

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ, РЕЧИ, ТЕКСТА И РАСПОЗНАВАНИЕ ОБРАЗОВ

SIGNAL, IMAGE, SPEECH, TEXT PROCESSING AND PATTERN RECOGNITION



УДК 004
DOI: 10.37661/1816-0301-2025-22-1-66-72

Оригинальная статья
Original Article

Сравнительный анализ алгоритмов отслеживания объектов

Б. А. Залесский[✉], В. А. Иванюкович, К. В. Реп, Д. А. Старикович

Объединенный институт проблем информатики
Национальной академии наук Беларусь,
ул. Сурганова, 6, Минск, 220012, Беларусь
[✉]E-mail: zalesky@newman.bas-net.by

Аннотация

Цели. Представляются результаты вычисления и сравнительного анализа характеристик нейросетевого алгоритма отслеживания объекта (трекера), предложенного авторами в работе [1], с двумя другими алгоритмами при решении актуальной задачи автоматического обнаружения и дальнейшего сопровождения дронов. Для сравнительного анализа были выбраны: ByteTrack – один из лучших в настоящее время среди трекеров открытого доступа и простой трекер, основанный на использовании нейросетевого детектора и корреляции вместе с фильтром Калмана. Первый из трекеров был выбран потому, что он может быть реализован на языке C++ без применения сторонних библиотек и фреймворков и использован на малых вычислителях в режиме реального времени. Второй трекер тестировался для выяснения того, насколько новые трекеры лучше простых хорошо известных. Особенность используемых алгоритмов заключается в автоматическом обнаружении и захвате дрона, его дальнейшем надежном сопровождении, быстрым повторном захвате в случае срыва сопровождения, захвате другого дрона при исчезновении сопровождаемого объекта. В анализируемых трекерах обнаружение дрона на кадрах видео осуществляется с помощью нейронной сети-детектора, а сопровождение – с помощью нейронной сети-детектора и разработанных алгоритмов.

Методы. Для проведения сравнительного анализа алгоритмов отслеживания объекта были созданы и размечены два датасета. Они представляют собой видео, на кадрах которых присутствуют дроны разных типов. Первый из датасетов, содержащий 36 895 кадров, использовался для обучения алгоритмов, а второй, состоящий из 8678 кадров, – для вычисления характеристик алгоритмов и выполнения сравнительного анализа. Видео обучающего и тестового датасетов сняты разными камерами в различных условиях. Для обучения нейросетевой части трекеров были написаны версии алгоритмов на языке программирования Python, а для вычисления и анализа характеристик в условиях, близких к реальным, – на языке C++, что потребовало конвертации обученной сети с помощью фреймворка TensorRT. Также были реализованы программные средства сбора и обработки экспериментальных данных.

Результаты. Проведенный сравнительный анализ трех алгоритмов отслеживания объекта позволил вычислить и сравнить их характеристики, а также сделать выводы о способе обучения использованной

нейронной сети-детектора; о возможности применения трекеров в режиме реального времени на бюджетных персональных компьютерах с бюджетными видеокартами, имеющими программно-аппаратную архитектуру CUDA, и о применимости двух из них для решения задачи практического отслеживания дронов, наблюдаемых видеокамерами, с достаточной точностью и надежностью. Из трех протестированных алгоритмов наилучшие характеристики имеет разработанный авторами.

Заключение. Проведенный сравнительный анализ трекеров показал возможность практического применения трекера и алгоритма ByteTrack для решения задачи отслеживания дронов, однако в настоящее время сохраняется проблема обнаружения малоразмерных беспилотных летательных аппаратов.

Ключевые слова: видео, детекторы объектов, нейронные сети, алгоритмы отслеживания, дроны, характеристики трекеров, сравнительный анализ

Для цитирования. Сравнительный анализ алгоритмов отслеживания объектов / Б. А. Залесский, В. А. Иванюкович, К. В. Рир, Д. А. Старикович // Информатика. – 2025. – Т. 22, № 1. – С. 66–72. – DOI: 10.37661/1816-0301-2025-22-1-66-72.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 05.02.2025

Подписана в печать | Accepted 21.02.2025

Опубликована | Published 31.03.2025

Comparative analysis of object tracking algorithms

Boris A. Zalesky[✉], Vladimir A. Ivanyukovich, Kiril V. Reer, Danila A. Starikovich

*The United Institute of Informatics Problems
of the National Academy of Sciences of Belarus,
st. Surganova, 6, Minsk, 220012, Belarus*

[✉]E-mail: zalesky@newman.bas-net.by

Abstract

Objectives. The article presents the results of calculation and comparative analysis of the characteristics of the algorithm proposed by the authors in [1] for tracking an object captured by a video camera, when solving the urgent task of automatic detection and tracking of drones. Two algorithms were selected for comparative analysis, one of which is the currently known open source ByteTrack tracker, and the other is a simple tracker based on the use of the neural network, correlation comparison together with Kalman filter. The first tracker was chosen because it can be implemented in C++ without using third-party libraries and frameworks and used on small computers in real time. The second tracker was used to determine how much better new trackers are than simple, long-used ones. The specificity of the used algorithms is automatic detection and capture of the drone, its further reliable tracking, quick repeated capture in case of tracking failure, capture of another drone when the tracked object disappears. In the used trackers, drone detection in video frames is carried out using a neural network detector, and tracking is done with the help of the neural network detector and developed tracking algorithms.

Methods. To perform a comparative analysis of object tracking algorithms, two datasets consisting of video frames that contain drone images were created and labeled. The training dataset consists of 36895 frames whereas testing one contains 8678 images. The videos of the training and test datasets were shot with different cameras in different conditions. To train the neural network part of the trackers, versions of the algorithms were written in the Python programming language, and to calculate and analyze characteristics in conditions close to real ones, in C++, which required converting the trained network using the TensorRT framework. Software tools for gathering and processing experimental data were also implemented.

Results. The comparative analysis of three object tracking algorithms allowed us to calculate and compare the characteristics of these trackers, as well as draw conclusions about the method of training the used neural network detector; about the possibility of using trackers in real time on budget personal computers with budget

video cards that have the CUDA software and hardware architecture, about the applicability of two of them for solving the problem of practical tracking of drones observed by video cameras with sufficient accuracy and reliability. Of the three tested algorithms the tracker previously developed by the authors has the best characteristics. Conclusion. The comparative analysis of the above-mentioned trackers showed the possibility of practical application of the tracker and the ByteTrack algorithm for solving the problem of tracking drones, however, there is still a problem with detecting small-sized unmanned aerial vehicles.

Keywords: video, object detectors, neural networks, tracking algorithms, drones, tracker characteristics, comparative analysis

For citation. Zalesky B. A., Ivanyukovich V. A., Reer K. V., Starikovich D. A. *Comparative analysis of object tracking algorithms*. Informatika [Informatics], 2025, vol. 22, no. 1, pp. 66–72 (In Russ.). DOI: 10.37661/1816-0301-2025-22-1-66-72.

Conflict of interest. The authors declare of no conflict of interest.

Введение. Задача автоматического обнаружения объекта заданного типа, наблюдаемого видеокамерой, и дальнейшего его сопровождения находит применение в широком спектре приложений, таких как мониторинг безопасности движения транспортных средств, медицинская диагностика, навигация роботов и автономного транспорта, оценка позы и походки человека, анализ активности диких животных, пешеходов и т. д. В подавляющем большинстве случаев наблюдаются движущиеся объекты движущейся видеокамерой. В настоящее время по данной тематике опубликовано значительное число работ, ссылки на которые можно найти, например, в работах¹ [2, 3], доступно множество программных средств^{2,3}, в том числе экспериментальных. Часть опубликованных программных средств успешно применяется на практике. Особенностью большинства опубликованных алгоритмов и разработанных программных средств является их использование для трекинга нейронных сетей.

В настоящей работе приводятся результаты сравнительного анализа характеристик нейросетевого алгоритма отслеживания трекера [1], который для краткости будем называть OFT (Optical Flow Tracker), и двух других алгоритмов при решении актуальной задачи автоматического обнаружения и дальнейшего сопровождения дронов. Задача отслеживания объекта в данном случае понимается следующим образом: алгоритм автоматически обнаруживает и локализует положения дронов на кадрах видео, выбирает по заранее заданному критерию (например, по уровню доверия (confidence) обнаруженных объектов, а также по размеру изображения объекта и (или) его расположению на кадре видео) один из них и выполняет его сопровождение. При срыве сопровождения, вызванного исчезновением объекта с кадра или другими причинами, алгоритм перезапускается для захвата и сопровождения нового объекта заданного типа.

Одной из целей работы является сравнение характеристик нейросетевого трекера OFT с одними из лучших в настоящее время трекеров DeepSort и ByteTrack, имеющих открытые программные реализации, которые используются для решения практических задач отслеживания объектов. Эксперименты проводились на тестовых наборах видео (датасетах) дронов разных типов. Для тестирования были выбраны видео, характеристики которых отличаются от характеристик обучающих видео. Помимо разработанного алгоритма и трекеров DeepSort и ByteTrack тестировался относительно простой алгоритм, основанный на использовании нейросетевого детектора Yolov10 и корреляционного поиска (вместе с фильтром Калмана) на текущем кадре объекта, найденного на предыдущем кадре. Проведенное сравнение позволило сделать оценку, на сколько сложные современные алгоритмы лучше прежних простых при отслеживании дронов.

Заметим также, что алгоритмы отслеживания могут применяться в качестве детекторов. Использование результатов отслеживания для детекции в большинстве случаев ощутимо повыша-

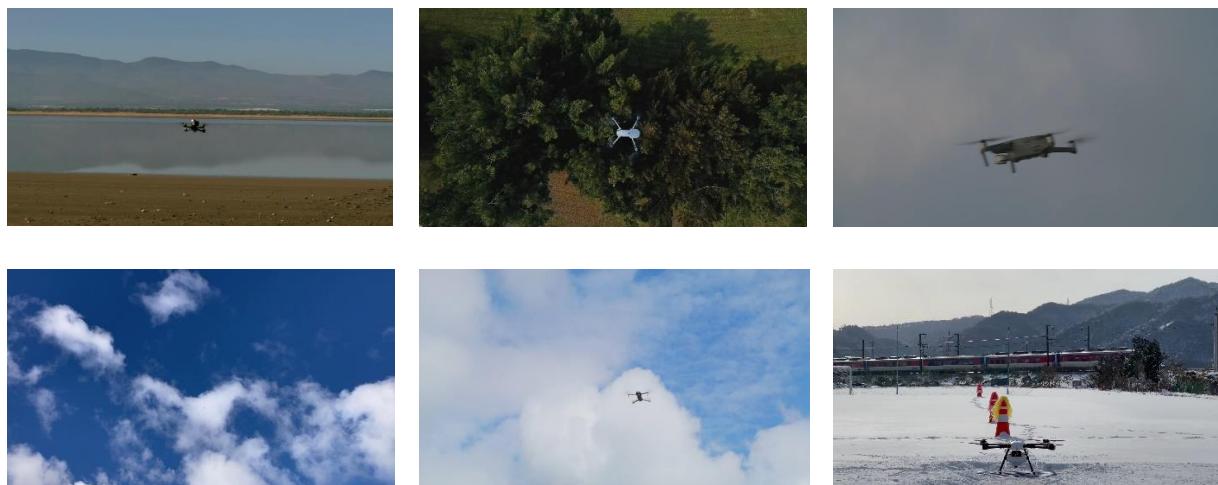
¹Object Tracking. – URL: <https://paperswithcode.com/task/object-tracking> (date of access: 24.01.2025).

²PyTorch. – URL: <https://pytorch.org/vision/stable/index.html> (date of access: 05.01.2024).

³Ultralytics YOLO Docs. – URL: <https://docs.ultralytics.com/ru/models/yolo11/> (date of access: 05.01.2024).

ет ее точность и надежность. В статье приводятся результаты сравнения обнаружения объекта детектором и разработанным трекером.

Сравнительный анализ точности и надежности трекеров. Сравнительный анализ точности и надежности трекеров проводился на датасете, сформированном, аугментированном и размеченном авторами, так как не удалось найти в открытом доступе датасеты, содержащие дроны. При этом аугментировался только обучающий датасет, в котором количество кадров было доведено до 36 895. Тестовый набор данных состоял из 8678 кадров. Особенность обоих датасетов – наличие в них большого числа изображений дронов малых размеров. На рисунке показаны кадры из созданного датасета.



Примеры кадров из тестового датасета, содержащих дрон
Examples of frames from the test dataset containing a drone

На начальном этапе сравнительного анализа было установлено, что, несмотря на опубликованные результаты сравнения трекеров DeepSort и ByteTrack, в которых доказывалось превосходство DeepSort над ByteTrack, в экспериментах на созданных авторами датасетах ByteTrack показал лучшие результаты, поэтому в дальнейшем только он был использован для сравнительного анализа.

Ранее было проведено тестирование трекера OFT, реализованного на языке Python. Результаты тестирования приведены в работе [1], где для сравнительного анализа использовалась версия трекера ByteTrack, реализованная на Python, и двух трекеров – OFT и простого корреляционного, реализованных на C++. Для этого нейросетевой детектор Yolov10, реализованный на Python, был обучен и затем конвертирован с помощью фреймворка TensorRT для применения в C++-приложении. Важно отметить, что такая конвертация снизила значения характеристик Accuracy и Recall на тестовых видео на 1–3 % (эффект снижения характеристик нейросетевых детекторов объектов при конвертации наблюдался и на других видео).

При проведении сравнительного анализа оказалось, что нейросетевой детектор Yolov10 недостаточно хорошо обучается на созданном авторами и аугментированном датасете, поэтому в статье приводятся результаты, полученные после обучения детектора только на созданном авторами датасете, и результаты, полученные после дообучения детектора, предварительно обученного на датасете COCO, не содержащем изображения дронов.

Тестовый датасет был образован кадрами девяти видео, снятых в разных условиях. На видео были запечатлены квадрокоптеры на разном удалении от камеры. В табл. 1–3 приведены средние результаты, полученные на всем тестовом датасете для обученного в разной степени детектора Yolov10.

Во втором столбце таблиц даны результаты обнаружения объектов обученным авторами детектором Yolov10 (в процентах). Числовые данные в остальных столбцах показывают, на

сколько процентов результат применения каждого трекера улучшает соответствующий показатель детектора. В табл. 1 приведены результаты использования детектора Yolov10, обученного только на созданном авторами датасете.

Таблица 1

Результаты тестирования трекеров, обученных только на датасете, созданном авторами

Table 1

Test results of trackers trained only on the dataset created by the authors

| Характеристики <i>Characteristics</i> | Детектор Yolov10, % <i>Detector Yolov10, %</i> | Трекер OFT, % <i>OFT Tracker, %</i> | Трекер корреляционный, % <i>Correlation tracker, %</i> |
|--|--|--|--|
| Accuracy | 36,55 | +12,29 | +12,18 |
| Recall | 38,72 | +12,51 | +12,36 |

В табл. 2 приведены результаты, полученные при использовании детектора Yolov10, предварительно обученного на датасете COCO и дообученного на созданном датасете.

Таблица 2

Результаты тестирования трекеров, обученных на COCO и после этого дообученных

Table 2

Test results of trackers trained on COCO and after retraining

| Характеристики <i>Characteristics</i> | Детектор Yolov10, % <i>Detector Yolov10, %</i> | Трекер OFT, % <i>OFT Tracker, %</i> | Трекер корреляционный, % <i>Correlation tracker, %</i> |
|--|--|--|--|
| Accuracy | 52,37 | +6,73 | +4,14 |
| Recall | 53,99 | +7,49 | +6,48 |

Результаты, полученные для ByteTrack, приводятся отдельно, так как была использована его программная реализация на языке Python в отличие от программных реализаций трекера OFT и корреляционного трекера, написанных на языке C++ (напомним, что при конвертации детектора Yolov10 с языка Python на язык C++ его характеристики ухудшаются на 1–3 %).

Таблица 3

Результаты тестирования ByteTrack, реализованного на Python

Table 3

Test results of the ByteTrack implemented in Python

| Характеристики <i>Characteristics</i> | ByteTrack, созданный датасет <i>ByteTrack, created dataset</i> | ByteTrack, COCO, созданный датасет <i>ByteTrack, COCO, created dataset</i> |
|--|--|--|
| Accuracy | +7,03 | +7,63 |
| Recall | +7,53 | +8,37 |

Результаты, приведенные во втором столбце табл. 3, необходимо сравнивать с результатами третьего и четвертого столбцов табл. 1, а результаты третьего столбца табл. 3 – с результатами третьего и четвертого столбцов табл. 2. Полученные данные позволяют сделать вывод о том, что при использовании трекеров с нейросетевым детектором Yolov10, обученным только на датасете, созданном авторами, трекер OFT превосходит ByteTrack по точности (Accuracy) на 5,26 %, а по чувствительности (Recall) – на 4,98 %.

При использовании трекеров с нейросетевым детектором Yolov10, обученным сначала на COCO, а затем дообученным на датасете, созданном авторами, OFT и ByteTrack показали примерно одинаковые результаты (в данном случае следует принять во внимание потерю точности ByteTrack при конвертации его программной реализации с языка Python на язык C++).

Следует заметить, что все использованные в статье трекеры работают в режиме реального времени, однако самым быстрым из них оказался трекер OFT.

Заключение. Проведенный сравнительный анализ трекеров показал возможность практического применения трекеров OFT и ByteTrack для решения задачи отслеживания дронов в режиме реального времени. Оба трекера обладают точностью и чувствительностью, достаточными для отслеживания летящих дронов. При использовании трекеров с нейросетевым детектором Yolov10, обученным на датасете, который содержит только изображения дронов, предложенный авторами алгоритм превзошел ByteTrack по точности на 5 % и по чувствительности приблизительно на 4 %. Однако при применении трекеров вместе с детектором Yolov10, обученным сначала на большом датасете COCO, а затем дообученным на меньшем, который содержит дроны, характеристики точности и чувствительности трекеров оказались одинаковыми.

Между тем следует заметить, что точность и чувствительность всех трех детекторов ощущимо падают при отслеживании дронов, изображения которых имеют слишком малый размер (меньший чем 40×40 пикселов).

Вклад авторов. *Б. А. Залесский* сформулировал задачу, разработал трекер совместно с *В. В. Иванюковичем*, подготовил текст статьи. *В. А. Иванюкович* реализовал и обучил OFT, создал один из двух использованных датасетов и протестировал на нем OFT и ByteTrack. *К. В. Реп* реализовал корреляционный трекер и выполнил тестирование всех использованных трекеров. *Д. А. Старикиевич* создал второй использованный датасет, реализовал OFT и корреляционный трекер на C++.

Список использованных источников

1. Залесский, Б. А. Алгоритм сопровождения объекта, наблюдаемого видеокамерой / Б. А. Залесский, В. А. Иванюкович // Доклады Национальной академии наук Беларуси. – 2024. – Т. 68, № 2. – С. 105–111.
2. Proc. of Conf. on Computer Vision and Pattern Recognition 2024 (CVPR 2024). – Seattle, 17–21 June 2024. – URL: <https://cvpr.thecvf.com/Conferences/2024> (date of access: 24.01.2025).
3. Proc. of Conf. on Computer Vision and Pattern Recognition 2023 (CVPR 2023). – Vancouver, 18–22 June 2023. – URL: <https://openaccess.thecvf.com/CVPR2023> (date of access: 24.01.2025).

References

1. Zalesky B. A., Ivanyukovich V. A. *Algorithm for tracking an object observed by a video camera*. Doklady Naciona'lnoj akademii nauk Belarusi [Doklady of the National Academy of Sciences of Belarus], 2024, vol. 68, no. 2, pp. 105–111 (In Russ.).
2. *Proceedings of Conference on Computer Vision and Pattern Recognition 2024 (CVPR 2024)*. Seattle, 17–21 June 2024. Available at: <https://cvpr.thecvf.com/Conferences/2024> (accessed 24.01.2025).
3. *Proceedings of Conference on Computer Vision and Pattern Recognition 2023 (CVPR 2023)*. Vancouver, 18–22 June 2023. Available at: <https://openaccess.thecvf.com/CVPR2023> (accessed 24.01.2025).

Информация об авторах

Залесский Борис Андреевич, доктор физико-математических наук, Объединенный институт проблем информатики Национальной академии наук Беларусь. E-mail: zalesky@newman.bas-net.by

Information about the authors

Boris A. Zalesky, D. Sc. (Eng.), The United Institute of Informatics Problems of the National Academy of Sciences of Belarus. E-mail: zalesky@newman.bas-net.by

Иванюкович Владимир Александрович, стажер младшего научного сотрудника, Объединенный институт проблем информатики Национальной академии наук Беларуси.

Реер Кирилл Васильевич, стажер младшего научного сотрудника, Объединенный институт проблем информатики Национальной академии наук Беларуси.

Старикович Данила Александрович, стажер младшего научного сотрудника, Объединенный институт проблем информатики Национальной академии наук Беларуси.

Vladimir A. Ivanyukovich, Trainee of Junior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

Kirill V. Reer, Trainee of Junior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.

Danila A. Starikovich, Trainee of Junior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.