DOI 10.31891/2307-5732-2024-335-3-9
УДК 004.773.2

ДУМИН ІРИНА
Національний університет "Львівська політехніка"
https://orcid.org/0000-0001-5569-2647
e-mail: iryna.b.shvorob@lpnu.ua
БОРСУК ВАСИЛЬ
Pinata Farms, Inc.
https://orcid.org/0009-0009-1063-1010
e-mail: vas.borsuk@gmail.com
ШАХОВСЬКА ХРИСТИНА
N-iX LLC
https://orcid.org/0000-0002-9914-229X
e-mail: kristin.shakhovska@gmail.com

РОЗРОБКА ЕФЕКТИВНОГО ТА ТОЧНОГО МЕТОДУ ВІДСТЕЖЕННЯ ОБ'ЄКТІВ НА ВІДЕО НА МОБІЛЬНИХ ПРИСТРОЯХ

В роботі наведено результати розроблення методу відстеження об'єктів за допомогою сіамських нейронних мереж. Крім того, представлено новий тест ефективності методу та протокол, де ефективність визначається як споживанням енергії, так і швидкістю виконання на периферійних пристроях

Ключові слова: візуальне відстеження об'єктів, сіамські нейронні мереж, периферійні пристрої.

DUMYN IRYNA
Lviv Polytechnic National University
BORSUK VASYL
Pinata Farms
SHAKHOVSKA KHRYSTYNA
Inc, N-iX LLC

DEVELOPMENT OF AN EFFECTIVE AND ACCURATE METHOD OF VISUAL VIDEO OBJECT TRACKING ON MOBILE DEVICES

One of the top research topics in computer vision area is visual object tracking. Main goal is to obtain the target object's location in the first video frame of a given video sequence. The recent innovations of deep neural networks, specifically Siamese networks has significant impact on visual object tracking. In spite of high accuracy and high results in academic benchmarks, there are drawbacks in current state-of-the-art approaches in particular compute-intensive and large memory footprint that cannot satisfy the performance requirements of real-world applications. The aim of this paper is to design a new lightweight framework for resource-efficient and accurate visual object tracking. To add, a new tracker of efficiency benchmark and protocol were introduced. Efficiency is defined in terms of both energy consumption and execution speed on edge devices. New dual template representation for object model adaptation was developed. The first template, static, fixes the original visible appearance and thus prevents deviation and, as a result, failures caused by adaptation. The other is dynamic; the state reflects the current conditions of assembly and the appearance of its object. Unlike STARK, which incorporates additional timing information by introducing a separate estimation prediction head, we introduce parameter-free module similarity as a template update rule optimized from the latest network. We show that the learned convex combination of two patterns is effective for tracking on multiple tests. A lightweight tracker was proposed, which includes functions, dual representation of patterns and pixel-by-pixel merged blocks in its compact network. The resulting FEAR-XS tracker runs at 205 FPS on the iPhone 11, which is 4.2 times faster than LightTrack and 26.6 times faster than Ocean, with high accuracy on many tests – no state-of-the-art tracker is more accurate and faster than any FEAR tracker. In addition, the algorithm is highly energy efficient.

Keywords: visual object tracking, siamese neural network, edge devices.

Постановка проблеми

Візуальне відстеження об'єктів — це одна з найбільш фундаментальних тем дослідження в області комп'ютерного зору, метою якої є визначення розташування цільового об'єкта у відео послідовності за початковим станом об'єкта в першому кадрі відео. Недавній розвиток глибоких нейронних мереж, зокрема сіамських нейронних мереж (Siamese neural networks), призвів до значного прогресу у візуальному відстеженні об'єктів. Незважаючи на точність та високі результати за академічними тестами, поточні підходи потребують важких обчислень і займають значний обсяг пам'яті, що не може задовольнити вимоги продуктивності додатків. Метою цієї роботи є розробити новий метод для ресурсоефективного та точного візуального відстеження об'єктів. Крім того, представлено новий тест ефективності трекера та протокол, де ефективність визначається як споживанням енергії, так і швидкістю виконання на периферійних пристроях.

Аналіз останніх джерел

У звичайних контрольних тестах відстеження, таких як щорічні тести VOT і онлайн-тест відстеження, історично домінували розроблені вручну рішення на основі функцій [1, 2]. З розвитком глибокого навчання вони втратили популярність, склавши лише 14% моделей учасників VOT-ST2020.

Останнім часом задачу короткочасного візуального відстеження об'єктів в основному вирішували за допомогою дискримінаційних кореляційних фільтрів [3] або сіамських нейронних мереж, а також обидва разом.

Трекери на основі сіамських кореляційних мереж виконують відстеження на основі офлайн-навчання функції відповідності. Ця функція діє як показник подібності між характеристиками шаблонного

зображення та обрізаною площиною з кандидатської області пошуку. Сіамські трекери спочатку стали популярними завдяки вражаючому компромісу між точністю та ефективністю [2]; однак вони не могли встигати за точністю методів онлайн-навчання [3]. Один із найсучасніших методів, Ocean [12], включає парадигму виявлення об'єктів FCOS [10] без прив'язки для відстеження, прямо регресуючи відстань від точки на карті класифікації до кутів обмежувальної рамки. Інший сучасний підхід, STARK [11], представляє кодер-декодер на основі трансформатора в сіамській манері: плоский і об'єднаний пошук і карти функцій шаблонів служать вхідними даними для трансформаторної мережі. Жодна з вищеназваних найсучасніших архітектур явно не вирішує завдання швидкого, високоякісного візуального відстеження об'єктів у різноманітних архітектурах GPU.

Розробка ефективних і легких нейронних мереж, оптимізованих для виконання на мобільних пристроях, привернула велику увагу останні кілька років завдяки багатьом практичним застосуванням. SqueezeNet [7] була однією з перших робіт, спрямованих на зменшення розміру нейронної мережі. Вони запровадили ефективну стратегію зменшення дискретизації, широке використання згорткових блоків 1x1 і кілька менших модулів для значного зменшення розміру мережі.

Для трекерів FEAR було дотримано найкращих практик для розробки ефективної та гнучкої архітектури нейронної мережі. Для надзвичайно полегшеної версії, де це було можливо, було використано згортки, які можна розділити по глибині, замість звичайних і розробили мережеві рівні таким чином, щоб блоки Conv-BN-ReLU могли бути об'єднані на етапі експорту.

Виклад основного матеріалу

Тренувальні дані. Ця робота використовує кілька загальнодоступних наборів даних відстеження відеооб'єктів для навчання запропонованого трекера.

YouTube-BoundingBoxes — це великомасштабний набір даних відео з високоякісними анотаціями обмежувальних рамок із щільною вибіркою для одного об'єкта. Набір даних складається з приблизно 380 000 відеосегментів тривалістю 15-20 секунд, витягнутих із 240 000 різних загальнодоступних відео YouTube, автоматично вибраних для показу об'єктів у природних умовах без редагування чи пост-обробки, з якістю запису, яка часто нагадує якість камери мобільного телефону. Усі ці відеосегменти були анотовані людиною за допомогою високоточних класифікацій і обмежувальних рамок зі швидкістю 1 кадр на секунду, що загалом містило понад 5,6 млн. обмежувальних рамок.

LaSOT — це високоякісний тест, що складається з 1400 послідовностей із загальною кількістю понад 3,5 млн кадрів. Кожне відео анотується вручну та точно зі швидкістю 30 кадрів на секунду з обмежувальною рамкою, що робить його одним із найбільших еталонних показників із щільними анотаціями для довгострокового відстеження. Кожна послідовність містить 2500 кадрів у середньому, а набір даних представляє 70 різних категорій об'єктів.

ImageNet-VID — це тест, створений для завдання виявлення відеооб'єктів. Він містить 30 категорій об'єктів, які є підмножиною 200 категорій базового рівня завдання виявлення об'єктів. Кожне відео має щільні анотації зі швидкістю 30 кадрів на секунду з набором обмежувальних рамок. Загалом тест складається з майже 2 мільйонів анотацій і понад 4000 відеорядів.

Метод. Трекер розроблено у єдиній уніфікованій моделі, що складається з мережі виділення ознак, блоків об'єднання функцій і підмереж для конкретних завдань для регресії та класифікації обмежувальної рамки. Враховуючи статичне зображення шаблону, I_T, обрізане зображення пошуку, I_S, і динамічне зображення шаблону, I_d, мережа виявлення параметрів (Feature Extraction Network) моделі дає параметри над цими входами. Представлення функції шаблону потім обчислюється як лінійна інтерполяція між статичними та динамічними елементами зображення шаблону. Далі воно об'єднується з функціями пошукових зображень у попіксельних блоках об'єднання. Нарешті, отримані тензори передаються до підмережі класифікації, яка передбачає карту ознак імовірностей присутності об'єкта, і підмережі регресії, яка оцінює відстані від кожного пікселя в межах цільової обмежувальної рамки до чотирьох сторін обмежувальної рамки істинності землі. Кожен етап детально описано далі, а огляд запропонованої архітектури мережі проілюстровано на рисунку 1.

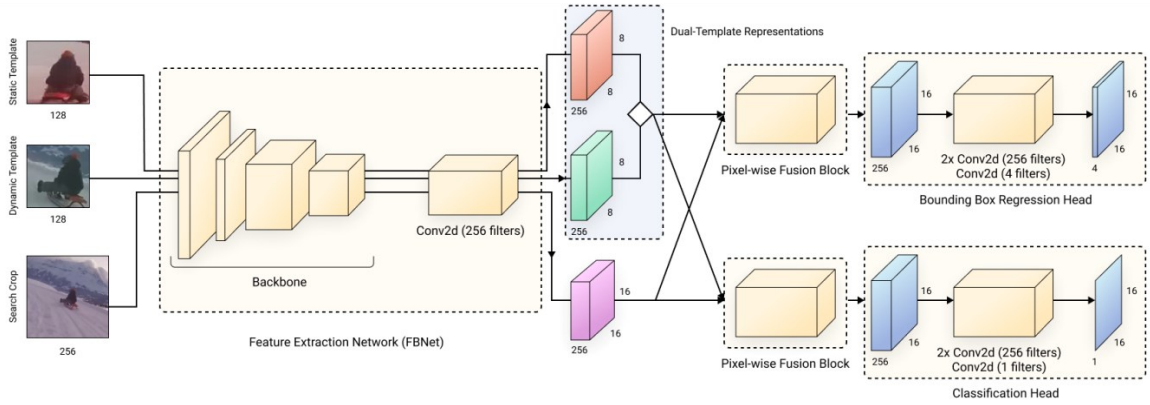

Рис. 1. Архітектура запропонованої мережі

Мережа виявлення параметрів. Ефективний конвеєр відстеження вимагає гнучкого, легкого, і точного виділення параметрів. Крім того, вихідні дані такої магістральної мережі повинні мати достатньо високу просторову роздільну здатність, щоб мати оптимальну функцію локалізації об'єкта [9], не збільшуючи при цьому обчислення для послідовних рівнів. Більшість сучасних сіамських трекерів підвищують просторову роздільну здатність останньої карти ознак, яка значно погіршує продуктивність наступних шарів. Було виявлено, що збереження вихідної просторової роздільної здатності значно знижує обчислювальні витрати як на мережу виявлення параметрів, так і на головки прогнозування, як показано в таблиці 1.

В роботі використано перші чотири етапи нейронної мережі, попередньо навчені на ImageNet [4] як модуль виявлення параметрів. Трекер FEAR-M використовує ResNet-50 [5], а трекер FEAR-L включає RegNet для найкращої якості відстеження, при тому залишаючись ефективною.

Результатом роботи модуля виявлення параметрів є карта функцій кроку 16 для шаблону і пошук зображень. Щоб відобразити глибину вихідної функції, відобразити постійне число каналів, використано простий AdjustLayer, який є комбінацією Convolutional і шару Batch Normalization [8].

Таблиця 1

GigaFLOPs, на кадр, архітектур трекера FEAR і OceanNet [12]; ↑ вказує на підвищену просторову роздільну здатність мережі щодо виявлення параметрів. Масштабування має незначний вплив на точність, водночас значно збільшуючи FLOP.

Model architecture	Backbone GigaFLOPs	Prediction heads GigaFLOPs
FEAR-XS tracker	0.318	0.160
FEAR-XS tracker↑	0.840	0.746
OceanNet	4.106	1.178
OceanNet ↑ (original)	14.137	11.843

Для підвищення ефективності під час виконання на мобільних пристроях, для мобільної версії трекера - FEAR-XS - використано сімейство моделей FBNet розроблений через NAS. Таблиця 1 демонструє, що навіть легкий кодер не покращує ефективність моделі сучасних трекерів через складне передбачення голови. Таким чином, розробка легкого та точного декодера все ще є проблемою.

Під час виконання для кожних N кадрів обирається пошукове зображення з найбільшим косинусом подібності до представлення подвійного шаблону та оновлюється динамічний шаблон за допомогою передбаченої обмежувальної рамки в цьому кадрі. Крім того, для кожної тренувальної пари обирається негативне кадрування I_N з кадру, який не містить цільового об'єкта. Далі він проходить через мережу виявлення параметрів і витягується негативне обрізання, вбудовування e_N, подібно до пошукового зображення через зважену середню суму. Потім ми обчислюємо Triplet Loss [6] із вкладеннями e_T, e_S, e_N, витягнутими з F'_T, F_S і F_N відповідно. Ця навчальна схема надає сигнал для оцінки динамічного шаблону, водночас зміщуючи модель, щоб віддавати перевагу більш загальним представленням.

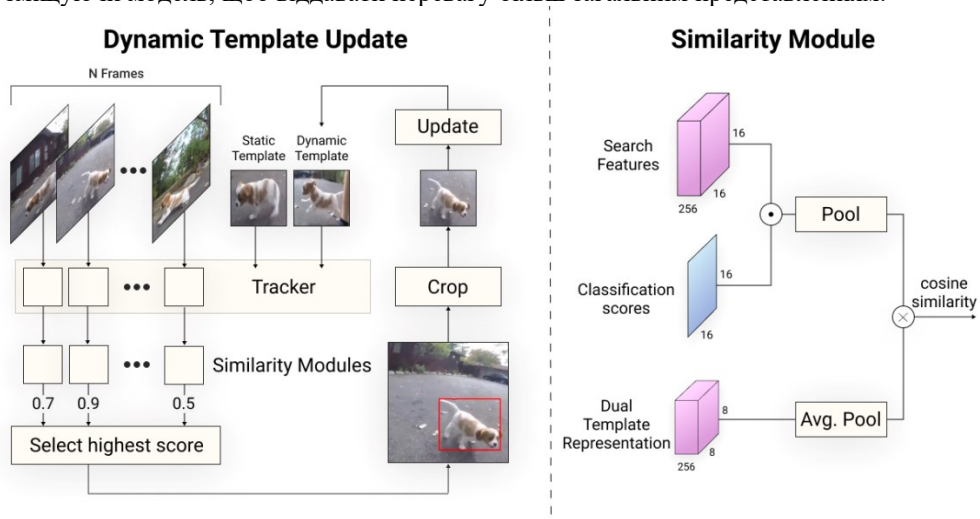

Рис. 2. Оновлення динамічного шаблону. Порівняння усередненого представлення подвійного шаблону з пошуковим зображенням, використовуючи косинусну подібність, і динамічно оновлюється представлення шаблону, коли зовнішній вигляд об'єкта різко змінюється

Попіксельний блок злиття. Модуль крос-кореляції створює спільне представлення шаблону та функцій пошукового зображення. Більшість існуючих сіамських трекерів використовують просту операцію крос-кореляції. Розширюючи цю ідею, в цій роботі представлено блок попіксельного злиття, який покращує інформацію про подібність, отриману за допомогою попіксельної кореляції з інформацією про положення та зовнішній вигляд, отриманою з пошукового зображення.

Далі пропускається карта функцій пошукового зображення через блок 3x3 Conv-BN-ReLU та обчислюється поточкова взаємна кореляція між цими функціями та функціями шаблонного зображення. Потім об'єднується розрахована карту кореляційних ознак із функціями пошукового зображення та передається результат через блок 1x1 Conv-BN-ReLU для агрегування їх. Завдяки цьому підходу вивчені ознаки є більш розрізняльними та можуть ефективно кодувати положення та зовнішній вигляд об'єкта. Загальна архітектура попіксельного блоку злиття зображена на рис. 3.

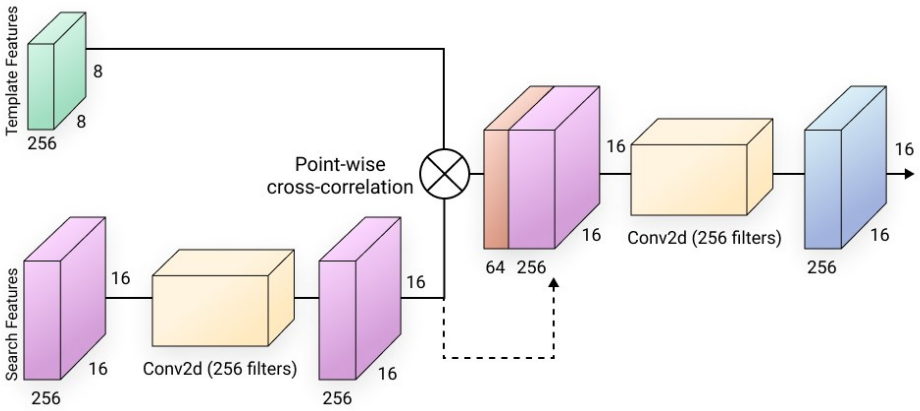

Рис. 3: Блок попіксельного злиття. Функції пошуку та шаблону об'єднані за допомогою модуля поточкової перехресної кореляції та збагачені функціями пошуку за допомогою конкатенації. Потім вихідні дані пересилаються до головок регресії

Головки регресії класифікації та обмежувальної рамки. Основна ідея регресійної головки обмежувальної рамки полягає в тому, щоб оцінити відстань від кожного пікселя в межах обмежувальної рамки цільового об'єкта до сторін обмежувальної рамки правди на землі [12]. Така регресія обмежувальної рамки враховує всі пікселі в рамці правди на землі під час навчання, тому вона може точно передбачити величину цільових об'єктів, навіть якщо лише крихітна частина сцени призначена як передній план. Мережа регресії обмежувальної рамки — це стек із двох простих блоків 3x3 Conv-BN-ReLU. Ми використовуємо лише два таких блоки замість чотирьох, запропонованих в Ocean [12], щоб зменшити обчислювальну складність. Керівник класифікації використовує ту саму структуру, що й головка регресії обмежувальної рамки. Єдина відмінність полягає в тому, що ми використовуємо один фільтр замість чотирьох в останньому блоці згортки. Цей заголовок передбачає карту оцінок 16x16, де кожен піксель представляє оцінку достовірності зовнішнього вигляду об'єкта у відповідній області кадру пошуку.

Загальна функція втрати. Навчання сіамської моделі відстеження вимагає багатокомпонентної цільової функції для одночасної оптимізації завдань класифікації та регресії. Як показано в попередніх підходах [12], втрати IoU і втрати класифікації використовуються для ефективного спільного навчання мереж регресії та класифікації. Крім того, для навчання трекерів FEAR доповнено ці навчальні цілі втратою триплетів, що дає змогу виконувати динамічне оновлення шаблону. Це покращує якість відстеження на 0,6% EAO за допомогою лише одного додаткового параметра, який можна навчити, і граничної вартості висновку. Наскільки відомо, це новий підхід до навчання трекерів об'єктів. Термін втрати триплету обчислюється на основі шаблонних (e_T), пошукових (e_S) і карт негативних зображень (e_N).

$$L_t = \max \{d(e_T , e_S) - d(e_T , e_N) + \text{margin}, 0))\} , \quad (2)$$

де $d(x_i, y_i) = \|x_i - y_i\|_2$. Термін втрати регресії обчислюється як:

$$L_{reg} = 1 - \sum_i \text{IoU} (t_{reg} , p_{reg}), \quad (3)$$

де t_{reg} позначає цільову обмежувальну рамку, preg позначає передбачувану обмежувальну рамку,і і індексує навчальні зразки. Для терміну втрати класифікації ми використовуємо Focal Loss:

$L_c = -(1 - p_t)^\gamma \log(p_t)$,

$$\text{pt} = \begin{cases} p \; if \; y = 1, \\ 1 - p \; в \; іншому \; випадку \end{cases} \quad (4)$$

Вище, y $\in$ {−1; 1} є елементом GT класу, і $0 \le p \le 1$ є передбаченою ймовірністю для класу s y = 1.

Загальна функція втрат є лінійною комбінацією трьох компонент:

$$L = \lambda_1 * L_t + \lambda_2 * L_{reg} + \lambda_3 * L_c. \quad (5)$$

На практиці, використовується 0.5, 1.0, 1.0 як λ_1, λ_2, λ_3, відповідно.

Результати

Навчання. Моделі реалізовано за допомогою PyTorch. Магістральна мережа ініціалізується за допомогою попередньо підготовлених ваг на ImageNet. Усі моделі навчаються на 4 графічних процесорах RTX A6000. Використано оптимізатор ADAM із швидкістю навчання = 4*10−4 і плато зниження швидкості навчання з коефіцієнтом = 0,5 кожні 10 епох моніторингу цільової метрики (середній IoU). Кожна епоха містить 10^6 пар зображень. Навчання займає 5 днів для зближення. Для кожної епохи випадковим чином

відібрано 20 000 зображень з LaSOT, 120 000 з COCO, 400 000 з YoutubeBB, 320 000 із GOT10k і 310 000 зображень із набору даних ImageNet отже, загалом 1 170 000 зображень використовується в кожній епосі.

З кожної відео послідовності в наборі даних випадковим чином відібрано шаблонний кадр I_T і пошуковий кадр I_S так, щоб відстань між ними становила d = 70 кадрів. Починаючи з 15 епохи, d збільшується на 2 кожну епоху. Це дозволяє мережі вивчати кореляцію між об'єктами спочатку на легших зразках і поступово збільшувати складність у міру навчання. Зображення динамічного шаблону вибирається з відео послідовності між статичним кадром шаблону та кадром пошукового зображення. Для негативного кадрування, де це можливо, обрано його з того самого кадру, що й динамічний шаблон, але без перекриття з цим кадруванням шаблону; інакше обирається негативне кадрування з іншої відео послідовності. Значення d було знайдено емпірично. Це узгоджується з приміткою в TrackingNet, що будь-який трекер є надійним протягом 1 секунди. За спостереженнями зовнішній вигляд об'єктів не змінюється різко протягом 2 секунд (60 кадрів), і встановлено d = 70 як компроміс між швидкістю логічного висновку та кількістю додатково включеної тимчасової інформації.

Попередня обробка. Обрано кадри шаблонного зображення з додатковим зміщенням 20% навколо обмежувальної рамки. Потім ми застосовано зсув світла (до 8 пікселів) і довільну зміну масштабу (до 5% з обох сторін), додається зображення до квадратного розміру із середнім значенням RGB кадрування та змінюється до розміру 128x128 пікселів. Застосовано такі ж розширення з більш серйозним зсувом (до 48 пікселів) і масштабом (між 65% і 135% від початкового розміру зображення) для пошукових і негативних зображень. Далі розмір зображення для пошуку змінюється до 256x256 пікселів із тією самою стратегією доповнення, що й у зображенні шаблону.

Тестування: під час виконання відстеження слідує тим же протоколам, що й у [2]. Характеристики статичних шаблонів цільового об'єкта обчислюються один раз у першому кадрі. Функції динамічного шаблону оновлюються кожні 70 кадрів і інтерполюються зі статичними функціями шаблону. Ці функції поєднуються з функціями пошуку зображень у кореляційні модулі, регресія та класифікаційні головки для отримання кінцевого результату.

У таблиці 2 наведено результати порівняння з VOT-ST2021. Запропонований трекер демонструє майже найкращу точність, перевершуючи LightTrack і STARKLightning на 3% і 4,4% EAO відповідно, маючи при цьому вищий FPS. Крім того, він лише на 2% відстає від Ocean, але має більш ніж у 18 разів менше параметрів ніж Ocean tracker, і він у 26 разів швидший за час визначення моделі на iPhone 11. Запропонований трекер демонструє ту саму EAO, що й трансформаторна мережа STARK-S50, але має набагато менше параметрів і ефективніша на мобільних пристроях.

Таблиця 2 додатково повідомляє споживання пам'яті моделлю та пікове споживання пам'яті під час прямого проходу в мегабайтах. Розміри моделей LightTrack і STARKLightning становлять 4,11 МБ і 6,28 МБ відповідно, тоді як наш метод споживає лише 3 МБ. Під час прямого проходу пікове використання пам'яті нашим трекером становить 10,1 МБ, LightTrack споживає трохи менше (9,21 МБ), використовуючи менше фільтрів у обмеженні згорткові шари коробкової регресії, а STARK-Lightning має пікове використання пам'яті 30,69 МБ через блоки самоуважності, які споживають пам'ять.

Таблиця 2

Порівняння з іншими алгоритмами на VOT-ST2021

	SiamFC++ (GoogleNet)	SiamRPN++ (MobileNet-V2)	SiamRPN++ (ResNet-50)	ATOM	KYS	Ocean (offline)	STARK (S50)	STARK (lightning)	LightTrack	FEAR-XS	FEAR-M	FEAR-L
EAO ↑	0,227	0,235	0,239	0,258	0,274	0,29	0,27	0,226	0,24	0,27	0,278	0,303
Accuracy ↑	0,418	0,432	0,438	0,457	0,453	0,479	0,464	0,433	0,417	0,471	0,476	0,501
Robustness ↑	0,667	0,656	0,668	0,691	0,736	0,732	0,719	0,627	0,684	0,708	0,728	0,755
iPhone 11 FPS ↑	7,11	6,86	3,49	-	-	7,72	11,2	87,41	49,13	205,12	56,2	38,3
Parameters (M) ↓	12,71	11,15	53,95	-	-	25,87	23,34	2,28	1,97	1,37	9,67	33,65
Memory (MB) ↓	24,77	21,63	103,74	-	-	102,81	109,63	6,28	4,11	3	18,82	66,24
Peak memory (MB) ↓	34,17	31,39	192,81	-	-	119,51	295,97	30,69	9,21	10,1	25,88	85,97

Висновки

У цій статті розглядається проблема ефективного відстеження візуальних об'єктів на мобільних пристроях. Основним вкладом цієї роботи є:

- Нове представлення подвійного шаблону для адаптації об'єктної моделі. Перший шаблон, статичний, закріплює оригінальний візуальний вигляд і таким чином запобігає девіації та, як наслідок, збоям, спричиненим адаптацією. Інший – динамічний; його стан відображає поточні умови комплектування та зовнішній вигляд об'єкта. На відміну від STARK [11s], який включає додаткову часову інформацію шляхом введення окремої головки прогнозування оцінки, ми вводимо модуль подібності без параметрів як правило оновлення шаблону, оптимізоване з рештою мережі. Ми показуємо, що вивчена опукла комбінація

двох шаблонів ефективна для відстеження на кількох тестах.

- Легкий трекер, який поєднує в собі компактну мережу вилучення функцій, подвійне представлення шаблонів і попіксельні об'єднані блоки. Отриманий трекер FEAR-XS працює зі швидкістю 205 FPS на iPhone 11, що на 4,2 раза швидше, ніж LightTrack і на 26,6 раза швидше, ніж Ocean [67], із високою точністю на багатьох тестах – жоден найсучасніший трекер не є точнішим і швидшим, ніж будь-який трекер FEAR. Крім того, алгоритм має високу енергоефективність.

Література

1. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1401–1409 (2016)

2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)

3. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6182–6191 (2019)

4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

6. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International workshop on similarity-based pattern recognition. pp. 84–92. Springer (2015)

7. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.:Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size.arXiv:1602.07360 (2016)

8. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456.PMLR (2015)

9. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of Siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4282–4291 (2019)

10. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In:Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)

11. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. arXiv preprint arXiv:2103.17154 (2021)

12. Zhang, Z., Peng, H., Fu, J., Li, B., Hu, W.: Ocean: Object-aware anchor-free tracking. In:Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28,2020, Proceedings, Part XXI 16. pp. 771–787. Springer (2020)