https://doi.org/10.31891/2307-5732-2025-357-5
УДК 004.8

ВЕРГУН КОСТЯНТИН
Західноукраїнський Національний Університет
e-mail: k.verhun@gmail.com
ДИВАК МИКОЛА
Західноукраїнський Національний Університет
e-mail: mdy@wunu.edu.ua

ВИБІР ТА ВИПРОБУВАННЯ МЕТОДУ ВІЗУАЛЬНОГО ВІДСТЕЖЕННЯ ОБ'ЄКТІВ ДЛЯ ВБУДОВАНИХ СИСТЕМ

У роботі розглянуто задачу візуального відстеження об'єктів для застосування у вбудованих системах з обмеженими обчислювальними ресурсами. При цьому складність задачі полягає у балансуванні між якістю результатів роботи алгоритму та його обчислювальною ефективністю. На основі аналізу підходів та методів візуального відстеження об'єктів було обрано метод Nanotrack для подальшого дослідження. Проведено ряд експериментів на відкритих наборах даних для аналізу якості роботи алгоритму у різноманітних складних ситуаціях, а також для визначення швидкодії роботи алгоритму на мікроком'ютері Raspberry Pi. Проведено числову оцінку якості роботи алгоритму на відкритому наборі даних LaSOT для порівняння з альтернативними підходами та визначення перспектив покращення якості роботи методу.

***Ключові слова**: штучні нейронні мережі, візуальне відстеження об'єктів, обробка зображень.*

KOSTIANTYN VERHUN, MYKOLA DYVAK
West Ukrainian National University

SELECTION AND TESTING OF A VISUAL OBJECT TRACKING METHOD FOR EMBEDDED SYSTEMS

This paper considers a task of visual object tracking with application in embedded systems with limited computational power. The complexity of the task lies in finding the proper balance between algorithm result quality and its runtime efficiency. To select the tracking method to be used in embedded systems multiple machine learning approaches were analyzed: convolutional neural network-based methods, recurrent neural network-based methods, one-shot learning-based methods, siamese neural networks and transformers. The method nanotrack was selected for further research and analysis. To perform runtime performance and algorithm accuracy evaluation the test environment was developed using C++ programming language which was deployed on Raspberry Pi hardware. Performance analysis showed the method is suitable for real time object tracking on Raspberry Pi 5 as it allows to process more than 25 frames per second, additionally leaving the room for usage of a more complex backbone model if needed. Accuracy was evaluated using LaSOT dataset. The main types of challenging situations for the tracking algorithm were identified and corresponding videos from LaSOT dataset were selected. Based on visually observing tracking results on these videos, the conclusions about overall sufficient adaptive qualities of the algorithm were made. To evaluate overall algorithm quality the accuracy metric was calculated on LaSOT dataset and compared to other state-of-the-art tracking methods. Accuracy metrics were computed on LaSOT dataset, showing good performance for some object classes and worse scores for others. Based on the analysis, it is expected that training neural network on a data representing limited target set of object classes will lead to satisfactory accuracy for these classes while preserving original runtime performance. The further research directions are experimenting with different backbone models for the Nanotrack method and evaluating their influence on accuracy and runtime performance.

***Keywords:** artificial neural networks, visual object tracking, image processing.*

Стаття надійшла до редакції / Received 30.07.2025
Прийнята до друку / Accepted 17.08.2025

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Задача візуального відстеження об'єктів у відеопотоці має широке застосування у різноманітних сферах: автономне керування, охоронні системи, контроль дорожнього руху, тощо. Залежно від форми реалізації системи, алгоритм відстеження об'єктів може виконуватись на сервері, або ж на вбудованих платформах, що є частиною апаратного рішення, наприклад, такого, як безпілотний літальний апарат (БПЛА). Виконання таких алгоритмів безпосередньо на вбудованих платформах дає можливість побудувати незалежний пристрій, що матиме можливість супроводжувати об'єкт та відслідковувати його у відеопотоці з камери, що встановлена на цьому ж пристрої, у режимі реального часу.

У випадку реалізації вбудованої системи з можливостями візуального відстеження об'єктів у відеопотоці, вимоги до швидкодії та енергоефективності алгоритмів суттєво зростають, у порівнянні з ситуацією, коли алгоритм виконується на сервері. Вимоги до обчислювальної складності алгоритму залежать від конкретної апаратної платформи, що використовується. У випадку БПЛА, поширеним є використання мікроком'ютерів на базі Raspberry Pi, що не мають окремого графічного прискорювача.

У цій роботі розглядається проблема вибору методу візуального супроводження об'єкта для використання на платформі Raspberry Pi у режимі реального часу, а також проведення тестування обраного методу на цій платформі для підтвердження якості роботи методу та його швидкодії.

Аналіз досліджень та публікацій

Серед методів відстеження об'єкта у відеопотоці виділяють такі головні класи: методи на основі застосування кореляційних фільтрів та методи на основі застосування штучних нейронних мереж. У випадку відносно простої ситуації, навіть базові алгоритми на базі кореляційних фільтрів дають результати прийнятної якості. Однак поява будь-якого ускладнення висуває додаткові вимоги до алгоритмів. Серед таких ускладнень можна виділити [1]: перекриття об'єкта у полі зору, зміна масштабу об'єкта, пов'язана з рухом самого об'єкта або камери, зміна ракурсу огляду об'єкта, зміни у освітленні, пов'язані як з погодніми умовами, так і зі штучним світлом чи неідеальним спрацюванням автоекспозиції камери, зменшення частоти кадрів відеопотоку, пов'язане з завадами або обмеженням обчислювальної спроможності системи, розмивання об'єкта при швидкому русі.

При спробі побудувати алгоритм таким чином, щоб забезпечити стійкість до усіх або максимально можливої кількості вищеперелічених вимог виявляється, що представлення об'єкта у відеопотоці може зазнавати різнотипних змін у широкому їх діапазоні таким чином, що застосування кореляційного фільтра безпосередньо на зображенні не дасть можливості встановити відповідність.

Таким чином, для побудови більш універсального методу відстеження об'єкта перспективним є застосування методів на основі штучних нейронних мереж, що дає змогу перекласти відповідальність за встановлення ключових особливостей об'єкта на нейронну мережу.

Методи на основі штучних нейронних мереж були зазнали широкого розвитку протягом останніх років, що пов'язано зі збільшення доступності обчислювальних потужностей, необхідних як для їх тренування, так і для їх запуску [2–5]. Головними напрямками розвитку цих методів є збільшення розміру структури самої нейронної мережі, та застосування нових підходів та ідей, що дасть можливість врахувати новий тип зв'язків та інформації.

Методи на основі штучних мереж можна поділити на класи залежно від ключової ідеї та підходу, що лежать в основі методу. Далі більш детально розглянемо кожен із таких класів.

Методи на базі згорткових нейронних мереж. Ідея цих методів базується на використанні повнозв'язних згорткових нейронних мереж, що дають змогу виявляти ключові особливості цільового об'єкта безпосередньо з зображення. При цьому результатом виконання нейронної мережі є не безпосередньо координати цільового об'єкта, а лише аналіз вхідного зображення у вигляді confidence map або подібних. Далі результат роботи нейронної мережі аналізується на предмет вибору найбільш ймовірної позиції цільового об'єкта. Певним чином ці методи можна вважати узагальненням більш простих кореляційних методів в напрямку використання більш гнучких та стійких особливостей об'єкта для виконання трекінгу. Прикладами методів є [6,7].

Методи на базі рекурентних нейронних мереж. Задача відстеження об'єкта базується на аналізі відеопослідовності, у якій кадри пов'язані між собою. Врахування результату відстеження на попередньому кроці (кадрі) є важливою інформацією для розв'язку задачі на поточному кроці (кадрі). Таким чином, логічним є використання механізму рекурентних нейронних мереж, що дає можливість врахувати попередній результат завдяки застосуванню технік LSTM (Long Short-Term Memory) [8] та GRU (Gated Recurrent Unit) [9]. Рекурентні нейронні мережі мають можливість враховувати часові властивості процесу руху і видавати позицію об'єкта як результат роботи нейронної мережі. Також методи, що базуються на рекурентних нейронних мережах, дають можливість оцінити такі властивості цільового об'єкта, як напрямок руху та швидкість, а також зробити оцінку непевності отриманого результату. Прикладами методів, що базуються на рекурентних нейронних мережах є [10, 11].

Методи з донавчанням на одному чи кількох кадрах. Методи цього класу розроблені для швидкої адаптації натренованої узагальненої нейронної мережі до потрібного об'єкта. При цьому цільових об'єкт представленний лише на одному чи кількох кадрах, що унеможливлює повноцінне тренування нейронної мережі за допомогою класичного підходу з великою кількістю даних. При цьому деякі методи модифікують лише ваги нейронної мережі у процесі донавчання [12, 13], в той час як деякі реалізують техніку пошуку архітектури нейронної мережі, таких як NAS [14]. Прикладом реалізації такого підходу є метод LightTrack [15], що пропонує систему для пошуку оптимальної архітектури нейронної мережі і дає результати з хорошим поєднанням точності і швидкодії.

Методи на базі сіамських нейронних мереж. Ідея сіамських нейронних мереж полягає у використанні двох однакових гілок згорткової нейронної мережі для порівняння цільового шаблону з областями зображення, що можуть містити цільовий об'єкт. За рахунок тренування саме в контексті виявлення відповідності, такі нейронні мережі вивчають надійні метрики подібності. Нейронна мережа SiamFC [16] використовує дві ідентичні згорткові нейронні мережі для генерування внутрішнього представлення шаблону і регіону пошуку. Далі на основі внутрішнього представлення будується мапа подібностей для регіону пошуку, з якої робиться висновок про позицію об'єкта. Мережа SiamBAN [17] додає додає адаптивну регресію для обмежувального прямокутника, що дає можливість уточнити позицію об'єкта, порівняно зі звичайною сіамською нейронною мережею. Більш детальний огляд застосування сіамських нейронних мереж надано в оглядовій статті [18].

Методи на базі трансформерів. Ідея архітектури трансформерів, що базується на механізмі attention, хоч і виникла в сфері обробки природної мови, була також застосована для задач візуального відстеження об'єктів [19]. Механізм attention дає можливість пов'язувати цільовий об'єкт з його

контекстом, встановлюючи складні взаємозв'язки об'єкта з його оточенням. Це покращує точність відстеження, особливо у випадках зі складною зміною геометрії об'єкта та його перекриття. Розвитком ідеї застосування трансформерів для відстеження об'єктів є метод TransT++ [20]. Застосування таких методів зазвичай вимагає суттєвих обчислювальних ресурсів.

Формулювання цілей статті

Метою роботи є:

- Вибір та обґрунтування методу візуального відстеження об'єктів у режимі реального часу для використання на вбудованих системах;
- Проведення тестування обраного методу на предмет якості результатів та швидкодії для підтвердження очікуваних характеристик, а також вибору оптимальної конфігурації методу.

Виклад основного матеріалу

Вибір методу для вбудованих платформ. При виборі методу для застосування в умовах необмеженої обчислювальної потужності (еквівалентної сучасному персональному комп'ютеру) найчастіше, достатнім буде покластись на вже обчислену точність застосування методу на відомих наборах даних, зокрема VOT Challenge [21, 22]. Однак у випадку вибору методу для вбудованої платформу Raspberry Pi, яка окрім обмеженої потужності також не має окремого графічного прискорювача, потрібно першочергово зважати на можливість запуску методу в умовах багатоядерного процесора загального призначення (CPU), а також наявність оптимізованої реалізації цього методу. При цьому, на даному етапі, достатньо буде переконатись, що якість роботи методу є прийнятною, тобто за метриками точності несуттєво поступається більш потужним методам, які, в свою чергу, вимагають суттєво більших обчислювальних ресурсів.

Метод Nanotrack [23] було розроблено як розвиток методу LightTrack [15], поєднуючи її з SiamBAN [17]. У результаті, було зменшено розмір нейронної мережі, отриманої з підходом LightTrack, а також отримати переваги характерні для методів, що ґрунтуються на застосуванні сіамських нейронних мереж.

Реалізація рішення на цільовій платформі. Для розгортання та тестування алгоритму на платформі Raspberry Pi було розроблено тестове середовище на основі OpenCV. Починаючи з версії OpenCV 4.8 [24], метод nanotrack є доступним у складі цієї бібліотеки, що дає можливість інтегрувати цей метод в оптимізоване програмне забезпечення розроблене за допомогою мови C++.

Розроблене тестове середовище дає можливість завантажити відео з файлу, обрати цільовий об'єкт на першому або будь-якому іншому кадрі, та візуально спостерігати за результатом роботи алгоритму відстеження на подальших кадрах, а також слідкувати за швидкодією. Головним показником швидкодії при цьому є тривалість обробки алгоритмом одного кадру з відеопотоку.

Операційна система для Raspberry Pi на базі Debian Linux надає застарілу версію бібліотеки OpenCV, тому для отримання доступу до алгоритму nanotrack необхідно провести збірку цієї бібліотеки з вихідного коду. В експериментах використовувалась версія бібліотеки OpenCV 4.11.

Аналіз швидкодії реалізації алгоритму. Для аналізу швидкодії алгоритму було проведено ряд експериментів з використанням платформ Raspberry Pi 5, а також Raspberry Pi Zero 2W. У якості даних для проведення експериментів було використано відеофайли з набору даних LaSOT [25]. Було виміряно середню тривалість обробки одного кадру з відео. У результаті, на платформі Raspberry Pi 5 середня тривалість оброки одного кадру становила 15 мс, а на платформі Raspberry Pi Zero 2W – 55 мс.

На основі цих вимірювань можна зробити висновок, що швидкодія алгоритму і його реалізації є достатньою для обробки відео у реальному часі (25 кадрів на секунду) на платформі Raspberry Pi 5, при цьому залишається деякий простір для використання більш складної моделі у разі необхідності покращення точності. Для платформи Raspberry Pi Zero 2W максимальна кількість оброблених кадрів за секунду становить 18, що є меншим ніж необхідно для обробки всіх кадрів у відеопотоці, однак у більшості випадків повинна бути достатньою для збереження точності відстеження.

Аналіз якості роботи алгоритму. Для аналізу якості роботи алгоритму було підібрано ряд відео з набору даних LaSOT [25]. Підбір відео для тестування здійснювався таким чином, щоб вибрані відео покривали різної природи складні ситуації для алгоритму відстеження об'єктів. Далі наведено результати застосування методу на наборі відео.

Експеримент 1: перекриття об'єкта. Перекриття об'єкта часто виникає у сценах з рухом об'єктів або камери. Для ілюстрації роботи алгоритму в умовах перекриття об'єкта використано відео bicycle-6. У цьому відео зйомка відбувається з рухомої камери, а велосипедист маневрує на дорозі серед автомобілів, періодично будучи перекритим у полі зору на періоди до 20 кадрів. Не зважаючи на це, об'єкт не втрачається і продовжує бути відстеженим після повернення в поле зору камери (рис. 1).

Рис. 1. Результат відстеження об'єкта у випадку перекриття

Експеримент 2: зміна відстані та масштабу об'єкта. У рамках цього експерименту було використано відео car-6. Зйомка відео відбувається з рухомої камери, а відстань до відстежуваного автомобіля змінюється впродовж відео. Незважаючи на зміни масштабу, об'єкт не втрачається, а відслідковується протягом усього відео (рис. 2). В окремих моментах, коли відстежуваний автомобіль різко маневрує, на деякий час в прямокутник потрапляє і інший автомобіль, що розташований близько до відстежуваного. За деякий час алгоритм стабілізується і знову відстежує правильний об'єкт (рис. 3).

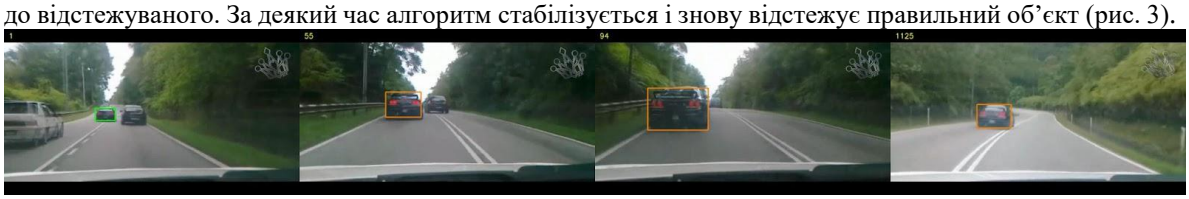

Рис. 2. Результат відстеження об'єкта у випадку перекриття (1)

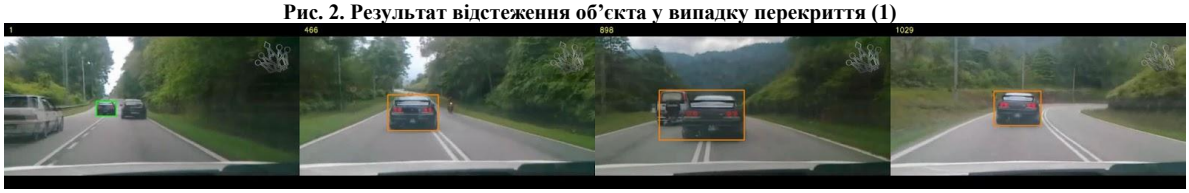

Рис. 3. Результат відстеження об'єкта у випадку перекриття (2)

Експеримент 3: зміна ракурсу огляду об'єкта. Зміна ракурсу об'єкта виникає у динамічних сценах при переміщенні об'єкта або русі камери, з якої проводиться зйомка. Для цього експерименту використано відео motorcycle-5. Кут огляду об'єкта змінюється, але відслідковування відбувається без збоїв (рис. 4).

Рис. 4. Результат відстеження об'єкта у випадку зміни ракурсу

Експеримент 4: зміни в освітленні. Зміни в освітленні виникають при переміщенні об'єкта, при зміні погодних умов, при динамічному штучному освітленні, а також при зміні експозиції камери, з якої відбувається зйомка. Для перевірки роботи алгоритму в таких умовах використано відео person-13 та person-15. У відео person-15 відбуваються зміни в штучному освітленні, що призводить до переходу від темного до освітленого середовища, а також до зміни в кольоровій гамі кадру. При цьому людина відстежується протягом усього відео.

Рис. 5. Результат відстеження об'єкта у випадку зміни освітлення: person-13

Рис. 6. Результат відстеження об'єкта у випадку зміни освітлення: person-15

Експеримент 5: наявність великої кількості однотипних об'єктів. Для перевірки роботи алгоритму в такій ситуації було використано відео person-7, що містить сцену з великою кількістю людей. Ситуація ускладнюється рухом камери та випаданням частини людей з поля зору камери, а також короткотривалим перекриттям іншими об'єктами. При цьому залежно від обрання тієї чи іншої людини для відстеження, алгоритм по-різному відпрацьовує. Наприклад, у випадку, наведеному на рисунку нижче, відстеження відбується безперебійно не зважаючи на близькість подібних об'єктів, а також тимчасове перекриття. У той же час, при виборі іншої людини, відстеження періодично збивається на інших об'єкт і вже не повертається до початково вибраного. Подібна поведінка спостерігалась також в деяких інших відео з великою кількістю об'єктів, схожих на відстежуваний.

Рис. 7. Результат відстеження у випадку наявності великої кількості об'єктів (1)

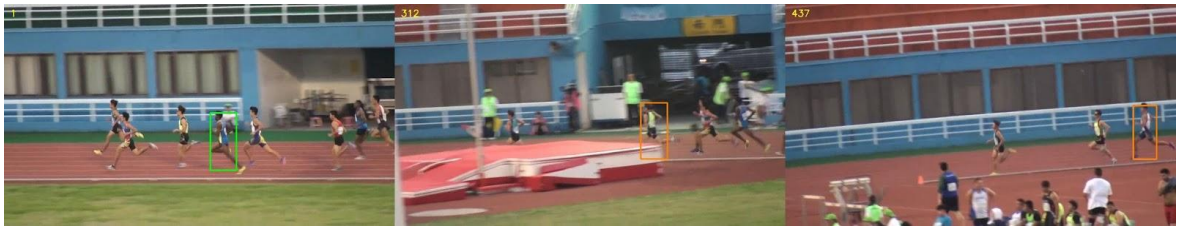

Рис. 8. відстеження у випадку наявності великої кількості об'єктів (2)

Оцінка точності на наборі даних LaSOT. Для отримання числової оцінки точності роботи алгоритму Nanotrack було обчислено метрики precision, normalized precision та AUC [25] на наборі даних LaSOT. Набір даних LaSOT містить відео з 70 категорій, кожна з яких присвячена певному типу об'єкта (літак, автомобіль, кіт, тощо). Загальна кількість відео у цьому наборі даних – 1400, середня тривалість одного відео становить приблизно 2500 кадрів. Значення обчислених метрик на усьому наборі даних: precision: 0.47, normalized precision: 0.44, AUC: 0.43. Ці значення поступаються показникам найкращих сучасних алгоритмів відстеження об'єктів [26]. Однак більш детальний аналіз показників алгоритму Nanotrack по окремих категоріях дає можливість зробити висновок, що на деяких типах об'єктів точність алгоритму не поступається іншим більш потужним. Значення метрик точності для кожної з категорій наведено у таблиці 1. Для зручності, у таблиці метрики мають позначення: precision – P, normalized precision – NP. Наприклад, для таких категорій об'єктів як «автобус», «прапор», «автомобіль» значення AUC перебувають у діапазоні 60% і більше. Найнижчу точність алгоритм показує на об'єктах таких категорій як «риба», «мікрофон», «номерний знак», «шолом», тощо. Оскільки для деяких типів об'єктів забезпечується висока точність відстеження, можна зробити припущення про можливість модифікувати ваги нейронної мережі, що використовується, для кращого представлення об'єктів з категорій, які є специфічними для передбачуваної сфери застосування алгоритму. При цьому, допускаючи зменшення загальності моделі, можна зберегти поточний рівень її швидкодії.

Таблиця 1

Значення метрик точності для алгоритму Nanotrack для окремих категорій набору даних LaSOT

Category	P	NP	AUC	**Category**	P	NP	AUC	**Category**	P	NP	AUC
bus	0,800	0,762	0,728	**spider**	0,577	0,524	0,497	**sheep**	0,512	0,366	0,397
coin	0,766	0,687	0,670	**truck**	0,594	0,529	0,492	**kangaroo**	0,396	0,390	0,395
flag	0,670	0,660	0,624	**bicycle**	0,521	0,503	0,479	**lizard**	0,240	0,367	0,388
deer	0,721	0,647	0,621	**person**	0,542	0,477	0,458	**hat**	0,504	0,376	0,379
car	0,747	0,638	0,600	**tiger**	0,478	0,468	0,456	**helmet**	0,392	0,308	0,305
crocodile	0,486	0,623	0,590	**motorcycle**	0,511	0,455	0,451	**zebra**	0,186	0,294	0,303
guitar	0,726	0,581	0,574	**dog**	0,489	0,433	0,446	**tank**	0,282	0,332	0,294
train	0,572	0,583	0,565	**total**	**0,474**	**0,437**	**0,429**	**licenseplate**	0,472	0,328	0,293
horse	0,477	0,538	0,548	**giraffe**	0,420	0,436	0,423	**racing**	0,342	0,251	0,242
pig	0,614	0,529	0,539	**gametarget**	0,518	0,435	0,415	**microphone**	0,354	0,190	0,190
cat	0,581	0,554	0,538	**surfboard**	0,696	0,423	0,401	**goldfish**	0,191	0,179	0,188

Висновки і перспективи подальших досліджень

На основі порівняння з іншими методами та на основі проведених експериментів, серед переваг методу Nanotrack можна виділити:

1. Достатній рівень узагальненості, притаманних методам на основі штучних нейронних мереж, що показано експериментами на відео з різного типу складними сценами;
2. Адаптованість для розгортання на вбудованих платформах та задовільна швидкодія на

процесорі загального призначення без застосування графічних чи інших спеціальних прискорювачів, що показано експериментами з розгортанням алгоритму на платформі Raspberry Pi;

3. Можливість адаптації моделей, що використовуються методом, шляхом дотреновування на наборах даних релевантних для специфічної предметної області застосування або використання моделей більшого розміру за умов наявності відповідних апаратних потужностей;
4. Наявність реалізації у бібліотеці OpenCV.
5. За деякими категоріями об'єктів, метрики точності не поступаються сучасним більш потужним алгоритмам відстеження.

Серед недоліків можна відзначити відставання по метрикам якості від більш загальних та точних state-of-the-art методів, що використовують архітектури нейронних мереж більшого розміру. Це, зокрема, призводить до неточностей при відстеженні на сценах з великою кількістю схожих об'єктів, а також низьку точність відстеження об'єктів окремих класів.

Оскільки метрики точності мають високі значення на одних категоріях об'єктів, і низькі для інших, доцільним є визначення очікуваної предметної області застосування алгоритму. Після такого визначення слід провести перетренування нейронної мережі на наборі даних з високим рівнем представлення об'єктів, що відповідають сфері застосування. Очікується, що при цьому точність відстеження таких об'єктів буде високою, зі збереженням поточних параметрів швидкодії алгоритму. Оскільки експерименти показали, що швидкодія алгоритму залишає простір для додаткових обчислень, доцільним також є проведення експериментів з частковою модифікацією структури нейронної мережі шляхом додавання шарів. Також важливим є дослідження поведінки алгоритму в умовах різної роздільної здатності оброблюваного відео, що дасть можливість встановити вимоги до характеристик камери, що використовується у рішенні.

References

1. Chen, F., Wang, X., Zhao, Y., Lv, S., & Niu, X. (2022). Visual object tracking: A survey. *Computer Vision and Image Understanding*, *222*, 103508. https://doi.org/10.1016/j.cviu.2022.103508
2. Mirzaei, B., Nezamabadi-pour, H., Raoof, A., & Derakhshani, R. (2023). Small Object Detection and Tracking: A Comprehensive Review. *Sensors*, *23*(15), 6887. https://doi.org/10.3390/s23156887.
3. Kadam, P., Fang, G., & Zou, J. J. (2024). Object Tracking Using Computer Vision: A Review. *Computers*, *13*(6), 136. https://doi.org/10.3390/computers13060136
4. Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., & Kim, T.-K. (2021). Multiple object tracking: A literature review. *Artificial Intelligence*, *293*, 103448. https://doi.org/10.1016/j.artint.2020.103448
5. Zhang, Y., Wang, T., Liu, K., Zhang, B., & Chen, L. (2021). Recent advances of single-object tracking methods: A brief survey. *Neurocomputing*, *455*, 1–11. https://doi.org/10.1016/j.neucom.2021.05.011
6. Nam, H., & Han, B. (2016). Learning Multi-domain Convolutional Neural Networks for Visual Tracking. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4293–4302. https://doi.org/10.1109/CVPR.2016.465
7. Wang, L., Ouyang, W., Wang, X., & Lu, H. (2015). Visual Tracking with Fully Convolutional Networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, 3119–3127. https://doi.org/10.1109/ICCV.2015.357
8. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
9. Cho, K., van Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1409.1259
10. Dequaire, J., Ondrúška, P., Rao, D., Wang, D., & Posner, I. (2018). Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *The International Journal of Robotics Research*, *37*(4–5), 492–512. https://doi.org/10.1177/0278364917710543
11. Ondruska, P., & Posner, I. (2016). Deep Tracking: Seeing Beyond Seeing Using Recurrent Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1). https://doi.org/10.1609/aaai.v30i1.10413
12. Bertinetto, L., Henriques, J. F., Valmadre, J., Torr, P. H. S., & Vedaldi, A. (2016). *Learning feed-forward one-shot learners* (No. arXiv:1606.05233). arXiv. https://doi.org/10.48550/arXiv.1606.05233
13. Dong, X., Shen, J., Wu, D., Guo, K., Jin, X., & Porikli, F. (2019). Quadruplet Network With One-Shot Learning for Fast Visual Object Tracking. *IEEE Transactions on Image Processing*, *28*(7), 3516–3527. https://doi.org/10.1109/TIP.2019.2898567
14. Zoph, B., & Le, Q. V. (2016). *Neural Architecture Search with Reinforcement Learning* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1611.01578
15. Yan, B., Peng, H., Wu, K., Wang, D., Fu, J., & Lu, H. (2021). *LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search* (Version 1). arXiv.

https://doi.org/10.48550/ARXIV.2104.14545

16. Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. S. (2021). *Fully-Convolutional Siamese Networks for Object Tracking* (No. arXiv:1606.09549). arXiv. http://arxiv.org/abs/1606.09549

17. Chen, Z., Zhong, B., Li, G., Zhang, S., & Ji, R. (2020). Siamese Box Adaptive Network for Visual Tracking. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6667–6676. https://doi.org/10.1109/CVPR42600.2020.00670

18. Javed, S., Danelljan, M., Khan, F. S., Khan, M. H., Felsberg, M., & Matas, J. (2022). Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20. https://doi.org/10.1109/TPAMI.2022.3212594

19. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., & Lu, H. (2021). Transformer Tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8122–8131. https://doi.org/10.1109/CVPR46437.2021.00803

20. Chen, X., Yan, B., Zhu, J., Lu, H., Ruan, X., & Wang, D. (2022). High-Performance Transformer Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–16. https://doi.org/10.1109/TPAMI.2022.3232535

21. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.-K., Chang, H. J., Danelljan, M., Zajc, L. Č., Lukežič, A., Drbohlav, O., Björklund, J., Zhang, Y., Zhang, Z., Yan, S., Yang, W., Cai, D., Mayer, C., Fernández, G., … Zhuang, Y. (2023). The Tenth Visual Object Tracking VOT2022 Challenge Results. In L. Karlinsky, T. Michaeli, & K. Nishino (Eds.), *Computer Vision – ECCV 2022 Workshops* (Vol. 13808, pp. 431–460). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25085-9_25

22. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.-K., Chang, H. J., Danelljan, M., Zajc, L. C., Lukezic, A., Drbohlav, O., Kapyla, J., Hager, G., Yan, S., Yang, J., Zhang, Z., Fernandez, G., Abdelpakey, M., Bhat, G., … Zhu, X.-F. (2021). The Ninth Visual Object Tracking VOT2021 Challenge Results. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2711–2738. https://doi.org/10.1109/ICCVW54120.2021.00305

23. Chu, H. (2025). *NanoTrack: Deep learning-based mobile model deployment (object tracking)* [GitHub repository]. GitHub. https://github.com/HonglinChu/NanoTrack

24. OpenCV. (2024). *cv::TrackerNano class reference*. OpenCV documentation. https://docs.opencv.org/4.x/d8/d69/classcv_1_1TrackerNano.html

25. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Harshit, Huang, M., Liu, J., Xu, Y., Liao, C., Yuan, L., & Ling, H. (2020). *LaSOT: A High-quality Large-scale Single Object Tracking Benchmark* (No. arXiv:2009.03465). arXiv. http://arxiv.org/abs/2009.03465

26. Papers with Code. (2024). *LaSOT benchmark on Papers with Code*. https://paperswithcode.com/sota/object-tracking-on-lasot