УДК 004.93:004.021 **DOI: 10.18372/2073-4751.84.20894**

А. С. Климова, к.т.н., доцент
orcid.org/0000-0002-4721-2241
asiia.klymova@npp.nau.edu.ua
А. В. Полухін, к.т.н., доцент
orcid.org/0000-0001-5915-7799
pav@kai.edu.ua
І. М. Карий
7360067@stud.kai.edu.ua

ЕКСПЕРІМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МАСШТАБОВАНИХ АРХІТЕКТУР YOLOV8(N-X) У ЗАДАЧАХ МУЛЬТИОБ'ЄКТНОГО СУПРОВОДУ НА БАЗІ АЛГОРИТМУ BYTETRACK

Державний університет «Київський авіаційний інститут»

Вступ

Стрімка урбанізація та збільшення навантаження на транспортну інфраструктуру вимагають впровадження інтелектуальних транспортних систем (ITS), здатних аналізувати потоки в режимі реального часу. У сучасних системах відеоаналітики недостатньо просто виявити об'єкти на кожному окремому кадрі. Для отримання змістовної статистики - наприклад, підрахунку унікальних автомобілів, визначення їхніх траєкторій, часу перебування в кадрі - необхідно виконувати процедуру трекінгу (tracking). Завдання трекінгу полягає у тому, щоб асоціювати детекції одного й того ж фізичного об'єкта на послідовних кадрах, присвоюючи йому унікальний ідентифікатор (ID) та підтримуючи цей ID, поки об'єкт знаходиться в полі зору камери [1].

Класичні підходи до мультиоб'єктного трекінгу (MOT) часто стикаються з проблемою втрати об'єктів під час часткового перекриття (оклюзії). Це відбувається тому, що стандартні алгоритми відсікають детекції з низькою впевненістю, вважаючи їх шумом. Сучасний алгоритм ByteTrack пропонує вирішення цієї проблеми шляхом використання дворівневої асоціації [3]. Проте ефективність роботи трекера напряму залежить від якості та швидкості базового детектора.

Мета

Метою даної роботи є визначення оптимальної архітектури нейромережі сімейства YOLOv8 для інтеграції з ByteTrack шляхом емпіричного порівняння швидкодії та якості роботи моделей різного масштабу на базі реальних відеоданих системи моніторингу інтенсивності трафіку**.**

Аналіз публікацій, алгоритмів та методів

Для забезпечення об'єктивності та порівнянності результатів, тестування проводилося за уніфікованою методикою. Основним вимірюваним параметром був загальний час, витрачений серверною частиною на повний аналіз кожного відеофайлу, а також швидкість аналізу в кадрах за секунду (FPS). Цей час включав повний цикл обробки: завантаження відповідної моделі YOLO, зчитування та декодування відеопотоку, детекцію об'єктів нейромережею та роботу алгоритму трекінгу. Слід зазначити, що час завантаження файлу на сервер та час передачі фінальних результатів клієнту не враховувався, щоб зосередитися виключно на продуктивності аналітичного ядра

системи Для всіх тестів використовувалися наступні консистентні налаштування, взяті з конфігурації розробленої системи:

FRAME_SKIP_INTERVAL = 2: аналізу підлягав кожен другий кадр для оптимізації навантаження.

CONFIDENCE_THRESHOLD = 0.2: поріг впевненості детектора [4].

Цільові класи: 'car', 'motorcycle', 'bus', 'truck', 'person'.

Процедура тестування передбачала послідовний аналіз двох тестових відеофайлів різної складності кожною з п'яти доступних моделей YOLO (nano, small, medium, large, x-large). Для забезпечення статистичної достовірності результатів, кожен тест (комбінація "модель + файл") проводився тричі, після чого розраховувалося середнє арифметичне значення часу обробки.

Алгоритмічна основа трекінгу (ByteTrack). Для вирішення завдання відстеження об'єктів у системі використовується сучасний алгоритм ByteTrack. Основний принцип його роботи полягає у дворівневій асоціації детекцій з існуючими треками, що робить його особливо стійким до перекриттів та тимчасових зникнень об'єктів х [3,7]. Логіку роботи алгоритму наведено на рис. 1.

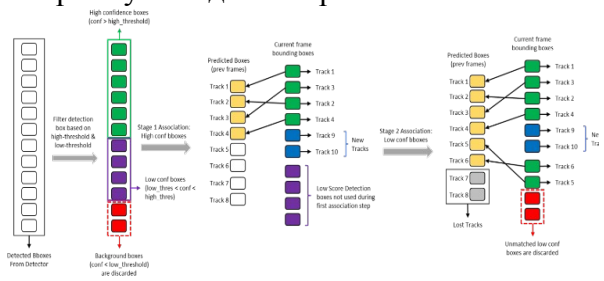

Рис. 1. Блок-схема алгоритму ByteTrack [3]

На першому етапі асоціації ByteTrack намагається зіставити детекції з високим рівнем впевненості (high confidence boxes) з активними треками. Це зіставлення зазвичай базується на метриці IoU (Intersection over Union) між обмежувальною рамкою передбаченого положення треку та новою детекцією.

На другому етапі детекції з низьким рівнем впевненості (low confidence boxes) не відкидаються одразу, як це робиться в багатьох інших трекерах. Замість цього, ByteTrack використовує ці "слабкі" детекції для реасоціації з тими треками, які не знайшли відповідності на першому етапі. Це ключова особливість, яка дозволяє системі ефективно "тримати" треки об'єктів навіть під час короткочасних перешкод, коли впевненість детектора падає [3].

Для глибшого розуміння механізму асоціації доцільно розглянути базовий принцип роботи IoU-трекера, який лежить в основі геометричної складової ByteTrack і зображений на рис. 2. Схема демонструє просторово-часову модель руху об'єктів, де сині та червоні площини відображають послідовні положення обмежувальних рамок (bounding boxes) у тривимірному просторі координат і часу. Ключова ідея полягає в тому, що за умови високої частоти кадрів та достатньої точності детектора зміщення об'єкта між сусідніми кадрами є незначним. Це дозволяє виконувати трекінг шляхом простого обчислення площі перекриття (Intersection over Union) між детекціями на поточному та попередньому кадрах: якщо просторове перекриття є значним, система з високою ймовірністю пов'язує ці детекції в одну траєкторію, що показано стрілками, які з'єднують послідовні положення об'єктів [6].

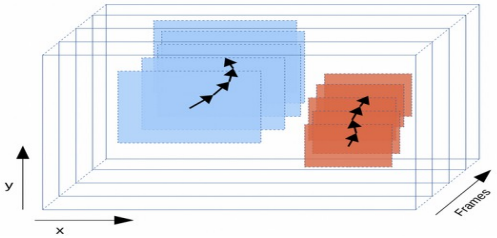

Рис. 2. Базовий принцип IoU-трекера [6]

Обґрунтування вибору саме алгоритму ByteTrack для даного дослідження базується на його доведеній перевазі над іншими сучасними методами, що наочно ілюструє діаграма порівняння продуктивності (рис. 3). Графік відображає співвідношення двох критичних метрик: точності відстеження (MOTA - вісь Y) та швидкості обробки (FPS - вісь X). Як видно з розподілу, ByteTrack (позначений найбільшим помаранчевим колом у правому верхньому куті) займає лідируючу позицію, демонструючи унікальний баланс характеристик. На відміну від трекерів типу FairMOT або CenterTrack, які поступаються або в точності, або в швидкості, ByteTrack забезпечує показник MOTA понад 80.0 при швидкості обробки близько 30 кадрів за секунду. Це свідчить про те, що даний алгоритм є оптимальним рішенням для систем реального часу, оскільки він мінімізує обчислювальні витрати, зберігаючи при цьому найвищу якість утримання траєкторій серед розглянутих аналогів [6].

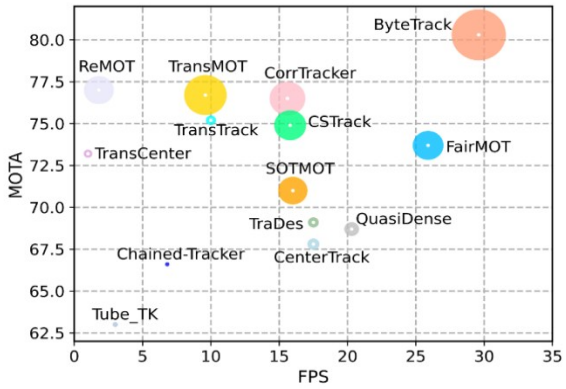

Рис. 3. Порівняльна діаграма ефективності сучасних алгоритмів трекінгу (точність MOTA проти швидкості FPS)

Результати дослідження

Аналіз швидкодії моделей (Inference Speed). Результати вимірювань часу обробки та розрахованої середньої кількості кадрів за секунду (FPS) для кожного з двох тестових відеофайлів представлені на діаграмах нижче. Чітко простежується залежність: чим більша та складніша архітектура моделі YOLO, тим більше часу потрібно на аналіз відео і, відповідно, тим нижчим є середній FPS обробки.

Для першого тестового файлу (відео з помірною інтенсивністю руху) модель YOLOv8n виявилася найшвидшою, обробляючи потік за 20 секунд із середнім показником 73 FPS. Кожна наступна модель поступово збільшує час обробки: YOLOv8s – 21 с (~70 FPS), YOLOv8m – 22 с (~67 FPS), YOLOv8l – 30 с (~49 FPS). Найпотужніша модель YOLOv8x витратила на обробку 38 секунд, показавши результат 38.5 FPS. Таким чином, на цьому етапі модель X виявилася приблизно в 1.9 рази повільнішою за найлегшу модель N (рис. 4, табл. 1).

Рис. 4. Гістограми для першого файлу

Таблиця 1

Табличне представлення результатів аналізу для першого файлу

	Час аналізу	Середня кількість FPS
YOLOv8 n	10 с	35,97
YOLOv8 s	12 с	32,04
YOLOv8 m	13 с	28,84
YOLOv8 l	21 с	18,29
YOLOv8 x	22 с	17,02

Для другого файлу, що характеризується складними умовами (щільний трафік, багато об'єктів), тенденція зберігається, але абсолютні показники FPS знижуються. YOLOv8n обробила цей файл із середньою швидкістю 36 FPS. Показники інших моделей склали: YOLOv8s – 32 FPS, YOLOv8m – 29 FPS, YOLOv8l – 18 FPS, і YOLOv8x – 17 FPS. На цьому складному файлі різниця у швидкодії між найлегшою та найважчою моделями зросла до 2.2 раза (рис. 5, табл. 2).

Рис. 5. Гістограми для 2 файлу

Таблиця 2

Табличне представлення результатів аналізу для другого файлу

	Час аналізу	Середня кількість FPS
YOLOv8 n	20 с	73,01
YOLOv8 s	21 с	69,75
YOLOv8 m	22 с	67,04
YOLOv8 l	30 с	49,35
YOLOv8 x	38 с	38,51

Варто зазначити, що для "Файлу 1" навіть найповільніша модель YOLOv8x забезпечує FPS вищий за реальний FPS відео (38.5 оброблених кадрів/с проти 25 реальних). Однак для "Файлу 2" показники FPS нижчі, що пов'язано зі збільшенням часу на аналіз кожного кадру нейромережею через насиченість сцени об'єктами.

Порівняльний аналіз якості трекінгу.

Критично важливим етапом дослідження стало порівняння не тільки кількісних показників швидкості, але й якісних аспектів роботи системи - зокрема, кількості унікальних стійких треків. Для цього аналізу використовувався складний файл (Файл 2). Тестування проводилося з параметром MIN_TRACK_PERSISTENCE_FRAMES = 10, що означає, що об'єкт вважається валідним лише якщо він утримується системою щонайменше 10 кадрів. Додаткові параметри надаються: CONFIDENCE_THRESHOLD = 0.2, FRAME_SKIP_INTERVAL = 2.

Результати виявили нелінійну залежність між розміром моделі та кількістю зафіксованих унікальних об'єктів (рис. 6).

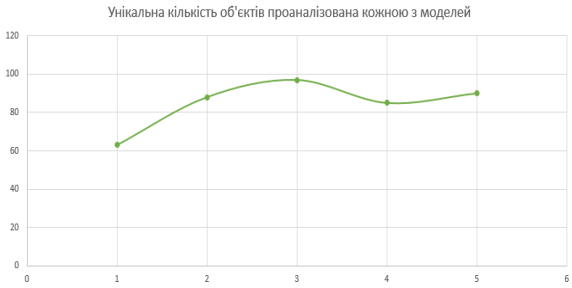

Рис. 6. Графічне порівняння кількості виявлених об'єктів, n -> x

Нижче наведено детальний аналіз результатів для кожної з п'яти протестованих архітектур.

Модель YOLOv8n (Nano). Найлегша модель показала найнижчий результат, зафіксувавши лише 63 унікальні стійкі об'єкти. Через спрощену архітектуру та малу кількість параметрів, ця модель має низьку дискримінативну здатність. Вона часто пропускає дрібні об'єкти (пішоходів на задньому плані) та не здатна утримувати детекцію, коли об'єкт частково перекривається іншим транспортним засобом [2]. Це призводить до того, що трекер ByteTrack не отримує достатньо даних для підтримки безперервності треку, і об'єкт відфільтровується як "шум".

Модель YOLOv8s (Small). Перехід до моделі Small продемонстрував значний стрибок у якості. Система зафіксувала 88 унікальних об'єктів, що на 39.6% більше, ніж у версії Nano. Це свідчить про те, що навіть незначне збільшення глибини нейромережі дозволяє значно краще виділяти ознаки об'єктів у складних сценах. Модель стала більш "впевненою" у детекції, що дозволило трекеру сформувати стійкіші траєкторії.

Модель YOLOv8m (Medium). Модель Medium продемонструвала найкращий результат у всьому експерименті - 97 унікальних стійких об'єктів. Цей показник є піковим і вказує на те, що архітектура Medium забезпечує оптимальний баланс для даного типу відеоданих. Детекції, генеровані цією моделлю, є не лише точними, але й, що критично важливо для алгоритму ByteTrack, геометрично стабільними. Рамки об'єктів не "стрибають" від кадру до кадру, що дозволяє досягти максимальної ефективності асоціації через метрику IoU.

Модель YOLOv8l (Large). Несподівано, при переході до більш потужної моделі Large, кількість зафіксованих унікальних об'єктів знизилася до 85. Це менше не тільки ніж у моделі Medium, але й навіть трохи менше, ніж у моделі Small. Такий спад пояснюється появою надмірної чутливості: модель починає реагувати на найменші зміни у позі пішоходів або відблисках на автомобілях, що призводить до нестабільності розмірів обмежувальних рамок (bounding boxes) та фрагментації єдиного треку на декілька коротких, які відсіюються фільтром тривалості.

Модель YOLOv8x (X-Large). Найпотужніша з доступних моделей, YOLOv8x, зафіксувала 90 унікальних об'єктів. Хоча це кращий результат, ніж у версії Large, він все одно поступається показнику моделі Medium (97). Незважаючи на величезні обчислювальні витрати (найнижчий FPS), ця модель не гарантує найкращого трекінгу. Висока деталізація детекції іноді призводить до короткочасної зміни класу об'єкта (наприклад, "Truck" на частку секунди розпізнається як "Bus"), що збиває алгоритм трекінгу і призводить до втрати ID.

Візуальний аналіз якості детекції (First Frame Analysis).

Для глибшого розуміння причин розбіжності у статистиці було проведено детальний візуальний аналіз роботи крайніх моделей лінійки - найлегшої (Nano) та найважчої (X-Large) - на прикладі одного й того ж кадру відеопотоку.

Аналіз роботи YOLOv8n. На рис. 7 представлено результат обробки кадру моделлю Nano. Як видно із зображення, модель демонструє обмежену здатність до розпізнавання сцени.

Пропуски об'єктів. Модель не змогла ідентифікувати великий автобус на задньому плані, а також пропустила значну частину пішоходів, що знаходяться далі від камери.

Неточні рамки. Обмежувальні рамки навколо виявлених автомобілів часто є нещільними або охоплюють лише частину об'єкта.

Втрата в натовпі. У зонах скупчення людей модель часто зливає декількох пішоходів в один об'єкт або ігнорує тих, хто частково перекритий. Це пояснює найнижчий показник унікальних треків у загальній статистиці порівняння моделей.

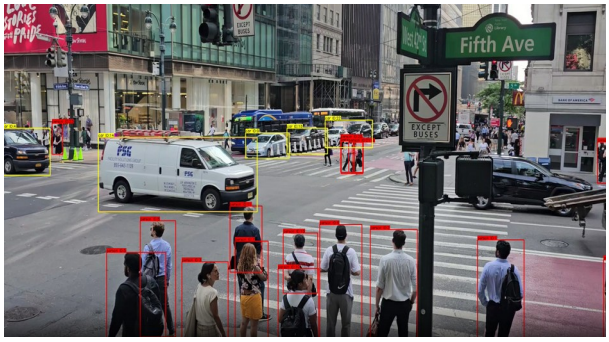

Рис. 7. Перший кадр аналізу для YOLOv8n

Аналіз роботи YOLOv8x

На рис. 8 зображено результат роботи моделі X-Large на тому ж кадрі. Різниця в деталізації є очевидною.

Повна сцена: Модель успішно детектувала автобус, який пропустила версія Nano, а також розпізнала автомобілі, що знаходяться на значній відстані.

Висока щільність детекції: У натовпі пішоходів модель виділила майже кожного індивідуума окремою рамкою, навіть за умов сильного перекриття.

Точність рамок: Bounding boxes дуже щільно прилягають до контурів об'єктів.

Однак, саме ця висока чутливість і створює парадокс трекінгу. На відео в динаміці ці ідеально точні рамки можуть "тремтіти" (змінювати розмір на кілька пікселів) або зникати/з'являтися у складних зонах, що алгоритм ByteTrack, який базується на перекритті площ (IoU), може інтерпретувати як появу нових об'єктів, розриваючи цілісність історії руху.

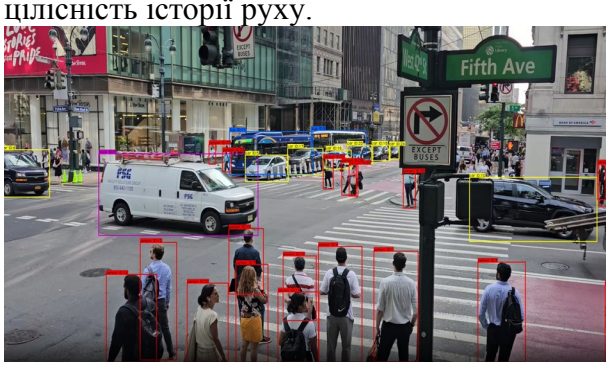

Рис. 8. Перший кадр аналізу для YOLOv8x

Обговорення результатів

Проведене дослідження виявило неочевидну та науково значущу нелінійну залежність між обчислювальною складністю нейромережевого детектора та підсумковою якістю мультиоб'єктного трекінгу. Отримані результати спростовують поширене інтуїтивне припущення про те, що використання "важчих" моделей (Large та X-Large) автоматично гарантує найкращий результат у задачах відеоаналітики. Феномен зниження кількості унікальних стійких треків для моделі YOLOv8x (90 об'єктів) порівняно з моделлю YOLOv8m (97 об'єктів) вимагає детального пояснення через призму механіки роботи алгоритмів асоціації.

Головною причиною такої поведінки є явище, яке в літературі часто називають "тремтінням детекцій" (detection jitter) або нестабільністю обмежувальних рамок. Моделі з великою кількістю параметрів (YOLOv8l/x) володіють надзвичайно високою чутливістю до дрібних деталей зображення. У динамічному відеопотоці, де освітлення та кут огляду постійно змінюються, ця чутливість призводить до того, що нейромережа реагує на мікроскопічні зміни пікселів. В результаті, координати та розміри обмежувальної рамки (bounding box) одного й того ж об'єкта можуть суттєво флуктувати від кадру до кадру. Крім того, спостерігалася нестабільність класифікації: наприклад, транспортний засіб міг на частку секунди змінити клас з "Truck" на "Bus" через зміну ракурсу.

Алгоритм ByteTrack, інтегрований у досліджувану систему, значною мірою покладається на геометричну метрику IoU (Intersection over Union) для зв'язування детекцій між сусідніми кадрами. Коли обмежувальна рамка "тремтить" або різко змінює форму через надмірну чутливість детектора,

значення IoU падає нижче порогового рівня асоціації. Математично трекер інтерпретує це як зникнення старого об'єкта та появу нового. Це призводить до фрагментації траєкторії: один реальний автомобіль, що перетинає перехрестя, отримує кілька різних ідентифікаторів (ID). Оскільки в експерименті застосовувався фільтр стійкості (MIN_TRACK_PERSISTENCE = 10 кадрів), короткі фрагментовані треки відсіювалися як шум, що і призвело до парадоксального зниження підсумкової статистики для найпотужніших моделей.

З іншого боку, моделі молодшого сегменту (YOLOv8n та YOLOv8s) продемонстрували класичну проблему недостатньої дискримінативної здатності. Спрощена архітектура не дозволяє їм ефективно виділяти ознаки об'єктів у складних умовах - наприклад, при частковій оклюзії (коли пішохід перекритий автомобілем) або при низькому контрасті з фоном. Це призводить до пропусків детекцій (False Negatives), що робить неможливим відновлення треку навіть за допомогою дворівневої асоціації ByteTrack.

Таким чином, модель YOLOv8m (Medium) виявилася оптимальним технологічним рішенням. Вона забезпечує необхідний баланс: її архітектура достатньо глибока для впевненої детекції об'єктів у складному трафіку, але при цьому вона генерує стабільніші та "спокійніші" передбачення, ніж версія X-Large. Це забезпечує високу часову когерентність (temporal consistency) даних, що є критично важливим для коректної роботи трекера на базі фільтра Калмана та IoU.

Висновки

У статті було успішно реалізовано та досліджено архітектуру системи комп'ютерного зору для моніторингу дорожнього трафіку, що поєднує сучасний детектор YOLOv8 та алгоритм асоціації ByteTrack. На основі проведеного порівняльного аналізу можна сформулювати наступні підсумкові висновки:

Визначення оптимальної конфігурації. Експериментально доведено, що для задач міського моніторингу в реальному часі на обладнанні середнього класу (рівня NVIDIA RTX 2060) найкращим вибором є модель YOLOv8m. Вона продемонструвала найвищу ефективність трекінгу, зафіксувавши 97 унікальних стійких об'єктів, що на 12.3% більше за результат моделі Large та на 7.7% більше за результат найпотужнішої моделі X-Large. При цьому швидкість обробки склала близько 29 FPS, що відповідає вимогам до систем реального часу.

Вплив апаратного прискорення. Дослідження підтвердило критичну необхідність використання технології CUDA для задач відеоаналітики. Перехід від обробки на CPU до GPU дозволив підвищити продуктивність системи у 5–30 разів залежно від складності моделі. Без використання графічних прискорювачів застосування сучасних алгоритмів трекінгу та складних нейромереж у реальному часі є неможливим [5].

Ефективність алгоритму ByteTrack. Інтеграція ByteTrack дозволила значно підвищити стійкість системи до оклюзій. Використання низькопорогових детекцій на другому етапі асоціації дозволило відновлювати траєкторії об'єктів, які частково перекривалися або мали низьку впевненість детекції, що є типовим сценарієм для щільного міського трафіку.

Практичні рекомендації. Розробникам подібних систем не варто сліпо гнатися за найпотужнішими моделями (State-of-the-Art за метрикою mAP). Як показало дослідження, надмірна чутливість детектора може негативно впливати на стабільність трекінгу. Рекомендується проводити

емпіричний підбір моделі ("тюнінг") під конкретні умови освітлення та щільності потоку, віддаючи перевагу моделям середньої складності для забезпечення стабільності траєкторій.

Перспективи подальших досліджень полягають у тестуванні методів згладжування траєкторій (post-processing smoothing), інтеграції більш стійких до "тремтіння" трекерів (наприклад, OC-SORT) та оптимізації моделей за допомогою технологій квантування (TensorRT) для розгортання на енергоефективних пристроях.

Література

1. Bernardin K. Evaluating multiple object tracking performance: the CLEAR MOT metrics [Electronic resource] / K. Bernardin, R. Stiefelhagen // EURASIP Journal on Image and Video Processing. – 2008. – Article ID 246309. – URL: https://jivp-eurasipjournals.springeropen.com/articles/10.1155/2008/246309 (дата звернення: 30.11.2025).

2. Vehicle Flow Detection and Tracking Based on an Improved YOLOv8n and ByteTrack Framework [Electronic resource] // MDPI Drones. – 2025. – URL: https://www.mdpi.com/2032-6653/16/1/13 (дата звернення: 30.11.2025).

3. Introduction to ByteTrack: Multi-Object Tracking by Associating Every Detection Box [Electronic resource] / Datature. – Electronic data. – URL: https://www.datature.io/blog/introduction-to-bytetrack-multi-object-tracking-by-associating-every-detection-box (дата звернення: 30.11.2025).

4. Tracking with Ultralytics YOLO [Electronic resource] / Ultralytics Official Documentation. – Electronic data. – URL: https://docs.ultralytics.com/modes/track/ (дата звернення: 30.11.2025).

5. CUDA Toolkit 12.0 Released for General Availability [Electronic resource] / NVIDIA Developer Blog. – Electronic data. – URL: https://developer.nvidia.com/blog/cuda-toolkit-12-0-released-for-general-availability/ (дата звернення: 30.11.2025).

6. A brief note on ByteTrack [Electronic resource] / Qure.ai Technical Blog. – Electronic data. – URL: https://blog.qure.ai/notes/byte-track-multi-object-tracking (дата звернення: 30.11.2025).

7. Simple and Fast Multi-Object Tracking on Video: SmileTrack [Electronic resource] / AI-SCHOLAR. – Electronic data. – URL: https://ai-scholar.tech/en/articles/object-tracking/smiletrack (дата звернення: 30.11.2025).

Климова А.С., Полухін А.В., Карий І.М.

ЕКСПЕРІМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МАСШТАБОВАНИХ АРХІТЕКТУР YOLOV8(n-x) У ЗАДАЧАХ МУЛЬТИОБ'ЄКТНОГО СУПРОВОДУ НА БАЗІ АЛГОРИТМУ BYTETRACK

У статті розглядаються результати досліджень продуктивності системи комп'ютерного зору для задач інтелектуального моніторингу дорожнього трафіку. Досліджено проблему пошуку балансу між швидкістю інференсу (Inference Speed) та стійкістю супроводу об'єктів (Tracking Stability) в умовах щільного міського потоку.

Класичні підходи до мультиоб'єктного трекінгу (MOT) часто стикаються з проблемою втрати об'єктів під час часткового перекриття (оклюзії). Сучасний алгоритм ByteTrack пропонує вирішення цієї проблеми шляхом використання дворівневої асоціації детекцій з існуючими треками, що робить його особливо стійким до перекриттів. Цей алгоритм є оптимальним рішенням для систем реального часу, оскільки він мінімізує обчислювальні витрати, зберігаючи при цьому найвищу якість утримання траєкторій серед розглянутих аналогів . Було розглянуто базовий принцип роботи IoU-трекера, який лежить в основі геометричної складової ByteTrack.

В роботі експериментально доведено, що для задач міського моніторингу в реальному часі на обладнанні середнього класу (рівня NVIDIA RTX 2060) найкращим вибором є модель YOLOv8m. В роботі наведено детальний аналіз п'яти протестованих архітектур YOLOv8m: Nano, Small, Medium, Large та X-Large. Шляхом емпіричного порівняння швидкодії та якості роботи моделей різного масштабу на базі реальних відеоданих системи моніторингу інтенсивності трафіку визначена оптимальна архітектури нейромережі сімейства YOLOv8 для інтеграції з ByteTrack.

В результаті аналізу модифікацій нейромережі YOLOv8 у поєднанні з трекером ByteTrack зроблено висновок про значущу нелінійну залежність між обчислювальною складністю нейромережевого детектора та підсумковою якістю мультиоб'єктного трекінгу.

Ключові слова: YOLOv8, ByteTrack, комп’ютерний зір, мультиоб'єктний трекінг (MOT), CUDA, відеоаналітика, асоціація даних.

Klimova A.S., Polukhin A.V., Karyi I.M.

EXPERIMENTAL STUDY OF THE EFFICIENCY OF SCALABLE YOLOV8(n-x) ARCHITECTURES IN MULTI-OBJECT TRACKING TASKS BASED ON THE BYTETRACK ALGORITHM

The article is devoted to the results of research on the performance of a computer vision system for intelligent road traffic monitoring tasks. The problem of finding a balance between Inference Speed and Tracking Stability in dense urban traffic conditions is investigated.

Classic approaches to multi-object tracking (MOT) often face the problem of object loss during partial overlap (occlusion). The modern ByteTrack algorithm offers a solution to this problem by using two-level association of detections with existing tracks, which makes it particularly resistant to overlaps. This algorithm is the optimal solution for real-time systems, as it minimises computational costs while maintaining the highest trajectory retention quality among the considered analogues. The basic principle of the IoU tracker, which underlies the geometric component of ByteTrack, was considered. The paper

experimentally proves that for real-time urban monitoring tasks on mid-range hardware (NVIDIA RTX 2060 level), the YOLOv8m model is the best choice. The paper provides a detailed analysis of the five tested YOLOv8m architectures: Nano, Small, Medium, Large, and X-Large. Through empirical comparison of the performance and quality of models of different scales based on real video data from a traffic intensity monitoring system, the optimal neural network architecture of the YOLOv8 family for integration with ByteTrack was determined.

As a result of the analysis of modifications of the YOLOv8 neural network in combination with the ByteTrack tracker, a conclusion is drawn regarding a significant non-linear dependence between the computational complexity of the neural network detector and the final quality of multi-object tracking.

Keywords: YOLOv8, ByteTrack, computer vision, multi-object tracking (MOT), CUDA, video analytics, data association.

Стаття подана до редакції: 9/12/2025
Стаття прийнята до опублікування:17/12/2025
Стаття опублікована: 30/12/2025