

УДК 004.8

DOI: 10.46548/21vek-2021-1054-0011

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ОБНАРУЖЕНИЯ И УЧЕТА МУСОРА НА ТЕРРИТОРИИ МУНИЦИПАЛИТЕТА

© 2021

Казakov Олег Дмитриевич, кандидат экономических наук, доцент,
заведующий кафедрой «Информационные технологии»

Азаренко Наталья Юрьевна, магистрант

Брянский государственный инженерно-технологический университет

(241050, Россия, Брянск, проспект Ленина 26а, e-mails: it.kazakov@yandex.ru, salovanat@mail.ru)

Аннотация. Представленная статья содержит исследование по разработке интеллектуальной системы обнаружения и учета мусора на территории муниципалитета. Цель представленной статьи состоит в программной реализации и развертывании нейронной сети для классификации наличия мусора на территории города Брянска, а также программной реализации и развертывании нейронной сети для детектирования мусора в разрезе географических координат. Для проведения исследования подготовлен набор данных для обучения нейронной сети классификации наличия мусора, состоящий из набора изображений, полученных с помощью квадрокоптера *DJI Mini 2*, а также скаченных с бесплатных фотостоков. С использованием метода *Data Augmentation* набор данных расширен до 1872 изображений. Применив метод *Transfer Learning* к сверточной нейронной сети *ResNet18*, обучили нейронную сеть для классификации наличия мусора на территории города Брянска. Задача детектирования мусора в разрезе географических координат реализована с помощью метода *Transfer Learning* к нейронной сети *YOLOV5*. В статье представлена реализованная клиентская часть *web*-приложения «Детектирование мусора на территории муниципалитета» с помощью фреймворка *Flask* на языке программирования *Python*. Практическая значимость исследования состоит в том, что интеллектуальная система обнаружения и учета мусора на территории муниципалитета позволит упорядочить работу с твердыми коммунальными отходами, а также контролировать своевременность вывоза мусора, повысить качество уборки территории, контролировать несанкционированные свалки в самом начале их образования, снизить экологическую нагрузку и улучшить экологическую культуру граждан. Реализация интеллектуальной системы обнаружения и учета мусора на территории муниципалитетов в Брянской области одно из направлений проекта *Smart City*.

Ключевые слова: *Transfer Learning*, *Smart City*, интеллектуальная система обнаружения и учета мусора, изображения с квадрокоптера *DJI Mini 2*.

INTELLIGENT SYSTEM FOR DETECTION AND ACCOUNTING OF GARBAGE IN THE TERRITORY OF THE MUNICIPALITY

© 2021

Kazakov Oleg Dmitrievich, candidate of Economic Sciences, senior Lecturer,
head of the Department of Information Technology

Azarenko Natalia Yurievna, master's student

Bryansk State University of Engineering and Technology

(241050, Russia, Bryansk, Lenin Avenue, 26a, e-mails: it.kazakov@yandex.ru, salovanat@mail.ru)

Abstract. The presented article contains a study on the development of an intelligent system for detecting and accounting for garbage in the territory of the municipality. The purpose of the presented article is to programmatically implement and deploy a neural network for classifying the presence of garbage in the city of Bryansk, as well as software implementation and deployment of a neural network for detecting garbage in the context of geographic coordinates. For the study, a dataset was prepared for training a neural network for classifying the presence of garbage, consisting of a set of images obtained using a *DJI Mini 2* quadcopter, as well as those downloaded from free photo stocks. Using the *Data Augmentation* method, the dataset has been expanded to 1872 images. Applying the *Transfer Learning* method to the *ResNet18* convolutional neural network, we trained the neural network to classify the presence of garbage in the city of Bryansk. The task of detecting garbage in the context of geographic coordinates was implemented using the *Transfer Learning* method to the *YOLOV5* neural network. The article presents the implemented client part of the "Garbage Detection on the Municipality Territory" web application using the *Flask* framework in the *Python* programming language. The article presents the implemented client part of the "Garbage detection in the territory of the municipality" web application using the *Flask* framework in the *Python* programming language. The practical significance of the study lies in the fact that an intelligent system for detecting and accounting for debris on the territory of the municipality will allow to streamline the work with solid municipal waste, as well as control the timeliness of garbage disposal, improve the quality of cleaning the territory, control unauthorized landfills at the very beginning education, reduce the environmental burden and improve the environmental culture of citizens. The implementation of an intelligent waste detection and accounting system on the territory of municipalities in the Bryansk region is one of the areas of the *Smart City* project.

Keywords: *Transfer Learning*, *Smart City*, Intelligent Trash Detection and Accounting, Data from *DJI Mini 2* Quadcopter.

Введение. Научные работы в области классификации и детектирования мусора на территории делают упор на решении следующих задач:

1. Представление специального набора данных для обнаружения мусора беспилотными летательными аппаратами. В работе [1] представлен набор данных, получивший название *UAVWaste*, содержит на момент написания работы 772 изображения с 3716 помеченными вручную аннотациями мусора в городской и природной среде, такой как улицы, парки и газоны. Набор данных с размеченными аннотациями мусора ТАСО [2], который содержит гораздо более крупные объекты с околосредней точки зрения, что делает его менее пригодным для использования.

Индийские исследователи Ариан, Хандельвал Абхилаша Синха, Амит Датта, Рохан Нишант, Раши Гупта, Шрей Рай реализовали проект размещения мусора на территории Индии, где представили набор данных, полученный с автомобильных камер автомобилей, дронов и фотографий, сделанных заинтересованными в уборке мусора [3].

2. Решение задачи детектирования объекта (мусора) на изображении. Точное обнаружение объектов на изображениях достигло огромного прогресса за последние несколько лет. Эта задача, хотя и является тривиальной для человека, все еще остается сложной даже для современных компьютеров. Однако применение сверточных нейронных сетей (СНН) позволило значительно повысить точность обнаружения объектов по сравнению с классическими подходами, основанными на ручной обработке признаков [4-7], приблизив результаты к человеческому уровню производительности. В результате подавляющее большинство современных систем обнаружения объектов базируется на решениях, основанных на CNN. В работе задача обнаружения мусора решена с помощью *ResNet-18* с добавлением полностью связанного слоя для классификации [8].

3. Детектирование мусора в разрезе географических координат. Наиболее точной нейросетью, которая используется в настоящее время для детектирования объектов, является *YOLOv5*, обученная на датасет *COCO*.

YOLOv5 – сеть, которая дает высокий процент распознавания изображений.

Архитектура *YoloV5* представлена на рисунке 1.

Она состоит из трех частей: (1) магистраль: *CSPDarknet*, (2) шея: *PANet* и (3) Голова: Слой *YOLO*. Данные сначала вводятся в *CSPDarknet* для извлечения признаков, а затем передаются в *PANet* для объединения функций. *Yolo Layer* выводит результаты обнаружения (класс, оценка, местоположение, размер).

Актуальность темы исследования заключается в том, что засорение территории и выброшенный мусор, несмотря на прилагаемые усилия, по-прежнему является серьезной эстетической, экологической и социальной проблемой [13-15]. Уборка бытовых отходов в Брянской области – это обязанность регионального оператора по обращению с ТКО в Брянской

области - АО «Чистая планета» [11]. Однако ликвидация несанкционированных свалок и утилизация специфических видов отходов – это также забота местных органов власти [12], [16].

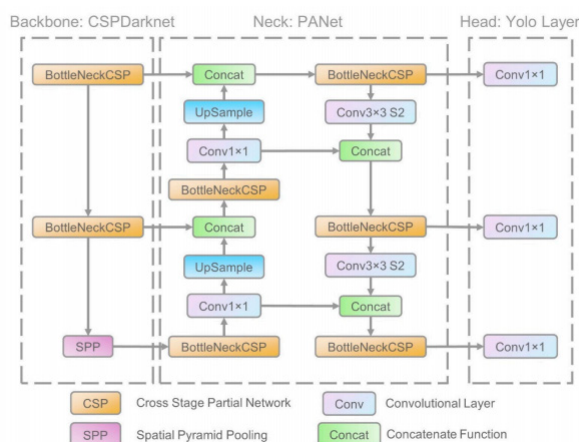


Рисунок 1 – Архитектура YoloV5 [9]

Целью исследования является программная реализация и развертывание нейронной сети для классификации наличия мусора на территории города Брянска, а также программная реализация и развертывание нейронной сети для детектирования мусора в разрезе географических координат. Для решения данной цели в статье поставлены следующие задачи.

Применяя метод *Transfer Learning* к сверточной нейронной сети *ResNet18*, обучить нейронную сеть для классификации наличия мусора на территории города Брянска. Входными параметрами модели сети должны являться фреймы видеопотока с частотой раз в 30 с. по данным видеосъемки квадрокоптера. В случае определения нейронной сетью класса «Наличие мусора» необходимо реализовать детектирование мусора в разрезе географических координат. Для этого, применяя метод *Transfer Learning* к нейронной сети *YOLOv5*, необходимо обучить нейронную сеть для детектирования мусора на территории города Брянска в разрезе географических координат. Входными параметрами модели сети должны являться фреймы видеопотока, а также файл метаданных с координатами *.str. Обнаруженный мусор необходимо вывести на географическую карту. Для снятия с учета убранного мусора необходимо реализовать пользовательский интерфейс.

Схема развертывания разрабатываемой системы представлена следующей схемой (рис. 2).

Материалы и результаты исследования. В проведенном исследовании использовался следующий технологический стек:

– *TensorFlow* – фреймворк для машинного обучения, разработанный компанией *Google* для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов.

– *PyTorch* – фреймворк машинного обучения для языка *Python* с открытым исходным кодом, созданный

на базе *Torch*, который используется для решения таких задач, как компьютерное зрение, обработка естественного языка.

– *Flask* – фреймворк для создания веб-приложений

на языке программирования *Python*, использующий набор инструментов *Werkzeug*, а также шаблонизатор *Jinja2*.

– СУБД: *SQLite*.

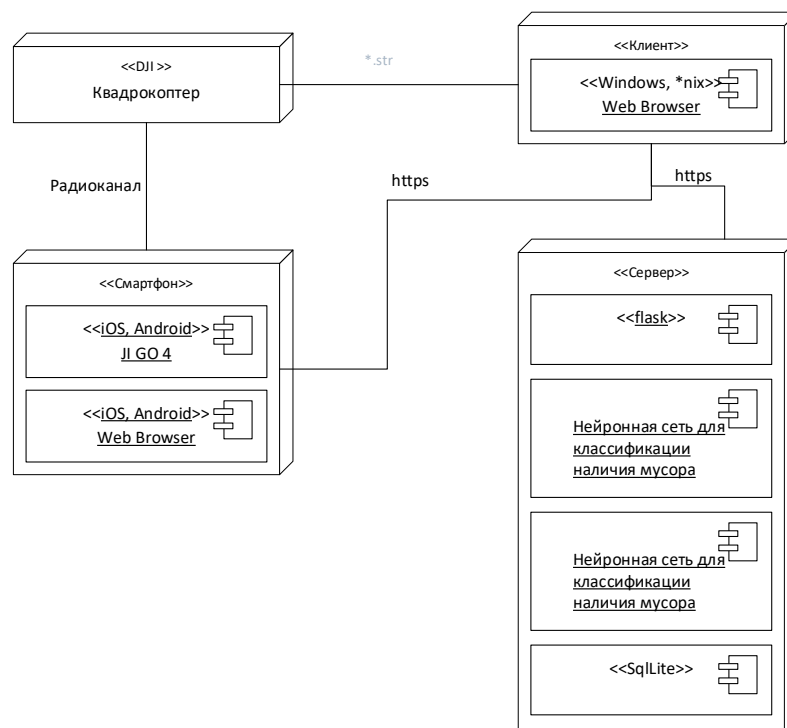


Рисунок 2 – Диаграмма развёртывания системы обнаружения и детектирования мусора (составлен авторами)

Этапы проведения исследования:

1. Подготовка набора данных для обучения нейронной сети классификации наличия мусора. Набор изображений из 112 фотографий, сделанных с помощью квадрокоптера *DJI Mini 2* в разрешении 4000 x 2250. Дополнительно с бесплатных фотостоков было скачено более 200 изображений с различными разре-

шениями.

В целях расширения исходного набора данных автором применен метод – *Data Augmentation*. Для этого был реализован класс *ImageDataGenerator* из *Python*-фреймворк *TensorFlow*.

Аугментация проводилась по следующим направлениям:

1. Поворот. Угол поворота задаем случайным значением в диапазоне от 0 до 360 градусов.



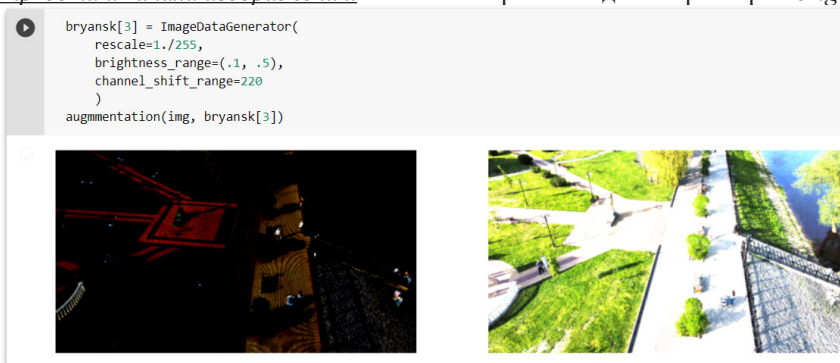
2. Смещение по ширине. Смещения на 1/2 доли



3. Смещение по высоте. Диапазоны изменений задаются параметром *height_shift_range*. Диапазоны изменений такие же, как и у смещения по ширине:



4. Изменение яркости и канала изображения. Изменение яркости задаем параметром *brightness_range*



5. Сдвиг изображений. Сдвиг выглядит следующим образом:



6. Масштабирование изображения. Ещё одним методом преобразования изображений, является его масштабирование. С помощью параметра *zoom_range* настраивается насколько изображение должно быть приближено/отдалено:



Таким образом исходный набор данных был расширен до 1872 изображений. «Скошенность» набора данных, т.е. присутствия изображения с различным разрешением, исправляется методом *transforms.Compose* из *PyTorch*.

Разметка объектов осуществлялась путем сортировки изображений в соответствующие директории, которые подаются на вход методу *ImageFolder*:

```
image_datasets = {x: datasets.ImageFolder(os.path.join(data_dir, x), data_transforms[x])
```

2. Обучение нейронной сети классификации наличия мусора. Исходя из поставленной задачи структура нейронной сети для классификации наличия мусора на территории города Брянска выглядит следующим образом:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 112, 112]	9,408
BatchNorm2d-2	[-1, 64, 112, 112]	128
ReLU-3	[-1, 64, 112, 112]	0
MaxPool2d-4	[-1, 64, 56, 56]	0
Conv2d-5	[-1, 64, 56, 56]	36,864
BatchNorm2d-6	[-1, 64, 56, 56]	128
ReLU-7	[-1, 64, 56, 56]	0
Conv2d-8	[-1, 64, 56, 56]	36,864
BatchNorm2d-9	[-1, 64, 56, 56]	128
ReLU-10	[-1, 64, 56, 56]	0
BasicBlock-11	[-1, 64, 56, 56]	0
Conv2d-12	[-1, 64, 56, 56]	36,864
BatchNorm2d-13	[-1, 64, 56, 56]	128
ReLU-14	[-1, 64, 56, 56]	0
Conv2d-15	[-1, 64, 56, 56]	36,864
BatchNorm2d-16	[-1, 64, 56, 56]	128
ReLU-17	[-1, 64, 56, 56]	0
BasicBlock-18	[-1, 64, 56, 56]	0
Conv2d-19	[-1, 128, 28, 28]	73,728
BatchNorm2d-20	[-1, 128, 28, 28]	256
ReLU-21	[-1, 128, 28, 28]	0
Conv2d-22	[-1, 128, 28, 28]	147,456
BatchNorm2d-23	[-1, 128, 28, 28]	256
Conv2d-24	[-1, 128, 28, 28]	8,192
BatchNorm2d-25	[-1, 128, 28, 28]	256
ReLU-26	[-1, 128, 28, 28]	0
BasicBlock-27	[-1, 128, 28, 28]	0
Conv2d-28	[-1, 128, 28, 28]	147,456
BatchNorm2d-29	[-1, 128, 28, 28]	256
ReLU-30	[-1, 128, 28, 28]	0
Conv2d-31	[-1, 128, 28, 28]	147,456
BatchNorm2d-32	[-1, 128, 28, 28]	256
ReLU-33	[-1, 128, 28, 28]	0
BasicBlock-34	[-1, 128, 28, 28]	0
Conv2d-35	[-1, 256, 14, 14]	294,912
BatchNorm2d-36	[-1, 256, 14, 14]	512
ReLU-37	[-1, 256, 14, 14]	0
Conv2d-38	[-1, 256, 14, 14]	589,824
BatchNorm2d-39	[-1, 256, 14, 14]	512
Conv2d-40	[-1, 256, 14, 14]	32,768
BatchNorm2d-41	[-1, 256, 14, 14]	512
ReLU-42	[-1, 256, 14, 14]	0
BasicBlock-43	[-1, 256, 14, 14]	0
Conv2d-44	[-1, 256, 14, 14]	589,824
BatchNorm2d-45	[-1, 256, 14, 14]	512
ReLU-46	[-1, 256, 14, 14]	0
Conv2d-47	[-1, 256, 14, 14]	589,824
BatchNorm2d-48	[-1, 256, 14, 14]	512
ReLU-49	[-1, 256, 14, 14]	0
BasicBlock-50	[-1, 256, 14, 14]	0
Conv2d-51	[-1, 512, 7, 7]	1,179,648
BatchNorm2d-52	[-1, 512, 7, 7]	1,024
ReLU-53	[-1, 512, 7, 7]	0
Conv2d-54	[-1, 512, 7, 7]	2,359,296
BatchNorm2d-55	[-1, 512, 7, 7]	1,024
Conv2d-56	[-1, 512, 7, 7]	131,072
BatchNorm2d-57	[-1, 512, 7, 7]	1,024
ReLU-58	[-1, 512, 7, 7]	0
BasicBlock-59	[-1, 512, 7, 7]	0
Conv2d-60	[-1, 512, 7, 7]	2,359,296
BatchNorm2d-61	[-1, 512, 7, 7]	1,024
ReLU-62	[-1, 512, 7, 7]	0
Conv2d-63	[-1, 512, 7, 7]	2,359,296
BatchNorm2d-64	[-1, 512, 7, 7]	1,024
ReLU-65	[-1, 512, 7, 7]	0
BasicBlock-66	[-1, 512, 7, 7]	0
AdaptiveAvgPool2d-67	[-1, 512, 1, 1]	0
Linear-68	[-1, 2]	1,026
Total params: 11,177,538		
Trainable params: 1,026		
Non-trainable params: 11,176,512		
Input size (MB): 0.57		
Forward/backward pass size (MB): 62.79		
Params size (MB): 42.64		
Estimated Total Size (MB): 106.00		

Первые 18 сверточных слоев взяты из нейронной сети *ResNet18*:

```
model_ft = models.resnet18(pretrained=True)
# замораживаем веса
for param in model_ft.parameters():
    param.requires_grad = False
```

Дополнительно мы создали линейный слой, у которого 2 выхода и входы определены по количеству классов *ResNet18*.

Функция потерь определена следующим образом:
criterion = nn.CrossEntropyLoss()

Оптимизатор определен следующим образом:
optimizer_ft=optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)

Количество эпох обучения: 25

Изменение значений функции потерь и метрики *Accuracy* в процессе обучения нейронной сети представлены на рисунке 3.

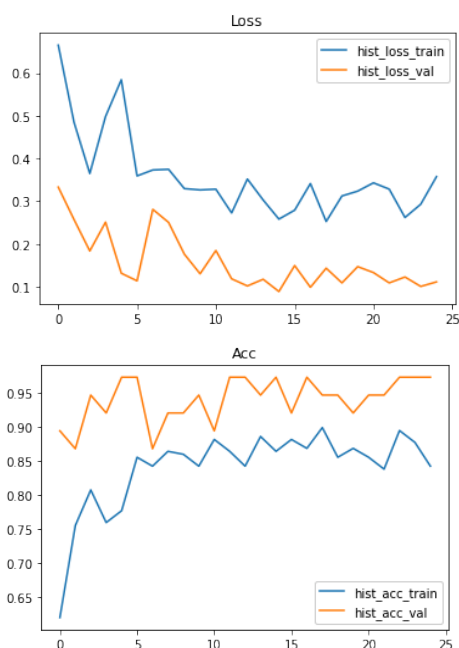


Рисунок 3 – Изменение значений функции потерь и метрики *Accuracy* в процессе обучения нейронной сети

3. Подготовка набора данных и обучение нейронной сети детектирования мусора в разрезе географических координат. Исходя из поставленной задачи нейронная сеть для детектирования мусора на территории города Брянска основывается на алгоритмах *YOLOv5*.

Detect.py выполняет логический вывод из различных источников, автоматически загружая модели из последней версии *YOLOv5* и сохраняя результаты в *runs/detect*.

```
$ python detect.py --source 0 # webcam
file.jpg # image
file.mp4 # video
path/ # directory
path/*.jpg # glob
'https://youtu.be/NUsoVlDfQZg' # YouTube video
'rtsp://example.com/media.mp4' # RTSP, RTMP, HTTP
stream
```

YOLO требует, чтобы набор данных был в формате *darknet*. Координаты прямоугольника должны быть нормализованы от 0 до 1.

Разметку изображений реализовали в программе *labellmg* (рис. 4).

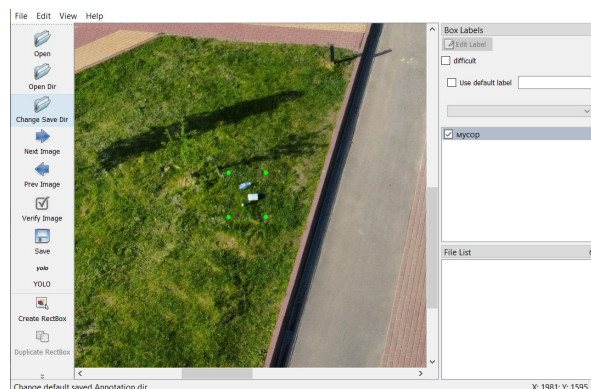


Рисунок 4 – Разметка изображений для детектирования мусора.

Для переноса обучения необходимо настроить конфигурационный файл, шаблон которого берем из исходных файлов *yolov5* по пути ... \yolov5_master\cfg. Затем сохраняем веса для настроенного конфигурационного файла.

Обучение проводилось на видеокарте *XFX Radeon RX 550 4GB*. Количество эпох: 570. По окончании обучения сеть сохранилась в файл *last.pt*.

Проект реализован в виде *web*-приложения с помощью фреймворка *Flask* на языке программирования *Python* (рис. 5).

В рамках полевых испытаний из трех объектов «Мусор», реально попавших в зону видимости квадрокоптера по маршруту вдоль Набережной города Брянска, было детектировано два объекта.

Деплой проекта осуществлялся на облачной *PaaS*-платформе – *Heroku*. Проект опубликован по адресу: <https://garbage32kod.herokuapp.com/>

Мы получаем изображения на сервере от дрона. Наша модель машинного обучения проверяет наличие мусора в изображениях и, если мусор обнаружен, он сохраняет изображение в базе данных *SQLite*. Мусор детектируется с помощью нейронной сети *YOLOv5* и выводятся координаты нахождения мусора на карту. Фрейм с детектированным мусором и изображения *Яндекс.карты* с отмеченным мусором в режиме реального времени видны эоактивисту / уборщику мусора.

Эоактивист/уборщик патрулирует свою зону, может менять график очистки мусора, не выходить на уборку, если мусора нет, наоборот, увеличивать количество выходов на уборку в случае обнаружения нового мусора сразу в нескольких местах. Приезжая на место, указанное в *web*-приложении, он убирает мусор, при этом выбирает команду «Мусор убран». Либо в случае ошибки нейросети, эоактивист / уборщик выбирает команду: «Это не мусор».

Важную роль играет здесь администрация, регулирующая благоустройство и решающая экологические

проблемы на территории муниципалитета. Муниципальные служащие видят все те области, запрос на уборку которых был подан как минимум 1 день назад и все еще не очищен, координируют деятельность муниципальных предприятий, отвечающих за уборку территории, либо служб ЖКХ. Увеличение плотности

мусора позволяет объявлять волонтерские акции по уборке мусора, привлекая эоактивистов, через соцсети и официальные сайты.

Тем самым, повысится эффективность уборки территории муниципалитетов от мусора, более рационально будет организована работа уборщиков.

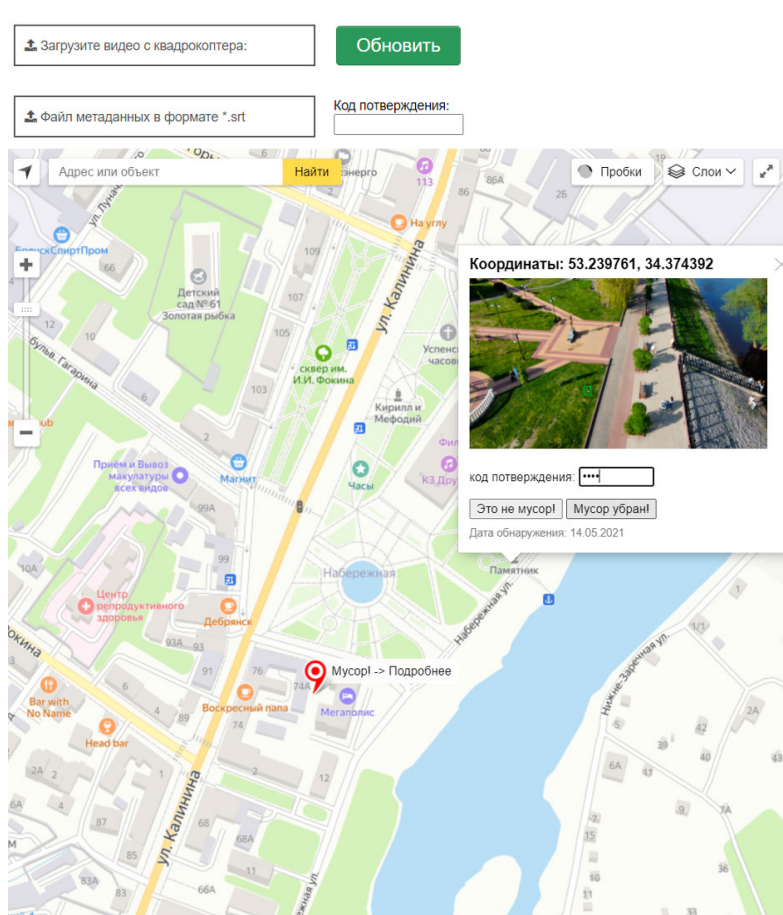


Рисунок 5 – Клиентская часть web-приложения «Детектирование мусора на территории муниципалитета»

Заключение. В статье получены результаты программной реализации и развертывания нейронной сети для классификации наличия мусора на территории города Брянска, а также программной реализации и развертывания нейронной сети для детектирования мусора в разрезе географических координат.

Подготовлен набор данных для обучения нейронной сети классификации наличия мусора, состоящий из набора изображений, полученных с помощью квадрокоптера *DJI Mini 2*, а также скаченных с бесплатных фотостоков. С использованием метода *Data Augmentation* набор данных расширен до 1872 изображений.

Применив метод *Transfer Learning* к сверточной нейронной сети *ResNet18*, обучили нейронную сеть для классификации наличия мусора на территории города Брянска.

Задача детектирования мусора в разрезе географических координат реализована с помощью метода *Transfer Learning* к нейронной сети *YOLOv5*.

Реализована клиентская часть web-приложения «Детектирование мусора на территории муниципали-

тета» с помощью фреймворка *Flask* на языке программирования *Python*.

Исследование, проведенное авторами статьи, имеет практическую значимость, так как разработанная интеллектуальная система позволит упорядочить работу с твердыми коммунальными отходами, а также контролировать своевременность вывоза мусора, повысить качество уборки территории, контролировать несанкционированные свалки в самом начале их образования, снизить экологическую нагрузку и улучшить экологическую культуру граждан. Интеллектуальная система обнаружения и учета мусора на территории муниципалитета – это создание *Smart City* в Брянской области [17].

Следует отметить возможность использования данного решения в управлении городским хозяйством Брянска, других муниципальных образований Брянской области.

Дальнейшие исследования авторов позволят изучить применение сверточной нейронной сети и метода *Transfer Learning* к решению других задач построения *Smart City* [18-20].

СПИСОК ЛИТЕРАТУРЫ:

1. Kraft M. et al. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle //Remote Sensing. – 2021. – Т. 13. – №. 5. – С. 965.
2. Proença, P.F.; Simões, P. TACO: Trash Annotations in Context for Litter Detection. arXiv 2020, arXiv:2003.06975
3. https://github.com/rondon1947/RA26_Ctrl-Alt-Elite_WIMDR
4. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
5. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
6. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 2009, 32, 1627–1645. [CrossRef] Remote Sens. 2021, 13, 965 16 of 18
7. Malisiewicz, T.; Gupta, A.; Efros, A.A. Ensemble of exemplar-SVMs for object detection and beyond. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 89–96.
8. <https://www.pluralsight.com/guides/introduction-to-resnet>
9. Glenn Jocher; Alex Stoken; Jirka Borovec; NanoCode012; Ayush Chaurasia; TaoXie; Liu Changyu; Abhiram V; Laughing; tkianai; yxNONG; Adam Hogan; lorenzomamma; AlexWang1900; Jan Hajek; Laurentiu Diaconu; Marc; Yonghye Kwon; oleg; wanghaoyang0106; Yann Defretin; Aditya Lohia; ml5ah; Ben Milanko; Benjamin Fineran; Daniel Khromov; Ding Yiwei; Doug; Durgesh; Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations Режим доступа: https://zenodo.org/record/4679653#.YK69_agzbc
10. Xu, R.; Lin, H.; Lu, K.; Cao, L.; Liu, Y. A Forest Fire Detection System Based on Ensemble Learning. Forests 2021, 12, 217. <https://doi.org/10.3390/f12020217>
11. Мирный А. Н. и др. Санитарная очистка и уборка населенных мест. – 2010.
12. Шкляр В. П., Хизов А. В. Особенности организации уборки мусора и его утилизации в городе Саратове //Инновации природообустройства и защиты окружающей среды. – 2019. – С. 325-327.
13. Нарбут Н. А. Экологический каркас как форма организации территории //Вестник Красноярского государственного аграрного университета. – 2008. – №. 4.
14. Горлов Д. М., Коваленко Л. В. К вопросу о роли прогрессивных технологий в современном менеджменте //Вестник академии знаний. – 2019. – №. 2 (31).
15. Абдулкаримова А. М. Экологические проблемы и перспективы переработки твердых бытовых отходов в Чеченской Республике //Природопользование и устойчивое развитие регионов России. – 2020. – С. 10-13.
16. Kazakov O.D., Kulagina N.A., Azarenko N.Y. Machine learning methods in municipal formation // Growth Poles of the Global Economy: Emergence, Changes and Future Perspectives. Сер. "Lecture Notes in Networks and Systems" Plek-hanov Russian University of Economics. Luxembourg, 2020. P. 339-346
17. Азаренко Н.Ю., Казаков О.Д. От интеллектуальной системы обнаружения и учета мусора на территории муниципалитета к созданию SMART CITY // Статистический анализ социально-экономического развития субъектов Российской Федерации: Материалы 8-ой Международной научно-практической конференции / Брянский государственный инженерно-технологический университет. – Брянск, 2021. С.17-20
18. Зотов В.Б. «Умный город»: методология построения системы //Муниципальная академия. – 2019. – №. 3. – С. 3-11.
19. Морозова Н.А., Пузевич А.В. Межведомственное взаимодействие на федеральном и муниципальном уровне в соответствии с концепцией «smart-city» //Современные научные исследования и разработки. – 2018. – №. 10. – С. 566-569.
20. Солдатов С. Smart City-город будущего //Современные технологии автоматизации. – 2015. – №. 2. – С. 24.

Статья поступила в редакцию 16.05.2021

Статья принята к публикации 16.06.2021