory management, such technologies are becoming increasingly sought after in retail and the food industry.

The main methods of image processing, machine learning, and deep learning are considered, along with their application for the classification and recognition of products based on visual information. The advantages and limitations of various approaches are discussed, as well as prospects and possible directions for further research in this area. Ultimately, the essay presents a review of modern achievements and trends in the application of computer vision algorithms for solving food product recognition tasks.

РЭЗЮМЭ

У рэфераце праводзіцца агляд сучасных метадаў і алгарытмаў камп'ютэрнага гледжання, якія ўжываюцца ў галіне распазнання прадуктаў харчавання. У кантэксце расце патрэбнасці ў аўтаматызацыі працэсаў продажаў і ўліку тавараў, такія тэхналогіі становяцца ўсё больш запатрабаванымі ў сферы рознічнага гандлю і харчовай прамысловасці.

Разглядаюцца асноўныя метады апрацоўкі малюнкаў, машыннага навучання і глыбокага навучання, а таксама іх прымяненне для класіфікацыі і распазнання прадуктаў на аснове візуальнай інфармацыі. Абмяркоўваюцца перавагі і абмежаванні розных падыходаў, а таксама перспектывы і магчымыя напрамкі далейшых даследаванняў у гэтай галіне. У канчатковым выніку, рэферат ўяўляе сабой агляд сучасных дасягненняў і тэндэнцый у галіне прымянення алгарытмаў камп'ютэрнага гледжання для вырашэння задач распазнання прадуктаў харчавання.

РЕЗЮМЕ

В реферате проводится обзор современных методов и алгоритмов компьютерного зрения, применяемых в области распознавания продуктов питания. В контексте растущей потребности в автоматизации процессов продаж и учета товаров, такие технологии становятся все более востребованными в сфере розничной торговли и пищевой промышленности.

Рассматриваются основные методы обработки изображений, машинного обучения и глубокого обучения, а также их применение для классификации и распознавания продуктов на основе визуальной информации. Обсуждаются преимущества и ограничения различных подходов, а также перспективы и возможные направления дальнейших исследований в данной области. В конечном итоге, реферат представляет собой обзор современных достижений и тенденций в области применения алгоритмов компьютерного зрения для решения задач распознавания продуктов питания.

INTRODUCTION

In the modern digital environment characterized by the continuous increase in information flows, the questions of effective data processing and analysis acquire paramount importance. With the advancement of image processing technologies and artificial intelligence, automated image analysis systems provide new perspectives for more efficient problem-solving. The ability to automatically recognize and classify food products becomes strategically important for business processes, improving logistical processes, optimizing inventory management, and enhancing the user experience.

Computer vision is an artificial intelligence-based technology for analyzing images and videos [1]. It encompasses a set of techniques through which a computer "sees" what is shown in a video or picture. The technology analyzes images or video frames, extracting the necessary information that a person could extract when viewing them. For example, computer vision can identify objects in a video, recognize their location, and when they appear.

Such technologies are developed using machine learning methods and vast datasets. These datasets are used to train the technology to identify relevant features and their combinations to recognize objects. So, the main thing that computer vision technology does is recognize parts of images or videos and extract the necessary information. This technology is crucial for classifying food products from user photographs, and there are numerous algorithms based on which it can be built. This paper will consider the main ones.

PART 1. COMPUTER VISION

Computer vision, and sometimes machine vision, is a scientific field that deals with research in automatic fixation and various types of image processing (detection, tracking, identification) using a computer. At the physical level, computer vision systems consist of image capture devices (a camera or multiple cameras) and a general-purpose computer used for image processing [2]. In this case, special software tools are used. Many software tools can work "in the cloud" (remotely), which allows you to scale computer vision systems and centralize their management. There are a number of common software platforms that allow effective implementation of computer vision systems, these include:

– OpenCV is the most popular distributed package for free and open source [3]. It is a library of mathematical algorithms for image analysis implemented in C++, but has an API for many popular programming languages such as Python, Java, Matlab and others.

– PCL is also an open platform for processing two-dimensional and three-dimensional images [4]. It contains many implemented algorithms of different directions: assessment of characteristics, surface reconstruction, scene restoration, search for matches in images and much more.

– ROS is a specialized robotics management platform that includes a rich set of algorithms used in this field [5].

– MATLAB is a high-level programming language of general mathematical purpose, popular in the academic environment. It contains a large number of tools that simplify research: from data analysis and visualization to the creation and implementation of models [6].

– CUDA is a proprietary software development by NVIDIA, specially sharpened to work with graphics processors manufactured by this company. Implements an extremely efficient architecture for parallel computing, which makes it practically indispensable in any high-load image processing system [7].

On the technical side, computer vision is a family of image or video analysis algorithms. Algorithms and methods of computer vision can be divided into two groups [8]:

1. “Classical” computer vision. It is used when it is necessary to obtain some quantitative information about an image (related to color, shape, number of objects, etc.). It works most reliably in tasks that can be formalized and divided into subtasks. Most of the methods in this group first extract useful characteristics from an image, and then work with them to solve the problem. Such algorithms are well suited for simpler and more deterministic tasks in terms of external factors: lighting and distance to the subject, a slight variation in the shape of the object. To develop such algorithms, a much smaller image database is required.

2. Machine learning systems, including deep machine learning (neural networks). These are complex systems that are much more demanding on computing resources and data volumes, but have been developing rapidly in recent years. Systems of this kind partially mimic human image perception abilities, and therefore allow you to extract much more complex information from an image. It makes sense to use such systems in cases, where splitting a task and compiling a step-by-step algorithm is extremely laborious or even impossible. These algorithms are more resistant to false positives when changing the illumination, size or angle of an object, and allow you to build more advanced and fault-tolerant computer vision systems. Neural algorithms are demanding on computational resources and the completeness of the image database for training, and also require appropriate experience in data preparation and model training.

Overall, by integrating food detection with recipe suggestion capabilities, the system enhances users' cooking experience, promotes efficient use of ingredients, and facilitates healthier and more sustainable eating habits. When a user uploads a photo of the contents of their refrigerator to an application or service utilizing computer vision for food detection, the process extends beyond simple identification. After the image is scanned, the system analyzes the various food items present and suggests potential recipes based on those ingredients. The result of food recognition in the photo is shown in Figure 2.

Figure 1 – Food detection in the photo using computer vision

Users can derive several benefits from this feature [9], starting with efficient meal planning, where they can receive personalized recipe suggestions based on the ingredients available in their refrigerator, which streamlining the meal planning process and making it easier to decide what to cook with existing ingredients. Additionally, the system helps reduce food waste by suggesting recipes that utilize ingredients already on hand, encouraging users to use up perishable items before they spoil and leading to more sustainable consumption habits. Users can also explore new culinary ideas and discover creative ways to prepare meals using the ingredients they have at home, inspiring experimentation in the kitchen and broadening their culinary repertoire. Furthermore, the convenience and time-saving aspect of the system allows users to rely on it for quick and individualized suggestions, eliminating the need for manual inventory checks and recipe searches. This saves time and effort in meal planning and preparation. Lastly, the system can prioritize recipes that align with users' dietary preferences and nutritional goals, promoting balanced and nutritious meals and encouraging healthier eating habits.

PART 2. CONTOUR ANALYSIS

Contour analysis is a method of describing, storing, recognizing, comparing and searching graphic images (objects) by their contours. A contour is a curve that describes the boundary of an object in an image. Using this approach assumes that the contour contains enough information about the shape of the object, while the internal points are not taken into account [10]. Considering only the contours of objects allows you to move away from the image space to the contour space, which significantly reduces the complexity of algorithms and calculations. The main advantage of contour analysis is the invariance with respect to the rotation, scale and displacement of the contour in the image under test. It’s great for searching for an object of some given shape.

To understand an object in an image, first, it's necessary to find its shape, determined by its contour. The boundary or contour marks the outline of an object in an image. So, detecting contours plays a vital role in applications for identifying and segmenting objects in an image. A contour consists of the pixels in an object’s boundary. These pixels are usually of the same color, differentiating them from the rest. To identify the contour, the initial step involves binarization of the image, which converts it from the standard red-green-blue (RGB) format to binary. A binary image contains only zeros and ones, where one indicates the white color, and zero stands for black. Notably, the most commonly used binarization technique is Otsu’s threshold method. Converting to binary form helps in identifying contour points. The result of finding contours is a bounding line drawn on the object’s outline in an image.

The contours can be traced using chain codes that indicate the directions of tracing along the border [11]. Tracing starts from the selected initial point and proceeds clockwise. There are two types of chain codes, 4-directional and 8-directional. They differ in the number of directions in which the contour can be drawn, and indicate a unique number for each direction. Let’s see how an object can be represented with chain codes in Figure 2.

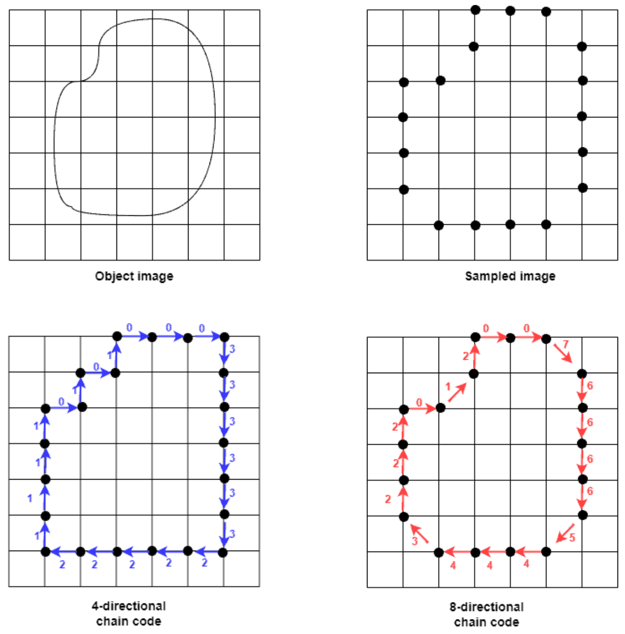

Figure 2 – Object represented by chain codes

The image after sampling shows the boundary pixels, which are the contour points. Assuming the starting position is top-left, the chain code moves clockwise. In our example, the 4-directional and 8-directional chain codes are 00033333322222211110101 and 0076666544432222012. However, the described assumptions about the contour impose significant limitations on the scope of this method. First of all, they are caused by contour selection problems in the image:

– with the same brightness as the background, the object may not have a clear border, or it may be noisy with interference, which makes it impossible to highlight the contour;

– overlapping of objects or their grouping leads to the fact that the contour is highlighted incorrectly and does not correspond to the border of the object.

Thus, contour analysis has a rather weak resistance to interference, and any violation of the integrity of the contour or poor visibility of the object leads either to the impossibility of detection or to false positives. However, the simplicity and speed of contour analysis make it possible to apply this approach quite successfully, provided that there is a clearly defined object on a contrasting background and there is no interference.

PART 3. TEMPLATE MATCHING

The goal of the template matching algorithm is to find the area on the test image that best matches the template. Thus, the input parameters of the method include the image on which the search for the template will be conducted and the image of the object intended to be found on the test image; the size of the template should be smaller than the size of the image being examined.

To perform template matching on an image, digital image processing methods are applied [12]. Initially, both the template and the image are converted into numerical arrays, where each element represents the brightness or color of a pixel. For each position of the template on the image, the similarity between the template and the corresponding area of the image is computed using various techniques, including Sum of Squared Differences (SSD) and Normalized Cross Correlation (NCC).

With SSD, the squared differences in intensity between each pixel of the image and the corresponding pixel of the template are calculated and then summed for all pixels in the template to derive an overall similarity measure. SSD relies on intensity differences between pixels in the two images, facilitating the determination of the matching point by identifying the minimum value in the resulting matrix of differences.

On the other hand, NCC begins by computing the average brightness (or color) of both the template and the corresponding area of the image. Then a normalized correlation is calculated between the intensities (or colors) of the template pixels and the corresponding image area. NCC is often preferred for its robustness, as it identifies the matching point between the template and the image by pinpointing the maximum value in the resulting correlation matrix.

Following the calculation of similarity for each template position, the area of the image with the highest similarity coefficient is identified, representing the result of matching the template with the image. Example of applying template matching is shown in Figure 3.

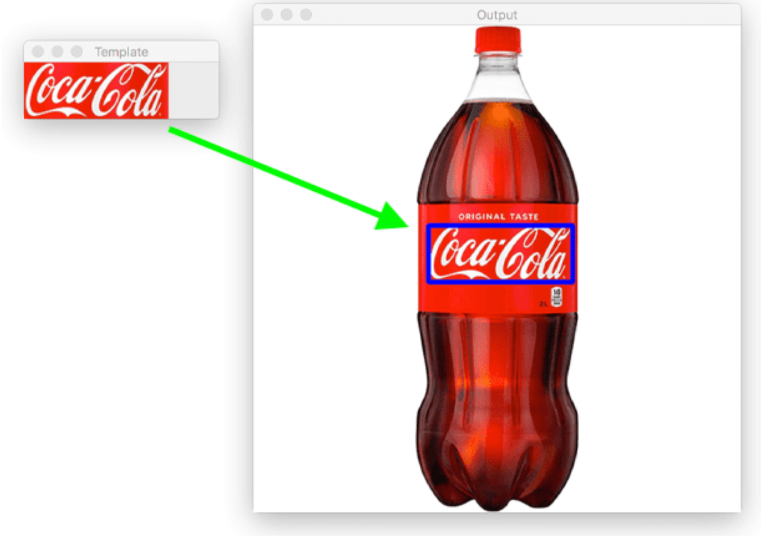

Figure 3 – Example of applying template matching

Template matching is a good choice when it is necessary to quickly check for the presence of a certain object in an image. This approach is significantly easier to implement than using trained algorithms. Various optimizations, such as employing windowing functions to reduce computational complexity during correlation calculation, may be implemented based on the task's complexity and performance requirements. However, it is important to understand that template matching does not allow us to confidently determine whether the original object has been found, as it’s a probabilistic characteristic that depends on the scale, viewing angles, image rotations, and the presence of physical interference. False positives of the algorithm are also possible when the desired object is not actually present, but there are some common details between the template and the area on the test image. Of course, such a situation can be avoided by checking the value of the matching coefficient (to ensure it is not below a certain threshold), but this may not always work properly due to the reasons described above.

PART 4. FEATURE DETECTION

The concept of feature detection in computer vision refers to methods aimed at computing abstractions of images and extracting their key characteristics. These features are then utilized to compare two images in order to identify common components [13]. There is no strict definition of what constitutes a key feature of an image. Typically, a key point is considered to be a point in the image that is likely to be found in another image of the same object. These points can be isolated points, curves, or certain connected regions. Examples of such features include object edges and corners. To find key points in images and subsequently compare them, three components are used:

1. Feature detector – searches for key points in the image.
2. Descriptor extractor – describes the found key points by evaluating their positions relative to the surrounding areas.
3. Matcher – establishes correspondences between two sets of image points.

Initially, the detector searches for key points in the image. The obtained points are then described using descriptors. This information is stored in a separate file or database to avoid repeating this process. When processing a video stream to search for a specified pattern, the described algorithm is executed for each frame. A matcher is applied to establish correspondences between key points and descriptors, as illustrated in Figure 4.

It should be noted that key points are invariant to changes in image scale and rotation, and partially invariant to changes in illumination and the 3D perspective of the camera. Such points are applicable both in spatial and frequency domains, reducing the likelihood of failures during transformations, noise, and interference. A large number of characteristics can be extracted from images using efficient algorithms. Furthermore, these characteristics have distinct features, which allows for the high probability computation of a particular characteristic from a large database of characteristics for further object or scene recognition.

Figure 4 – Example of key points detection

During the initial stages of computation, analysis of image scale and position is performed. The method's efficiency is achieved by using the difference of Gaussians function to determine potentially interesting points invariant to scale and orientation. During the second stage, key points are placed at each potential location, describing a detailed model to find coordinates and scale. Key points are chosen based on their stability degree. In the third stage, one or more coordinates are assigned to each key point based on the local direction of image gradient. All subsequent stages are performed on the transformed image data according to the specified coordinates, scale, and location of each characteristic, thereby ensuring invariance to these transformations. During the fourth stage, local image gradients are measured at the selected scale in the area of each key point. They are transformed into a representation that allows for distortion of local shape and changes in illumination at important levels. This approach is called Scale-Invariant Feature Transform (SIFT) because it converts image data into scale-invariant coordinates according to local characteristics.

An important aspect of this approach is that it generates a large number of features that densely cover the image across all scales and coordinates. A typical 500×500 pixel image includes around 2000 stable characteristics, although this number depends on both the image content and the choice of various parameters [14]. The number of characteristics is particularly important for object recognition, where the ability to detect small objects from cluttered backgrounds requires at least three features to be correctly matched for reliable identification. For comparing and recognizing images, SIFT features are first extracted from a set of corresponding images and stored in a database. Each feature of a new image is then compared to the features of images in the previous database, finding common comparative features based on the Euclidean distance of the feature vector.

Key point descriptors have strong distinctive features, allowing any feature to find a match in a large database with a high probability. However, in images with noise, most background features will not find a match in the database, increasing the likelihood of false matches. True matches can be sorted from the entire set of matches by determining a subset of key points that are consistent with the object and its position, scale, and orientation. Each such group, consisting of three or more features and consistent with the object and its position, undergoes detailed verification. Any other features corresponding to this position are retained, while non-corresponding ones are discarded. A detailed calculation is performed to assess the probability that this set of features determines the presence of the object, the accuracy of the comparison, and the number of potential false matches. It is assumed that comparative features of objects that pass these checks with high accuracy are correct.

PART 5. MACHINE LEARNING AND DEEP LEARNING

Machine Learning (ML) is a field of Artificial Intelligence (AI) focused on creating systems that learn and evolve based on the data they receive. Artificial Intelligence is a broad term encompassing computer systems that mimic human intelligence. Machine Learning involves various algorithms that learn from data to discover patterns, make optimal decisions, and generate predictions. Depending on the desired outcome and input type, four main learning styles are identified:

1. Supervised Machine Learning. This approach involves training a model based on "input" and "output" data pairs, where the output data has corresponding labels. Algorithms used in this type of learning analyze the data to identify correlational dependencies and create a model capable of predicting responses based on new input data.

2. Unsupervised Machine Learning. In this case, the model is trained on unlabeled data without an explicit key to the answer. Algorithms analyze the data structure, identify patterns, and group them based on input properties without prior labels.

3. Semi-supervised Machine Learning. In this learning style, the model uses both labeled and unlabeled data. Labeled data is used to supplement unlabeled datasets, enhancing the accuracy of training. Algorithms analyze labeled data to identify properties that can be applied to unlabeled data.

4. Reinforcement Learning. In this approach, the algorithm learns by interacting with the environment. It makes decisions and receives feedback in the form of rewards or penalties, aiding in improving strategies and making more optimal decisions in the future. In the context of food product classification from photographs, this means that if the algorithm's classification matches what was actually in the photo, the user can confirm the accuracy of the results or provide additional information about the products. In this case, the algorithm receives a positive reward for correct classification. However, if the results are incorrect, the user can correct them, and the

algorithm receives feedback in the form of a negative assessment, helping it improve its classification in the future.

Basic machine learning algorithms find wide application in solving a variety of applied tasks [15]. Knowledge of them is essential for any specialist involved in data analytics and artificial intelligence application development. These include:

– linear regression;

– logistic regression;

– regression trees;

– random forests;

– support vector machine (SVM);

– k-means clustering;

– neural networks;

– gradient boosting;

– association rules.

Data preprocessing is an important stage in machine learning. It involves a series of steps that need to be performed to make the data suitable for model training. The key data processing includes the following steps:

1. Data collection. This step involves gathering and obtaining the data needed for the task at hand. Data can be obtained from various sources, including databases, files, and other methods.

2. Data cleaning. In this step, errors, missing values, outliers, and other inaccuracies in the data are removed or corrected. This may involve removing duplicates, filling in missing values, correcting erroneous records, etc.

3. Data transformation. Some machine learning models require a specific data format or certain types of features. In this step, the data is transformed or encoded to meet the model's requirements. For example, categorical variables may be converted into numerical values, textual data may be vectorized, etc.

4. Data scaling. Some machine learning models require data normalization or standardization. In this step, the data is scaled so that it has a similar range of values

and distribution, which helps the model learn better and make more accurate predictions.

5. Splitting the data into training and testing sets. To evaluate the model's performance, it's necessary to split the data into a training set, on which the model will be trained, and a testing set, on which its effectiveness will be assessed. Typically, data is split in a proportion such as 70% for training and 30% for testing.

Data preprocessing requires attention and thoroughness because the quality and cleanliness of the data affect the accuracy and reliability of the machine learning model. Deep learning is a type of machine learning that utilizes deep neural networks, which self-learn on large datasets. Artificial intelligence with deep learning discovers the algorithm for solving the initial task by itself, learns from its mistakes, and after each training iteration provides a more accurate result. Functionally, neural networks are divided into layers – structures of neurons with a common task, as shown in Figure 5.

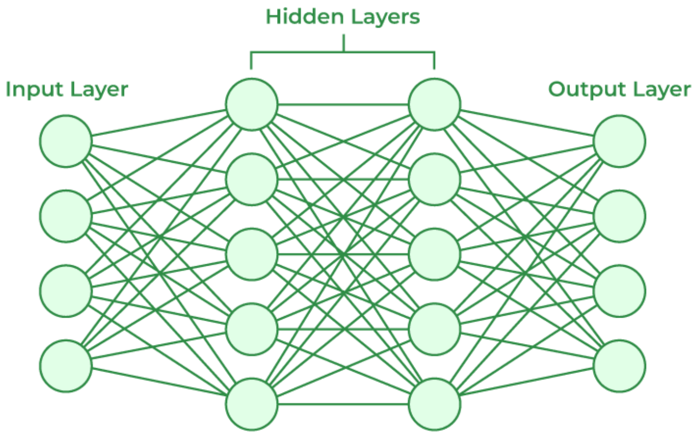

Figure 5 – Neural network structure

The input layer receives a dataset. In the simplest case, each neuron is responsible for one parameter. For example, in neural networks for food product detection in images, these would be arrays of pixels from these images. Each neuron in this layer is responsible for one pixel of the image or for a characteristic of this pixel (such as brightness or color). The input layer passes information about these parameters to the hidden layers. Hidden layers perform calculations based on the input parameters.

Connections between neurons have their weight – the importance of the parameter among all the data. For example, in a neural network for detecting products in images, characteristics associated with the unique features of each product, such as its shape, color, texture, or the presence of certain elements of packaging, would have a high weight. These characteristics can be crucial in determining a specific product in an image and can have a significant impact on the network's output. The output layer outputs the result of the calculations, such as the names of the food products found in the image.

In deep learning, more than one hidden layer is used, allowing the neural network to find more relationships in the input data. Such models are called deep neural networks. For example, convolutional neural networks are used in computer vision. In the architecture of such neural networks, multiple layers are used, adjusting their number for each task. The further the information from the input image progresses through the neural network, the more abstract details the network finds. For example, on the first layers, the model finds basic shapes, such as squares and circles, from which any image consists. On the next levels, it can analyze textures and colors, determining, for example, labels or packaging. At higher levels, the network can identify characteristics of specific products, such as logos, brands, or unique packaging shapes.

CONCLUSION

Research on computer vision algorithms for food product recognition reveals important prospects for automating and improving processes related to image processing in the food industry, gastronomy, and healthcare. The application of contour analysis, template matching, feature detection, as well as machine and deep learning methods, provides unique opportunities for developing modern food product recognition systems capable of automatically analyzing and classifying vast amounts of data with high accuracy and speed. These technologies have the potential to revolutionize inventory processes, quality control, and inventory management in retail and food service industries. Moreover, they can be a key element in creating innovative approaches to personalized nutrition and diet monitoring in medical applications.

Each computer vision algorithm has its advantages and limitations. Contour Analysis offers precise boundary delineation but struggles with complex shapes. Template Matching is efficient for recognizing predefined patterns but can be hindered by scale and rotation variations. Feature Detection provides adaptability and versatility in identifying visual attributes but depends on feature quality. Machine Learning and Deep Learning excel in learning complex patterns, offering high accuracy and adaptability, yet demanding significant computational resources and extensive training data. Thus, the development of computer vision and its algorithms for food product recognition represents tremendous potential for transforming modern industries, providing more efficient and accurate methods of data analysis, thereby enhancing service quality and meeting the growing market demand for innovative solutions.

REFERENCES

[1] Computer Vision: What Is It And How Does It Work? [Digital source]. – Access mode: https://eastgate-software.com/computer-vision-what-is-it-and-how-does-it-work/ – Access date: 03.02.2024.

[2] 5 components of computer vision hardware you need to know [Digital source]. – Access mode: https://www.aiacceleratorinstitute.com/5-components-of-computer-vision-hardware-you-need-to-know/ – Access date: 03.02.2024.

[3] Dawson-Howe, K.M. A Practical Introduction to Computer Vision with OpenCV / K.M Dawson-Howe // John Wiley & Sons Ltd. – United Kingdom, 2014. P. 235.

[4] Rusu, R.B. 3D is here: Point Cloud Library (PCL) / R.B. Rusu, S. Cousins // In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA), May 9–13. – Shanghai, China, 2011. P. 4.

[5] ROS – Robot Operating System [Digital source]. – Access mode: https://www.ros.org/ – Access date: 04.02.2024.

[6] Vedaldi, A. Vlfeat: an open and portable library of computer vision algorithms / A. Vedaldi, B. Fulkerson // Proceedings of the 18th ACM International Conference on Multimedia, Association for Computing Machinery. – Firenze, Italy, 2010. P. 1469–1472.

[7] Park, S.I. Low-Cost, High-Speed Computer Vision Using NVIDIA's CUDA Architecture / S.I. Park, S.P. Ponce, J. Huang, Y. Cao, F. Quek // Center of Human Computer Interaction, Virginia Polytechnic Institute and University. – Blacksburg, USA, 2008. P. 7.

[8] Unleashing the Power of Computer Vision Algorithms [Digital source]. – Access mode: https://www.augmentedstartups.com/blog/unleashing-the-power-of-computer-vision-algorithms – Access date: 06.02.2024.

[9] Using Computer Vision to Understand Food and Cuisines [Digital source]. – Access mode: https://blog.roboflow.com/cuisine-classification/ – Access date: 06.02.2024.

[10] An overview of edge and object contour detection [Digital source]. – Access mode: https://www.sciencedirect.com/science/article/abs/pii/S092523122200248X – Access date: 09.02.2024.

[11] What Are Contours in Computer Vision? [Digital source]. – Access mode: https://www.baeldung.com/cs/computer-vision-contours – Access date: 09.02.2024.

[12] Hashemi, N. S. Template Matching Advances and Applications in Image Analysis / N.S. Hashemi, R.B. Aghdam, A.S.B. Ghiasi, P. Fatemi // ArXiv, Volume abs/1610.07231. – 2016. P. 9.

[13] Feature Detection and Matching Concepts [Digital source]. – Access mode: https://www.ni.com/docs/en-US/bundle/ni-vision-concepts-help/page/feature_detection_concepts.html – Access date: 17.02.2024.

[14] Object recognition from local scale-invariant features [Digital source]. – Access mode: https://ieeexplore.ieee.org/document/790410 – Access date: 17.02.2024.

[15] Bonaccorso, G. Machine Learning Algorithms: A reference guide to popular algorithms for data science and machine learning / G. Bonaccorso // Packt Publishing, 2nd ed. – 2017. P. 360.

GLOSSARY

№	Term	Translation
1	Artificial Intelligence (AI)	Искусственный интеллект (ИИ)
2	Automation	Автоматизация
3	Binarization	Бинаризация
4	Brightness	Яркость
5	Chain	Цепочка
6	Classification	Классификация
7	Complexity	Сложность
8	Computer vision	Компьютерное зрение
9	Computing	Вычисления
10	Concept	Концепция
11	Contour analysis	Контурный анализ
12	Convolutional neural networks (CNN)	Сверточные нейронные сети (CNN)
13	Corner	Угол
14	Correlation	Взаимосвязь
15	Curve	Кривая
16	Data analysis	Анализ данных
17	Data preparation	Подготовка данных
18	Database	База данных
19	Dataset	Набор данных
20	Deep learning	Глубокое обучение
21	Digital	Цифровой
22	Encoding	Шифрование
23	Erroneous record	Ошибочная запись
24	Feedback	Обратная связь
25	Frame	Кадр
26	General-purpose computer	Компьютер общего назначения

27	Graphics processor	Графический процессор
28	Hidden layer	Скрытый слой
29	High-level	Высокоуровневый
30	High-load system	Высоконагруженная система
31	Identification	Выявление
32	Image noise	Шум изображения
33	Implementation	Осуществление
34	Intensity	Интенсивность
35	Invariance	Неизменность
36	Capture devices	Устройство захвата
37	Key point	Ключевая точка
38	Likelihood	Вероятность
39	Machine Learning (ML)	Машинное обучение (ML)
40	Modern	Современный
41	Neural network	Нейронная сеть
42	Neuron	Нейрон
43	Normalization	Нормализация
44	Open source	Открытый исходный код
45	Distributed package	Распространяемый пакет
46	Parallel computing	Параллельные вычисления
47	Pattern	Шаблон
48	Perishable	Скоропортящийся
49	Physical level	Физический уровень
50	Pixel	Пиксель
51	Probabilistic characteristic	Вероятностная характеристика
52	Programming language	Язык программирования
53	Quantitative	Количественный
54	Recognition	Распознавание
55	Refrigerator	Холодильник

56	Reinforcement learning	Обучение с подкреплением
57	Robustness	Надежность
58	Rotation	Вращение
59	Scale	Масштаб
60	Semi-supervised	Частично контролируемый
61	Software platform	Программная платформа
62	Step-by-step algorithm	Пошаговый алгоритм
63	Stream	Поток
64	Supervised	Контролируемый
65	Template matching	Сопоставление шаблонов
66	Tracking	Отслеживание
67	Transformation	Преобразование
68	Visualization	Визуализация