ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.42
CC BY 4.0
DOI: 10.21822/2073-6185-2024-51-3-123-129 Оригинальная статья/ Original article

Веб-сервис с моделью машинного обучения для мониторинга воздушного пространства

А.Д. Попов
Воронежский институт МВД России,
394065, г. Воронеж, пр. Патриотов, 53, Россия

Резюме. Цель. Целью исследования является разработка веб-сервиса детектирования воздушных объектов, обнаруживающего любой летящий объект, выделяя его на изображении и классифицирующий его на предмет угрозы, поскольку современные системы обнаружения воздушных объектов (ПВО) не всегда эффективно справляются с задачей обнаружения беспилотных летательных аппаратов (БПЛА) из-за их малых размеров, низкой высоты полета и использования материалов, малозаметных для радиолокационных станций. БПЛА, функционирующие без управления оператора, также затрудняют обнаружение по радиосигналам. Для эффективного обнаружения БПЛА предлагается использовать систему, основанную на оптическом сканировании неба вокруг охраняемых объектов. Такая система должна обладать возможностью автономной работы и включать в себя детекторы воздушных объектов, созданные на основе технологий компьютерного зрения и искусственного интеллекта (ИИ). **Метод**. Исследование и разработка веб-сервиса мониторинга воздушного пространства основаны на методах системного анализа, синтеза, дедукции. **Результат.** Спроектирована и разработана визуальная часть веб интерфейса; сформирован датасет из открытых источников для корректного обнаружения летящих объектов; разработан нейросетевой детектор для классификации летящих объектов представляющих опасность; разработан программный модуль, позволяющий проводить автоматическое обнаружение идентификационных флагов опасных воздушных объектов с последующим предоставлением отчетов в txt файлах в yolo формате (координаты нормализованы). **Вывод.** Разделение визуальной части сервиса позволит обеспечить распределенное развертывание серверной части, повышая гибкость и масштабируемость. Разработка панели управления администратора позволит эффективно контролировать работу сервиса, управлять настройками и пользователями. В результате внедрения этих улучшений, веб-сервис сможет: выполнять мониторинг неба вокруг охраняемых объектов, автоматически обнаруживая и классифицируя воздушные объекты и идентифицировать воздушные объекты по степени угрозы, предоставляя информацию для принятия необходимых мер.

Ключевые слова: БПЛА, РЛС, компьютерное зрение, веб-сервис, детектирование, архитектура, камеры, фреймворк Flask, датасет, Roboflow, искусственный интеллект, ultralytics, JupyterLab, YOLO, OpenCV, Minio

Для цитирования: А.Д. Попов. Веб-сервис с моделью машинного обучения для мониторинга воздушного пространства. Вестник Дагестанского государственного технического университета. Технические науки. 2024; 51(3):123-129. DOI:10.21822/2073-6185-2024-51-3-123-129

Web service with machine learning model for airspace monitoring

A.D. Popov
Voronezh Institute of the Ministry of Internal Affairs of Russia,
53 Patriotov Ave., Voronezh 394065, Russia

Abstract. Objective. The goal is to develop a web service for detecting aerial objects that detects a flying object, highlights it in an image and classifies it as a threat, since modern aerial object detection systems do not always cope with the task of detecting unmanned aerial

vehicles due to their small size, low flight altitude and the use of materials that are barely noticeable to radar stations. Unmanned aerial vehicles that operate without operator control make it difficult to detect by radio signals. To detect UAVs, it is proposed to use a system based on optical scanning of the sky around protected objects. The system should be capable of autonomous operation and include aerial object detectors created on the basis of computer vision and artificial intelligence technologies. **Method.** The research and development of the airspace monitoring web service are based on the methods of system analysis, synthesis, and deduction. **Result.** The visual part of the web interface has been designed and developed; a dataset has been formed from open sources for the correct detection of flying objects; a neural network detector has been developed for classifying flying objects that pose a danger; a software module has been developed that allows for the automatic detection of identification flags of dangerous air objects with subsequent provision of reports in txt files in yolo format (coordinates are normalized). **Conclusion.** Separation of the visual part of the service will allow for distributed deployment of the server part, increasing flexibility and scalability. Development of the administrator control panel will allow for effective control of the service, management of settings and users. As a result, the web service will be able to: monitor the sky around protected objects, automatically detecting and classifying air objects and identifying air objects by threat level, providing information for taking necessary measures.

Keywords: UAV, radar, computer vision, web service, detection, architecture, cameras, Flask framework, dataset, Roboflow, artificial intelligence, ultralytics, JupyterLab, YOLO, OpenCV, Minio

For citation: A.D. Popov. Web service with machine learning model for airspace monitoring. Herald of Daghestan State Technical University. Technical Sciences. 2024; 51(3): 123-129. DOI:10.21822/2073-6185-2024-51-3-123-129

Введение. В настоящее время все большую актуальность получает направление обнаружения БПЛА. Стандартные средства обнаружения воздушных объектов для данной задачи подходят не всегда. Для ПВО БПЛА имеют слишком низкую видимость в связи с малой высотой полета, малым размером и материалами корпуса, малозаметными для РЛС. Радиосканеры эфира на предмет обнаружения управляющих команд для БПЛА на характерных частотах также не всегда способны обнаружить угрозу, так как существуют БПЛА, движущиеся без управления оператором [1,2].

Постановка задачи. В качестве одного из способов решения задачи обнаружения БПЛА предлагается использование оптических средств сканирования неба вокруг охраняемых объектов для визуального обнаружения БПЛА. Чтобы такая система могла работать автономно, требуется разработка детекторов воздушных объектов на базе технологий компьютерного зрения и ИИ, которые могли бы заранее обнаружить опасный объект, классифицировать его и в случае угрозы уведомить оператора для дальнейшего принятия мер по его подавлению [3,4]. Разработка веб сервиса, позволяющего проводить загрузку видеофайлов, изображений и потокового видео с камер для классификации обнаруженного детектором любого воздушного объекта, является актуальной задачей.

Методы исследования. Архитектура системы. Систему планируется использовать на различных объектах критической инфраструктуры, а также для охраны частной территории от незаконного нарушения границ частной собственности. Общая схема функционирования веб-сервиса представлена на рис. 1.

Для создания визуальной части веб-сервиса были использованы языки разметки (HTML), таблиц стилей (CSS), программирования (JS).

Функциональная часть была реализована с использованием подхода прослушивания событий (add EventListener). Возможность сохранения текущего состояния страницы обусловлена использованием local Storage, который позволяет управлять настройками подключения камеры: добавления и удаления с сохранением состояния при перезагрузке страницы.

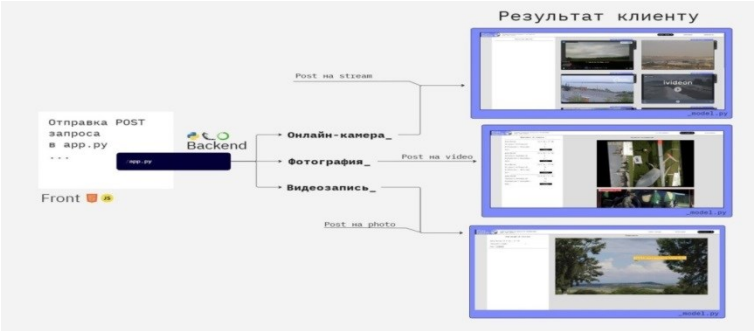

Рис. 1. Общая схема функционирования веб-сервиса
Fig. 1. General diagram of the web service functioning

Основой для серверной части веб сервиса послужил фреймворк Flask, его использование обеспечило гибкость и простоту в управлении HTTP-запросами и их обработку. Веб-сервис настроен на перехват и обработку входящих запросов от пользователей. Загруженные файлы сохранялись в специально отведенных каталогах на сервере, после чего происходил их анализ с помощью модели YOLOv8x и возвращались пользователю.

Данная модель была выбрана за ее высокую точность и быстродействие в задачах детекции объектов по сравнению с другими моделями YOLO. Модель обучена на расширенном датасете, основу которого составили размеченные изображения из открытых источников [5,6]. На начальном этапе работы системы происходит добавление видеофайлов, множества изображений и стримингового видеопотока рис. 2 - 4.

Главная страница представлена формой отображения онлайн камер, с возможностью их добавления: указания названия и ссылки для подключения рис. 2.

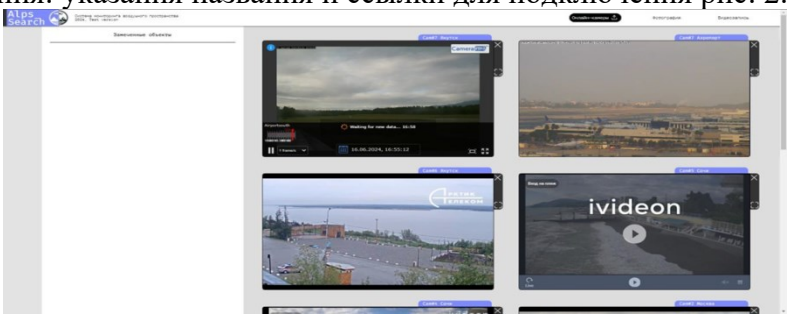

Рис. 2. Интерфейс страницы детектирования летательных объектов по стриминговому видеопотоку
Fig. 2. Interface of the page for detecting flying objects by streaming video stream

На рис. 3 представлена страница анализа изображений, особенность которой заключается добавление множества отдельный файлов в виде изображений.

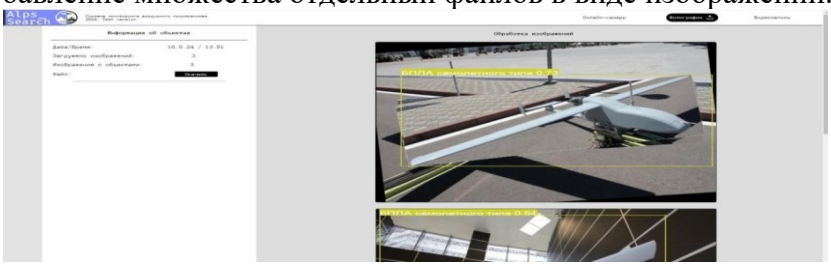

Рис. 3. Интерфейс страницы детектирования летательных объектов по фотографиям
Fig. 3. Interface of the page for detecting flying objects from photographs

На рис. 4 представлена страница анализа видеофайлов, особенность которой заключается в полной загрузке видеофайла на серверную часть веб-сервиса для обработки нейронной сетью.

После происходит отрисовка объекта на визуальной части веб сервиса, с подсвечиванием рамкой объекта в вероятностном соотношении. В случае идентификации наиболее опасных объектов происходит сохранение видеофрагмента для последующего предоставления оператору веб сервиса.

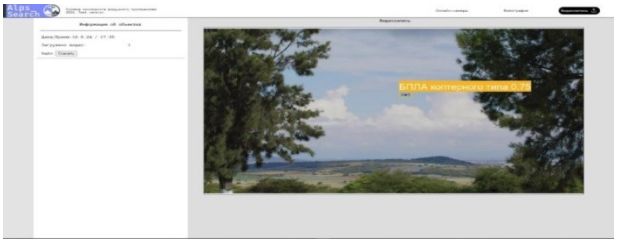

Рис. 4. Интерфейс страницы детектирования летательных объектов по видеофайлу
Fig. 4. Interface of the page for detecting flying objects by video file

Полученный массив изображений, видеофайл и стриминговый видеопоток подается на вход нейронной сети, которая производит классификацию и отбор полученных фрагментов, представляющих наибольшую опасность. При обработке входных данных автоматически формируется отчет с идентификацией флагов обнаружения опасных воздушных объектов в txt файлах в yolo формате, с возможностью скачивания в архиве рис. 5.

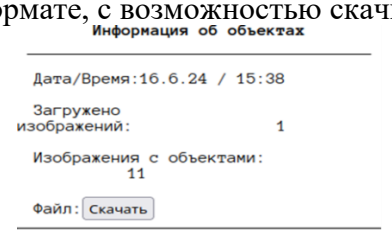

Рис. 5. Отчет с идентификацией флагов обнаружения опасных воздушных объектов
Fig. 5. Report with identification of dangerous air object detection flags

Минимальная комплектация сервера:

- операционная система: Ubuntu22.04 и выше;
- тактовая частота процессора: не менее 2 ГГц, Intel Core i5 8-го поколения или аналогичный;
- оперативная память: не менее 8 Гб;
- видеокарта: не менее NVIDIA GeForce GTX 1650 или аналоги;
- жесткий диск: не менее 30 Гб свободного дискового пространства;
- стабильная подключение к сети Интернет.

Обсуждение результатов. Описание алгоритмов, использованных при разработке веб-сервиса обнаружения БПЛА в воздушном пространстве:

1.1. Описание алгоритма автоматизированного сбора данных из открытых источников. В качестве примеров открытых источников данных используются: видеохостинг youtube, различные телеграмм каналы, сайты, СМИ, а также предоставленный набор данных.

1.2. Описание алгоритма искусственного интеллекта, позволяющего проводить автоматическую классификацию и отбор фотографий, позволяющих предупредить потенциальную угрозу в воздушном пространстве.

Для обучения искусственной нейронной сети было размечено порядка 10000 фотографий, создано 5 классов обнаружения объектов в: БПЛА самолетного типа; БПЛА коптерного типа; самолет; вертолет; птица.

Для разметки изображений и объединения их в набор данных для обучения нейросетевой модели использовалась платформа Roboflow (рис. 6), также при создании датасета были использованы фрагменты, на которых присутствовало несколько классов объектов [7]. Roboflow представляет собой платформу для разработчиков компьютерного зрения, которая позволяет улучшить сбор данных, их предварительную обработку и методы обучения моделей.

В ходе разработки алгоритма ИИ, позволяющего проводить автоматическую классификацию и обнаружение летающих объектов, представляющих потенциальную угрозу, использовалась среда разработки JupyterLab[8] с использованием библиотеки ultralytics и модулями:

1) YOLO – предоставляет функциональность для работы с моделями YOLO (You Only Look Once), популярными в задачах детекции объектов.

2) Hub – используется для взаимодействия с Ultralytics Hub, облачной платформой, предназначенной для управления, тренировки и развертывания моделей YOLO.

3) OpenCV (cv2) – это библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений и др. [9].

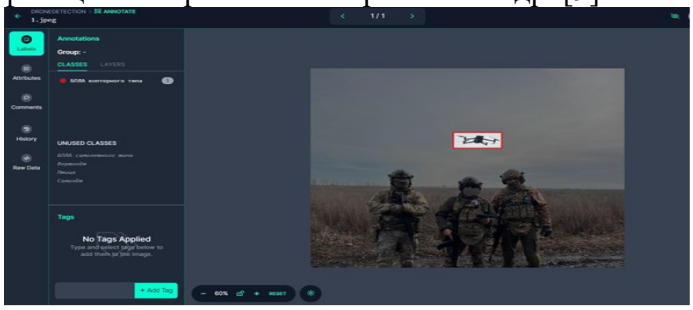

Рис.6. Процесс разметки изображений в платформе Roboflow
Fig. 6. The process of image marking in the Roboflow platform

Обоснование использования модели искусственного интеллекта **YOLOv8x** [10].

Model: Версия модели YOLO. Например, YOLOv5n, YOLOv6n и т.д.

Эти версии представляют различные архитектуры и конфигурации модели YOLO, разработанные для разных целей, таких как повышение точности, снижение вычислительной сложности и т. д.

Size (pixels): Размер входного изображения, которое используется моделью для предсказания. Обычно это фиксированный размер, например, 640x640 пикселей.

mAPval (50-95): Средняя точность предсказаний (mean Average Precision) на валидационном наборе данных, измеренная по порогам IoU (Intersection over Union) от 50% до 95%. Это важный показатель качества модели, характеризующий, насколько точно модель может предсказывать объекты в изображениях.

Speed CPU (мс): Время выполнения одного предсказания на процессоре (CPU), измеренное в миллисекундах. Пустые значения могут означать, что данные не предоставлены или не измерены.

Speed T4 GPU (мс): Время выполнения одного предсказания на графическом процессоре NVIDIA T4 (GPU), измеренное в миллисекундах. Пустые значения могут означать, что данные не предоставлены или не измерены. Сравнительная таблица производительности моделей YOLO (рис. 7).

Models

Model	size (pixels)	mAP^{val} 50-95	Speed CPU (ms)	Speed T4 GPU (ms)	params (M)	FLOPs (B)
YOLOv5n	640	28.0	-	-	**1.9**	**4.5**
YOLOv6n	640	35.9	-	-	4.3	11.1
YOLOv8n	640	**37.5**	-	-	3.2	8.9
YOLOv5s	640	37.4	-	-	7.2	16.5
YOLOv6s	640	43.5	-	-	17.2	44.2
YOLOv8s	640	**44.7**	-	-	11.2	28.8
YOLOv5m	640	45.4	-	-	21.2	49.0
YOLOv6m	640	49.5	-	-	34.3	82.2
YOLOv8m	640	**50.3**	-	-	25.9	79.3
YOLOv5l	640	49.0	-	-	46.5	109.1
YOLOv6l	640	52.5	-	-	58.5	144.0
YOLOv7	640	51.2	-	-	36.9	104.7
YOLOv8l	640	**52.8**	-	-	43.7	165.7
YOLOv5x	640	50.7	-	-	86.7	205.7
YOLOv7-X	640	52.9	-	-	71.3	189.9
YOLOv8x	640	**53.7**	-	-	68.2	258.5

Рис.7. Сводная таблица характеристик моделей YOLO
Fig. 7. Summary table of YOLO model characteristics

Params (M): Количество параметров в модели, измеренное в миллионах (M). Параметры включают веса и смещения (biases) в нейронных сетях и являются показателем сложности модели. Чем больше параметров, тем сложнее модель.

FLOPs (B): Количество операций с плавающей точкой в секунду (Floating Point Operations per Second), измеренное в миллиардах (B). Это показатель вычислительной сложности модели. Чем больше FLOPs, тем больше вычислительных ресурсов требуется для работы модели.

YOLOv8x выделяется среди других моделей благодаря самой высокой точности (mAP 53.7), что делает её идеальной для задач, требующих максимальной точности обнаружения объектов. Несмотря на большее количество параметров и FLOPs, она обеспечивает высокую вычислительную мощность и эффективность использования памяти по сравнению с другими моделями, такими как YOLOv5x и YOLOv7. Таким образом, YOLOv8x является лучшим выбором для высокоточных применений, где доступны мощные вычислительные ресурсы.

1.3. Развертывание веб-сервиса мониторинга воздушного пространства.

Развертывание веб сервиса предполагается с использованием технологии контейнеризированной Docker [11,12]. Данная технология обеспечивает лучшую изоляцию приложения, легковесность, быструю переносимость на другие системы, мастабируемостью, упрощенным подходом к управлению версиями.

В связи с ограничение размера загрузки файлов на GitHub необходимо скачать модель по адресу https://disk.yandex.ru/d/qU_n3h_2Rd8WEw.

1) Клонирование репозитория;
git clone https://github.com/verlliann/LCT.git
2) После скачивания требуется разархивировать архив;
unzip lct23_AlpsSearch.zip
3) Перенести файл drone.pt в каталог проекта по адресу ./static/python
mv drone.pt ./static/python
4) Сборка Dockerfile
sudo docker build -t lct_cuda05.
5) Развертывание контейнера
sudo docker run --gpus all -p 80:80 --name=lct_cuda05 lct_cuda05

Вывод. В качестве перспективного направления развития веб-сервиса мониторинга воздушного пространства может послужить развертывание Minio, для сохранения видеофайлов и отчетов, создание обособленной визуальной части с целью распределенного развертывания серверной части, разработка панели администрирования. Таким образом, использование веб-сервиса позволит осуществлять сканирования неба вокруг охраняемых объектов и идентифицировать воздушные объекты по степени опасности.

Исходя из поставленной цели, в ходе работы был разработан веб сервис для детектирования воздушных объектов. Он позволяет загружать видеофайлы, изображения и потоковое видео с камер, чтобы осуществить классификацию обнаруженных объектов. Данная система предназначена для использования на объектах критической инфраструктуры и охраны частной территории.

Библиографический список:

1. Беспилотные летательные аппараты: библиографический указатель /сост. О. В. Давыденко; под ред. Н. Н. Астаповой. – Кемерово: ИИО Кузбасской ГСХА, 2021. – 23 с.
2. Астахова, Н. Л. Дроны и их пилотирование. С чего начать / Н. Л. Астахова, В. А. Лукашов. - СПб.: БХВ-Петербург, 2021. – 224 с.: ил.
3. Дудкин А.А., Ганченко В.В., Инютин А.В., Марушко Е.Е. Идентификация и классификация объектов на изображениях, полученных с помощью съемочных средств БПЛА и орбитального базирования // Системный анализ и прикладная информатика. 2022. № 4. С. 30–37.
4. Бубеньщиков А.А., Кастырин М.И., Потаенков Г.О. Разработка аппаратно-программного модуля обнаружения и идентификации БПЛА с запретом его доступа в охраняемую зону на основе подмены сигналов управления//Сборник: Техника и безопасность объектов уголовно-исполнительной системы. Сб. мат. Международной научно-практической конференции. Иваново, 2022. С. 186–189.
5. Создание и развертывание моделей компьютерного зрения//URL: https://roboflow.com/ (дата обращения 01.06.2024).
6. Струков С.А., Печенкин А.Ю., Белослудцев В.Н. Реализация принципов определения расстояния до объектов на базе искусственной нейронной сети // Сборник: Приборостроение в XXI веке – 2022.

интеграция науки, образования и производства. Сборник материалов XVIII Всероссийской научно-технической конференции. Ижевск, 2023. С. 56–60.
7. Цурихин О.И., Сокольников В.В. Процесс создания выборки для обучения нейросети с помощью Roboflow // Информационные технологии в строительных, социальных и экономических системах. 2024. № 1–2 (31–32). С. 58–61.
8. Ochkov V.F., Tikhonov A.I., Stevens A. Jupyter notebook, Jupyterlab-integrated environment for stem education // В сборнике: 2022 6th International Conference on Information Technologies in Engineering Education, Inforino 2022 - Proceedings. 6. 2022.
9. Ильичев В.Ю. Применение библиотеки OpenCV языка Python для распознавания образов объектов // Системный администратор. 2021. № 7-8 (224-225). С. 130-132.
10. Практическое применение моделей YOLO и ResNet для обнаружения нежелательных предметов на фотографиях // URL: https://habr.com/ru/articles/761200/ (дата обращения 01.06.2024).
11. Chae M.S., Lee H.M., Lee K. A. Performance comparison of Linux containers and virtual machines using Docker and Kvm // Cluster Computing. 2019. Т. 22. № Suppl. 1. С. 1765–1775.
12. Захарчёнок В.Ф., Бизюк А.Н. Контейниризация и развертывание приложений с помощью Docker и Docker-compose // В сборнике: Материалы докладов 56-й международной научно-технической конференции преподавателей и студентов. в двух томах. Витебск, 2023. С. 80–82.

References:

1. Unmanned aerial vehicles: bibliographic index. Compiled by O. V. Davydenko; edited by N. N. Astapova. - Kemerovo: IIO Kuzbass State Agricultural Academy, 2021; 23. (In Russ)
2. Astakhova N. L., Lukashov V. A. Drones and their piloting. Where to start. St. Petersburg: BHV-Petersburg, 2021; 224l. (In Russ)
3. Dudkin A. A., Ganchenko V. V., Inyutin A. V., Marushko E. E. Identification and classification of objects in images obtained using UAV and orbital-based survey equipment. Systems analysis and applied informatics. 2022; 4.: 30–37. (In Russ)
4. Bubenshchikov A.A., Kastyrin M.I., Potaenkov G.O. Development of a hardware and software module for detecting and identifying UAVs with a ban on their access to a protected area based on the substitution of control signals. In the collection: Technology and safety of penal system facilities. collection of materials of the International scientific and practical conference. Ivanovo, 2022; 186–189. (In Russ)
5. Creation and deployment of computer vision models//https://roboflow.com/ (date of access 01.06.2024).
6. Strukov S.A., Pechenkin A.Yu., Belosludtsev V.N. Implementation of the principles of determining the distance to objects based on an artificial neural network. In the collection: Instrument making in the 21st century – 2022. integration of science, education and production. Collection of materials of the XVIII All-Russian scientific and technical conference. Izhevsk, 2023;56–60. (In Russ)
7. Tsurikhin O.I.,Sokolnikov V.V.The process of creating a sample for training a neural network using Roboflow. *Information technologies in construction, social and economic systems*. 2024;1–2(31–32):58–61. (In Russ)
8. Ochkov V.F., Tikhonov A.I., Stevens A. Jupyter notebook, Jupyterlab-integrated environment for stem education. 2022 6th International Conference on Information Technologies in Engineering Education, Inforino 2022 - Proceedings. 6. 2022.
9. Ilyichev V.Yu. Application of the OpenCV library of the Python language for object pattern recognition // System administrator. 2021;7-8 (224-225):130-132. (In Russ)
10. Practical application of YOLO and ResNet models for detecting unwanted objects in photographs // URL: https://habr.com/ru/articles/761200/ (date of access 01.06.2024) (In Russ)
11. Chae M.S., Lee H.M., Lee K. A. Performance comparison of Linux containers and virtual machines using Docker and Kvm. *Cluster Computing*. 2019; 22(l. 1):1765–1775.
12. Zakharchenko V.F., Bizyuk A.N. Containerization and deployment of applications using Docker and Docker-compose. *In the collection: Proceedings of the 56th international scientific and technical conference of teachers and students. in two volumes*. Vitebsk; 2023:80–82.

Сведения об авторе:
Попов Антон Дмитриевич, кандидат технических наук, доцент кафедры автоматизированных информационных систем органов внутренних дел; anton.holmes@mail.ru
Information about the author:
Anton D. Popov, Cand. Sci. (Eng.), Assoc. Prof., Department of automated information systems of internal organs; anton.holmes@mail.ru.

Конфликт интересов/Conflict of interest.
Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.
Поступила в редакцию/ Received 05.07.2024.
Одобрена после рецензирования / Reviced 30. 07.2024.
Принята в печать /Accepted for publication 30.07.2024.