

В. А. Архипов^{1}, А. А. Шаранов¹*

Обучение нейросети для поиска людей по фотоснимкам, полученным с БПЛА

¹ Сибирский государственный университет геосистем и технологий, г. Новосибирск, Российская Федерация

* e-mail: arhipov-vo2020@sgugit.ru

Аннотация. Работа посвящена обучению нейросети для поиска людей по фотоснимкам, полученным с беспилотного летательного аппарата. В процессе выполнения работы был проведен анализ предметной области, подготовлены фотоснимки для обучения нейронной сети, обучена модель RT-DETR с помощью фреймворка Ultralytics на датасетах Lacmus Drone Dataset и Vis Drone Dataset, спроектировано и разработано кроссплатформенное (для ОС Windows и Linux) приложение на python 3.12 с помощью библиотеки пользовательского интерфейса DearPyGUI для работы с обученной нейросетью, а также проведена оценка точности и скорости нейросети на реальных данных. Итоговая точность модели составила 95 % на датасете LaDD. Разработанное программное обеспечение позволит ускорить процесс обработки изображений и повысить шанс на обнаружение потерявшегося человека.

Ключевые слова: нейронная сеть, python, датасет, ultralytics, пользовательский интерфейс, dearpygui

V. O. Arkhipov^{1}, A. A. Sharapov¹*

Training a neural network to search for people in photos obtained from a UAV

¹ Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation

* e-mail: arhipov-vo2020@sgugit.ru

Abstract. The work is dedicated to training a neural network for finding people through photographs taken from an unmanned aerial vehicle. During the execution of the work, a subject area analysis was conducted, photographs were prepared for training the neural network, the RT-DETR model was trained using the Ultralytics framework on the Lacmus Drone Dataset and Vis Drone Dataset, a cross-platform (for Windows and Linux OS) application was designed and developed in Python 3.12 using the DearPyGUI user interface library to work with the trained neural network, and the accuracy and speed of the neural network on real data were assessed. The final model accuracy was 95% on the LaDD dataset. The developed software will speed up the image processing and increase the chance of finding a missing person.

Keywords: neural network, python, dataset, ultralytics, user interface, dearpygui

Введение

В последнее время для поиска людей спасательные отряды стали применять БПЛА. С помощью дрона они облетают всю область поиска, делают снимки, и потом осматривают эти снимки на наличие людей. Проблема заключается в том,

что за день дрон может сделать около 10 000 снимков, и даже при условии, что их будут просматривать около 30 человек, это может занять около 8 часов, за это время человеческий глаз может устать и упустить важные моменты [17, 19].

Для того чтобы ускорить обработку снимков и увеличить шанс на обнаружение человека можно использовать нейросеть. Основной целью работы является обучение нейронной сети для поиска людей по фотоснимкам, чтобы упростить и ускорить процесс обработки фотоснимков.

Анализ предметной области

Анализ предметной области включал в себя изучение проблематики, анализ существующих решений, разработку требований к ПО, анализ основных параметров глубокого обучения.

Существует несколько готовых решений:

1. «Beeline AI – Поиск людей», применяемое отрядом «ЛизаАлерт». Для него заявлены следующие характеристики [18, 21]:

- точность – 98 % (не указано, на каких данных);
- скорость распознавания – 0,3 мс (не указано, на каком устройстве).

2. Open-source проект Lastus – единственное бесплатное и общедоступное приложение, которое удалось найти и запустить [20].

Преимущества:

- хорошая скорость обработки;
- кроссплатформенность;
- производительность интерфейса.

Недостатки:

- ограниченная возможность кастомизации интерфейса;
- при просмотре и прокрутке изображения FPS может сильно проседать;
- долгая загрузка изображений (на открытие 1 000 изображений тратится 2 минуты, на 8 000 изображений – уже 16 минут).

Обучение нейросети для поиска людей

Для обучения были подобраны подходящие датасеты LaDD и Vis Drone (рис. 1) [6, 15].

Существует множество фреймворков для обучения нейросетей-детекторов. Одни из самых популярных – это Ultralytics, Detectron2, MMDetection [3, 7, 14]. С ними были обучены несколько моделей на платформах Google Colab и Kaggle, для экономии времени было взято только 100 изображений из датасета LaDD. Результаты пробного обучения представлены в табл. 1.

На основе результатов пробного обучения была выбрана модель RT-DETR из фреймворка Ultralytics, затем она обучалась на RTX 4090. В процессе были испробованы различные значения гиперпараметров (табл. 2).

Итоговая точность модели составила 95 % на датасете LaDD, а время обработки изображения на ноутбучном процессоре Ryzen 5 2500U ~ 8 секунд.



Рис. 1. Изображение из датасета VDD

Разработка программного обеспечения

Были выделены основные требования к приложению:

- производительность и отзывчивость интерфейса;
- кроссплатформенность (Windows и Linux);
- стабильность;
- возможность изменения темы (отдельно цвет каждого элемента);
- возможность выбора ускорителя для нейросети.

Были найдены несколько подходящих UI-библиотек, а именно Avalonia UI, DearPyGui и FLTK [2, 5, 16]. При выборе библиотеки учитывалась производительность и объем программного кода, написанных на них приложений. Было разработано по тестовому приложению для каждой библиотеки, затем с помощью счетчика был измерен их FPS (табл. 3) [22].

На основе полученных результатов, а также с учетом объема программного кода была выбрана библиотека DearPyGui.

Итоговый стек технологий разработки включает в себя: Python 3.12, VS Code и библиотеки: dearpygui, xdialog, pillow, numpy, CUDA, opencv, auto-py-to-exe, ruyaml, onnxruntime-gpu, onnxruntime [1, 2, 8–10, 12, 13].

Интерфейс приложения представлен на рис. 2.

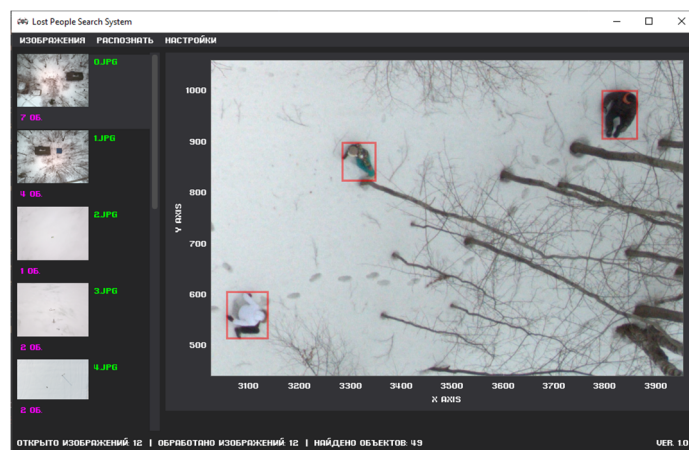


Рис. 2. Интерфейс приложения

Обученная модель была встроена в приложение и протестирована на изображениях из видео, взятых из свободного доступа в сети интернет [4, 11]. Для оценки модели был выбран эталон – модель из приложения Lacmus. Порог точности прогнозов был выставлен на 15 % для обеих моделей (значения скорости в таблице указаны в изображениях в секунду) (табл. 4).

В целом обученная модель оказалась медленнее при обработке на CPU, но при этом точнее в 1,5 раза, а скорость обработки на видеокарте оказалась даже выше. Таким образом, можно сказать, что модель была обучена успешно.

Результаты

Результаты представлены в табл. 1–4.

Таблица 1

Результаты пробного обучения

Фреймворк	Модель	mAP50-95	mAP50	Время обучения (сек.)
Ultralytics	rtdetr-l	0,6964	0,9137	3110
	yolov8l	0,5504	0,8463	2509
	yolov5lu	0,6711	0,8914	2621
Detectron2	retinanet R 50 FPN 3x	0,4964	0,7966	6249
	faster rcnn R 50 FPN 3x	0,5860	0,9348	5821
MMDetection	sparse-rcnn	0,0200	0,0990	2724
	faster rcnn R 50 FPN 3x	0,4540	0,8190	4021
	deformable-detr	0,2690	0,6470	2456

Таблица 2

Значения точности при различных значениях гиперпараметров

Набор данных	Гиперпараметры				Точность (mAP50)	Количество затраченных эпох
	lr0	batch	momentum	lrf		
Vis Drone Dataset	0,001	8	0,937	0,02	0,534	19
	0,0001	8	0,937	0,02	0,734	92
	0,00001	8	0,937	0,02	0,580	193
Lacmus Drone Dataset	0,0001	8	0,937	0,01	0,825	342
	0,0001	8	0,937	0,001	0,807	201
	0,0001	8	0,937	0,02	0,845	263

Таблица 3

Результаты измерения FPS

Устройство	FLTK	DearPyGUI	AvaloniaUI	WPF
AMD Ryzen 5 2500U, Vega Mobile, 1920x1080	23	20	11	4
AMD Ryzen 5 5600X, GTX 1060 3gb, 2560x1440	30	33	23	6

Результаты тестирования моделей

	RT-DETR	YOLOv5 (Lacmus App)
Точность	95 %	60 %
Распознано объектов	1263 из 1329	791 из 1329
Скорость обработки на AMD Ryzen 5 2500U	0,1	0,14
Скорость обработки на AMD Ryzen 5 5600X	0,73	1,24
Скорость обработки на RTX 3060	2,98	1,23

Заключение

В результате выполнения работы была обучена нейросеть для поиска людей по фотоснимкам, полученным с БПЛА, а также разработано приложение, имеющее следующий функционал:

- открытие изображений;
- обработка изображений;
- просмотр изображений;
- просмотр количества открытых и распознанных изображений, найденных объектов;
- изменение цветовой схемы;
- выбор ускорителя для нейросети.

Разработка позволит ускорить и упростить обработку фотоснимков, а также позволит повысить шансы на обнаружение человека.

Благодарности

Автор благодарит Сибирский государственный университет геосистем и технологий за предоставление доступа к графическому ускорителю RTX 4090.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. CUDA Installation Guide for Microsoft Windows: [сайт] – URL: <https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html/> (дата обращения: 10.05.2024).
2. Dear PyGui's Documentation: [сайт] – URL: <https://dearpygui.readthedocs.io/en/latest/> (дата обращения: 10.05.2024)
3. Detectron2 Docs: [сайт] – URL: <https://detectron2.readthedocs.io/en/latest/> (дата обращения: 10.05.2024).
4. DroneStock: [сайт] – URL: <https://dronestock.com/categories/people/> (дата обращения: 10.05.2024).
5. FLTK Documentation: [сайт] – URL: <https://www.fltk.org/documentation.php/> (дата обращения: 10.05.2024).
6. Lacmus Drone Dataset: [сайт] – URL: <https://www.kaggle.com/datasets/mersico/lacmus-drone-dataset-ladd-v40/> (дата обращения: 10.05.2024).

7. MMDetection Docs: [сайт] – URL: <https://mmdetection.readthedocs.io/en/dev-3.x/> (дата обращения: 10.05.2024).
8. NumPy Documentation: [сайт] – URL: <https://numpy.org/doc/> (дата обращения: 10.05.2024).
9. ONNX Docs: [сайт] – URL: <https://onnxruntime.ai/docs/api/> (дата обращения: 10.05.2024).
10. OpenCV Documentation: [сайт] – URL: <https://opencv.org/get-started/> (дата обращения: 10.05.2024).
11. Pexels: [сайт] – URL: <https://www.pexels.com/> (дата обращения: 10.05.2024).
12. Pillow Documentation: [сайт] – URL: <https://pillow.readthedocs.io/en/stable/index.html/> (дата обращения: 10.05.2024).
13. Python 3.12.3 Documentation: [сайт] – URL: <https://cyberleninka.ru/article/n/model-sistemy-raspoznavaniya-nazemnyh-obektov-na-osnove-neyronnoy-seti> (дата обращения: 10.05.2024).
14. Ultralytics Docs: [сайт] – URL: <https://docs.ultralytics.com/> (дата обращения: 10.05.2024).
15. Vis Drone Dataset: [сайт] – URL: <https://github.com/VisDrone/VisDrone-Dataset/> (дата обращения: 10.05.2024).
16. Документация по Avalonia: [сайт] – URL: <https://docs.avaloniaui.net/ru/docs/welcome/> (дата обращения: 10.05.2024).
17. Дроны-спасатели: как ИИ помогает находить потерявшихся людей в лесу: [сайт] – URL: <https://ai.gov.ru/mediacenter/drony-spasateli-kak-ii-pomogaet-nakhodit-poteryavshikhsya-lyudey-v-lesu/> (дата обращения: 10.05.2024).
18. Как билайн 10 лет помогает поисково-спасательному отряду «ЛизаАлерт» искать людей: [сайт] – URL: <https://beelinenow.ru/articles/kak-bilayn-10-let-pomogaet-poiskovo-spasatelnomu-otryadu-lizaalert-iskat-lyudey/?ysclid=1welvz7y9o237197887> (дата обращения: 10.05.2024).
19. Поиск пропавших людей на снимках лесного массива, полученных с помощью БПЛА или ещё один разбор задачи Цифрового Прорыва: [сайт] – URL: <https://habr.com/ru/articles/699076/> (дата обращения: 10.05.2024).
20. Проект Lactus: как компьютерное зрение помогает спасти потерявшихся людей: [сайт] – URL: <https://habr.com/ru/companies/ods/articles/483616/> (дата обращения: 10.05.2024).
21. Сотрудничество Билайн и «ЛизаАлерт»: [сайт] – URL: <https://lizaalert.org/19835-2/?ysclid=1welwkxeuw486987250> (дата обращения: 10.05.2024).
22. Сравнение производительности UI в WPF, Qt, WinForms и FLTK: [сайт] – 2024. – URL: <https://habr.com/ru/articles/263897/> (дата обращения: 10.05.2024).

© В. О. Архипов, А. А. Шаранов, 2024