

## **НЕЙРОСЕТЕВОЙ МОДУЛЬ РАСПОЗНАВАНИЯ И СОРТИРОВКИ БЫТОВЫХ ОТХОДОВ**

*Одной из актуальных задач в настоящее время является проблема сбора, сортировки и переработки бытовых отходов. Решение данной проблемы важно не только с экологической и научно-технической стороны, но также и в информационном аспекте. Многие страны обеспокоены этим вопросом и пытаются найти решение по эффективной организации сортировки и переработки отходов для их последующего вторичного использования.*

*В данной работе рассматривается подход к организации процесса сортировки бытовых отходов на предприятии, направленный на повышение эффективности классификации отходов с использованием автоматизированных технологий. Данный подход основан на использовании нейросетевого модуля распознавания архитектуры YOLOv4 и алгоритма SORT для отслеживания объектов. Результаты тестирования модуля показывают целесообразность предложенного подхода для организации автоматизированной классификации бытовых отходов.*

*Нейросетевой модуль, распознавание объектов, классификация и сортировка отходов.*

Prudnikov Artyom Alexandrovich, Maxim Yuryevich Polenov

## **NEURAL NETWORK MODULE FOR RECOGNIZING AND SORTING OF HOUSEHOLD WASTE**

*One of the urgent tasks at present is the problem of collection, sorting and recycling of household waste. Solving this problem is important not only from the environmental, scientific and technical side, but also from the information aspect. Many countries are concerned about this issue and are trying to find a solution for the effective organization of waste sorting and recycling for their subsequent recycling.*

*This paper considers an approach to organizing the process of sorting household waste at an enterprise aimed at improving the efficiency of waste classification using automated technologies. This approach is based on the use of the YOLOv4 neural network recognition module and the SORT algorithm for tracking objects. The test results of the module show the suitability of the proposed approach for the organization of automated classification of household waste.*

*Neural network module, object recognition, waste classification and sorting.*

### **Введение**

Бытовые отходы, возникающие в результате человеческой деятельности, характеризуются разнообразным морфологическим составом, включающим бумагу, пластик и металл. Эти отходы могут быть разделены и повторно использованы, однако во многих регионах страны отсутствуют специализированные предприятия для их сортировки и переработки. Решение этой проблемы будет способствовать улучшению экологической ситуации, поскольку объемы бытовых отходов постоянно увеличиваются.

С каждым годом состав отходов становится всё сложнее, включая в себя больше экологически опасных элементов. Управление отходами усложняется, а стоимость утилизации отходов растёт.

В то же время активно разрабатываются новые технологии утилизации, включая современные системы разделения и использование отходов в качестве вторичного сырья. Далее рассмотрим подход к организации процесса сортировки бытовых отходов на основе нейросетевого модуля распознавания.

### **Предлагаемый подход**

В результате проведенного сравнительного анализа методов и средств [1-3] для решения поставленной задачи разработки автоматизированной системы распознавания и сортировки бытовых отходов, было предложено

использование нейросетевого подхода с разработкой нейросетевого модуля распознавания на основе архитектуры YOLOv4 и алгоритма SORT для обнаружения и классификации объектов. Данная архитектура и алгоритм позволяют повысить производительность и эффективность системы распознавания и классификации отходов в режиме реального времени.

### Система распознавания изображений

Рассмотрим реализацию предложенного подхода. Как известно, цифровое изображение состоит из пикселей. В RGB изображении каждый пиксель имеет 3 слоя, а именно красный, синий, зеленый, но также есть особенность с оттенками серого цвета, они в свою очередь имеют двумерную матрицу, где у каждого пикселя есть свой номер в зависимости от его уровня затемнения. В этом случае самый темный пиксель будет иметь значение равное нулю, а самый яркий пиксель будет 255. Пример такого представления приведен на рис. 1.

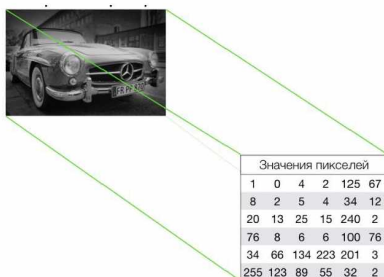


Рис. 1. Представление изображения в виде матрицы

Перед тем как распределить используемые наборы данных на обучающие и тестирующие, мы должны убедиться, что данные достаточно хорошо подготовлены. Для того чтобы использовать алгоритм CNN (Convolution Neural Network) необходимо задать параметры входных данных до стандартных, а именно привести их к значению  $32 \times 32 \times 3$ , то есть размерность изображения должна соответствовать приведенным параметрам. Эта размерность предполагает, что использованное изображение  $32 \times 32$  с тремя каналами и определёнными параметрами может использоваться с ML-алгоритмом машинного обучения. В качестве последнего используется метод опорных векторов SVM (Support Vector Machines), который способен обрабатывать изображения любого размера, но загружаемые изображения должны быть идентичными.

Алгоритм представляет собой традиционную нейронную сеть, добавляемую в конце слоев CNN. На рис. 2 показано визуальное представление полностью связанных слоев нейронной сети.

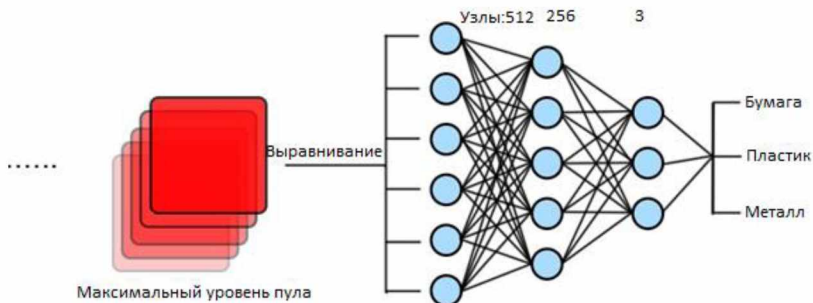


Рис. 2. Выравнивание выходных данных

Вместо использования менее эффективного процесса выполнения расчета для всего набора данных, можно использовать градиентный спуск SGD (Stochastic gradient descent), так как он выполняет вычисления только на основе случайного выбора для небольшого подмножества. Это уменьшит избыточность, так как мы вычисляем стоимость только одного примера для каждого шага. SGD – это одна из разновидностей градиентного спуска [4].

Далее рассмотрим программную реализацию модуля обработки изображения в среде MATLAB, которая будет выполнять поставленные задачи, такие как изменение цвета, изменение размера изображения [5]. Для решения данных задач применим toolbox, а именно функции `imresize`, `rgb2gray` и `gray2rgb`. Первая функция используется для изменения размера изображения и помогает конвертировать изображение до нужного размера, а вторые функции помогают изменять цвет входного изображения. В свою очередь функция `imwrite` записывает все выходные данные с указанием имени файла. Присутствуют также и недостатки данной функции, так как идет переопределение предыдущего файла.

На рис. 3 представлен фрагмент программной реализации модуля, который осуществляет изменение размера и цветокоррекцию.

```

for i = 1:length(trainingset.Labels)
    imgtrain = readimage(trainingset,i);
    trainingsetsize(i,1,1) = size(imgtrain);
end
minx_train = min(trainingsetsize(:,1));
miny_train = min(trainingsetsize(:,2));

for i = 1:length(testset.Labels)
    imgtest = readimage(testset,i);
    testsetsize(i,1,1) = size(imgtest);
end
minx_test = min(testsetsize(:,1));
miny_test = min(testsetsize(:,2));

minx = 150;
miny = 150;

for i=1:length(trainingset.Labels)
    if trainingsetsize(i,1) == minx || trainingsetsize(i,2) == miny
        RGBtrain = imread(trainingset.Files(i));
        RGBtrain2 = imresize(RGBtrain, [miny minx]);
        graysize = rgb2gray(RGBtrain2);
        imwrite(graysize,trainingset.Files(i))
    end
end

for i=1:length(testset.Labels)
    if testsetsize(i,1) == minx || testsetsize(i,2) == miny
        RGBtest = imread(testset.Files(i));
        RGBtest2 = imresize(RGBtest, [miny minx]);
        graysize = rgb2gray(RGBtest2);
        imwrite(graysize,testset.Files(i))
    end
end

for i = 1:length(trainingset.Labels)
    imgtrain = readimage(trainingset,i);
    trainingsetsize(i,1,1) = size(imgtrain);
end
minx_train = min(trainingsetsize(:,1));
miny_train = min(trainingsetsize(:,2));

for i = 1:length(testset.Labels)
    imgtest = readimage(testset,i);
    testsetsize(i,1,1) = size(imgtest);
end
minx_test = min(testsetsize(:,1));
miny_test = min(testsetsize(:,2));

minx = 150;
miny = 150;

for i=1:length(trainingset.Labels)
    if trainingsetsize(i,1) == minx || trainingsetsize(i,2) == miny
        RGBtrain = imread(trainingset.Files(i));
        RGBtrain2 = imresize(RGBtrain, [miny minx]);
        imwrite(RGBtrain2,trainingset.Files(i))
    end
end

for i=1:length(testset.Labels)
    if testsetsize(i,1) == minx || testsetsize(i,2) == miny
        RGBtest = imread(testset.Files(i));
        RGBtest2 = imresize(RGBtest, [miny minx]);
        imwrite(RGBtest2,testset.Files(i))
    end
end

```

Рис. 3. Фрагмент кода модуля для корректировки цвета, изменения размера и записи данных

На рис. 4 приведен фрагмент кода в среде MATLAB, позволяющий быстро изменить размер входного изображения, которое будет использоваться для глубокого обучения. Также рассмотрим расширение данных, данный процесс позволяет внести изменения в имеющиеся данные. Основной задачей этого метода является то, что повышается производительность модели, сокращаются затраты и устраняется проблема, связанная с переобучением. В данном методе используем формулу `augmentedImageDatastore`, которая устраняет необходимость в ручной обработке данных и позволяет дополнять пакет данных без изменения размеров и цветовой коррекции.

```

augmentedTrainingSet = augmentedImageDatastore([28 28 3],trainingSet, 'ColorPreprocessing','rgb2gray');
augmentedTestSet = augmentedImageDatastore([28 28 3],testSet, 'ColorPreprocessing','rgb2gray');
augmentedTrainingSet = augmentedImageDatastore([28 28 3],trainingSet, 'ColorPreprocessing','gray2rgb');
augmentedTestSet = augmentedImageDatastore([28 28 3],testSet, 'ColorPreprocessing','gray2rgb');

```

Рис. 4. Увеличение объема данных

Мы используем два типа данных, поскольку разделение на обучающий и тестовый наборы позволяет избежать проблем, связанных с пере-

обучением модели. Обучая модель на одном наборе данных, мы рискуем столкнуться с проблемами при тестировании на новых данных, так как модель может быть слишком адаптирована к обучающим данным и показывать низкую точность на новых примерах. Разделение на обучающий и тестовый наборы помогает проверить производительность модели на независимых данных и убедиться, что она способна обобщать информацию и давать необходимый результат на новых данных.

### **Выводы**

Разработанный нейросетевой модуль позволит повысить эффективность процесса сортировки бытовых отходов, обеспечивая более точное и быстрое распознавание различных типов отходов. Благодаря многослойной структуре, подобной зрительной коре головного мозга, нейронная сеть, положенная в основу модуля способна обучаться на примерах различных наборов данных. Это позволит разработанному модулю адаптироваться к различным условиям работы и повышать точность распознавания различных видов отходов.

### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Nikolov A., D'Aquin M., Motta E.* Unsupervised learning of link discovery configuration // Extended Semantic Web Conference, 2012. – pp.119-133.[https://www.researchgate.net/publication/262271720\\_Unsupervised\\_Learning\\_of\\_Link\\_Discovery\\_Configuration](https://www.researchgate.net/publication/262271720_Unsupervised_Learning_of_Link_Discovery_Configuration)
2. *Traore B. B., Kamsu-Foguem B., Tangara F.* Deep convolution neural network for image recognition // Ecological Informatics, 2018. – vol. 48. – Elsevier B.V. – pp. 257-268. [https://www.researchgate.net/publication/283526194\\_Fuzzy\\_Feature\\_subset\\_selection\\_for\\_metagenomics](https://www.researchgate.net/publication/283526194_Fuzzy_Feature_subset_selection_for_metagenomics)
3. *Albawi S., Mohammed T. A., Al-Zawi S.* Understanding of a convolutional neural network // Proceedings of 2017 International Conference on Engineering and Technology (ICET 2017). – 2017. – IEEE. – pp. 1-6. [https://www.researchgate.net/publication/319253577\\_Understanding\\_of\\_a\\_Convolutional\\_Neural\\_Network](https://www.researchgate.net/publication/319253577_Understanding_of_a_Convolutional_Neural_Network)
4. *Ditzler, G., Morrison, J.C., Lan, Y. et al.* Fuzzy: feature subset selection for metagenomics // BMC Bioinformatics 16, 358 (2015). <https://doi.org/10.1186/s12859-015-0793-8>.
5. *Gogul I., Kumar V. S.* Flower species recognition system using convolution neural networks and transfer learning // 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), – 2017. – IEEE. – pp. 1-6.

**Прудников Артем Александрович**, студент Института компьютерных технологий и информационной безопасности Южного федерального университета, Россия, город Таганрог, пер. Некрасовский 44, 347922: телефон +7(918)562-73-51, e-mail: aprudnikov@sfnedu.ru.

**Поленов Максим Юрьевич**, кандидат технических наук, доцент кафедры вычислительной техники Института компьютерных технологий и информационной безопасности Южного федерального университета, Россия, город Таганрог, пер. Некрасовский, 44, 347922, телефон: +7(8634)37-16-56, e-mail: mypolenov@sfnedu.ru.

**Prudnikov Artyom Alexandrovich**, student of the Institute of Computer Technology and Information Security of the Southern Federal University, Russia, Taganrog, 44 Nekrasovsky lane, 347922, phone: +7(918)562-73-51, e-mail: aprudnikov@sfnedu.ru

**Polenov Maxim**, Candidate of Engineering Sciences, Associate Professor of the Department of Computer Engineering at the Institute of Computer Technology and Information Security of the Southern Federal University, Russia, Taganrog, 44 Nekrasovsky lane, 347922, phone: +7(8634)37-16-56, e-mail: mypolenov@sfnedu.ru