УДК 004.932.2

Дослідження застосування ознак швидкодіючих нейронних мереж у системах відслідковування на основі дискримінантних кореляційних фільтрів

Варфоломеєв А. Ю.

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

E-mail: *a.varfolomieiev@kpi.ua*

Візуальне відслідковування об'єктів є важливою задачею комп'ютерного зору, що має широке коло застосувань. Не зважаючи на те, що на сьогодні запропоновано чимало різноманітних рішень для вирішення даної задачі, далеко не всі з них спроможні працювати в реальному часі, особливо на мобільних та вбудованих обчислювальних платформах. Одними з методів, які здатні забезпечити високу швидкодію та надійність відслідковування залишаються підходи, основані на дискримінантних кореляційних фільтрах (DCF). Найостанніші реалізації цих підходів використовують ознаки, отримувані з великих і обчислювально-складних нейронних мереж, таких, зокрема, як VGG та ResNet, що з одного боку дозволяє підвищити якість відслідковування, але з іншого – робить такі рішення ресурсоємними та подекуди недопустимо повільними. При цьому застосування ознак із легких і швидкодіючих нейронних мереж у системах відслідковування на базі DCF практично не досліджені. Ця робота якраз і присвячена даному питанню. Із застосуванням базової системи відслідковування на основі дискримінантного кореляційного фільтра (DCF), що в якості оптимізаційної процедури використовує метод множників зі зміною напрямків (ADMM) проведено аналіз ознак, отримуваних із ряду швидкодіючих нейронних мереж, серед яких: SqueezeNet, MobileNetV3, ShuffleNet-9, а також малих моделей детекторів YOLO. При цьому обґрунтовано вибір шарів із яких доцільно отримувати ознаки та за методикою тесту VOT Challenge 2019 оцінено точність і надійність відслідковування, а також на мобільному ПК та двох популярних вбудованих обчислювальних платформах (Nvidia Jetson Nano та Raspberry Pi 5) визначено середню швидкодію обробки кадрів при використанні ознак кожної з зазначених мереж. В результаті проведеного дослідження встановлено, що найбільш доцільними у використанні з огляду на компроміс між швидкодією та якістю відслідковування є ознаки отримувані із мереж YOLOv3 tiny та SqueezeNet.

Ключові слова: візуальне відслідковування об'єктів; нейромережні ознаки; дискримінантні кореляційні фільтри (DCF); метод множників зі зміною напрямків (ADMM)

DOI: 10.20535/RADAP.2025.101.39-50

Вступ

Візуальне відслідковування об'єктів є задачею комп'ютерного зору, що полягає у пошуку заданого об'єкта на кожному кадрі відеопослідовності та має широке коло застосувань, основними серед яких є робототехніка, людино-машинні інтерфейси, медицина, оборонні системи, відеоаналітика тощо.

У більшості вказаних застосувань, відслідковування має виконуватись в реальному часі (з частотою кадрів щонайменше 15-20 Гц) та досить часто на системах з обмеженими обчислювальними ресурсами такими, як одноплатні комп'ютери чи мобільні пристрої.

Більшість розроблених останнім часом методів відслідковування основані на нейронних мережах та спроможні працювати в реальному часі на найбільш потужних дискретних графічних прискорювачах [1–4].

Це обумовлено тим, що у них в якості ознак використовуються тензори вилучені з проміжних шарів глибоких і відносно складних у обчисленні нейронних мереж таких, наприклад, як ResNet [5], або ж весь метод відслідковування є спеціалізованою нейронною мережею, основаною на популярних нині моделях уваги [6] та візуальних трансформерах [7,8], які також не є простими і швидкими в обчисленні.

Варто зауважити, що існують також і оптимізовані для роботи в реальному часі нейромережні методи відслідковування, однак вони здатні швидко працювати лише або на графічних прискорювачах мобільних пристроїв останніх поколінь [9], або якщо і досягають реального часу обробки кадрів на звичайних процесорах, то як правило мова йде про високопродуктивні чи взагалі серверні рішення [6].

Таким чином, дослідження в сфері швидкодіючих методів відслідковування, що здатні працювати на мобільних та вбудованих обчислювальних платформах досі залишається актуальною задачею. Саме цьому питанню і присвячена дана публікація.

Так, з огляду на здатність забезпечувати компромісну якість та швидкодію у даній роботі розглядається метод відслідковування на основі дискримінантного кореляційного фільтра (discriminative correlation filter, DCF), який в якості ознак використовує ознаки, обчислені за допомогою швидкодіючих нейронних мереж, таких як SqueezeNet [10], MobileNet [11], ShuffleNet [12], а також моделей YOLO [13, 14].

Головною метою при цьому є дослідження та порівняння здатності зазначених нейронних мереж генерувати ознаки, які для DCF підходу забезпечували б найкращу якість відслідковування та найвищу швидкодію. Порівняння якості відслідковування здійснено за методикою VOT Challenge 2019 [15], а швидкодія оцінена на мобільному ПК та на популярних одноплатних комп'ютерах з процесорними ядрами ARM у вигляді середнього часу на обробку кадру.

Загальна структура статті є такою: у першому розділі розглянуто версію дискримінантного кореляційного фільтра (DCF) з просторовою регуляризацією, що застосовувався для досліджень у даній роботі, у наступному розділі наведено короткі відомості про нейронні мережі, що використовувались в якості генераторів ознак для DCF та аргументацію щодо вибору шарів, з яких було отримано тензори ознак, в останньому розділі наведено результати щодо якості відслідковування та швидкодії, а також обговорено отримані результати.

Зауважимо, що хоча і у дослідженнях використовувалась конкретна версія дискримінантного кореляційного фільтра (DCF), а саме: фільтр з просторовою регуляризацією, який обчислюється за допомогою метода оптимізації ADMM (alternating direction method of multipliers) [16], отримані в даній роботі результати можуть бути цілком успішно застосовані і до інших систем відслідковування, що базуються на використанні DCF фільтрів [3, 17, 18, 20, 21].

1 Дискримінантний кореляційний фільтр з просторовою регуляризацією для візуального відслідковування об'єктів

Фільтр DCF є спеціалізованим цифровим фільтром, в якому кодується зовнішній вигляд об'єкта, що необхідно шукати на кожному кадрі відеопослідовності. Цей фільтр має бути сформований таким чином, щоб при його застосуванні до зображення було отримано відгук, в якому містився б чіткий пік саме в тій області, яка є найбільш подібною до об'єкта закодованого в фільтрі. Обробка може здійснюватись у будь-якому просторі ознак – в найпростішому випадку це може бути звичайна яскравість зображення [22], однак в більш сучасних рішеннях застосовуються багатоканальні ознаки HOG чи ColorNames [19, 23] або відгуки певного шару нейронних мереж [3, 17, 18, 20]. Застосування дискримінантного кореляційного фільтра (DCF) для відслідковування об'єктів передбачає таку послідовність дій:

1) на першому кадрі відеопослідовності задається область розташування об'єкта, для якої формується початковий DCF фільтр;
2) на наступному кадрі виконується локалізація об'єкта, для чого обчислюється кореляція (або згортка) фільтра з областю пошуку та в отриманому відгуку визначаються координати максимального значення (в якості області пошуку зазвичай обирається фрагмент кадру центрований на минулому розташуванні об'єкта);
3) область знайденого нового положення об'єкта використовується для оновлення фільтра;
4) кроки 2-3 повторюються для кожного кадру відеопослідовності.

В одному з варіантів DCF, що розглядатиметься у даній роботі, процедура отримання фільтра полягає у вирішенні такої мінімізаційної задачі:

$$\arg\min_{h} \sum_{i=1}^{d} \|t_i * h_i - r\|^2 + \lambda \|h\|^2, \qquad (1)$$

де $\|\cdot\|^2$ – позначає суму квадратів елементів матриці (квадрат норми Фробеніуса); "$*$" – позначає згортку між i-тим каналом фільтра h_i та шаблону t_i, з якого формується даний фільтр; r – бажаний відгук, у ролі якого зазвичай виступає Гаусіан з малим параметром дисперсії [19, 22, 23]; шаблон t та фільтр h вважаються багатоканальними зображеннями (тривимірними масивами), r – матрицею (двовимірним масивом), d позначає кількість каналів, а індекс i – номер каналу; λ – параметр регуляризації.

Класичне аналітичне рішення регресійної задачі (1) має вигляд [23]:

$$\bar{h} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\bar{r}, \tag{2}$$

де $\bar{h}$ та $\bar{r}$ – фільтр та бажаний відгук витягнуті у вектор-стовпчики; $\mathbf{X}$ – матриця сформована з шаблону t таким чином, що кожен її рядок є векторизованим фрагментом t, на який при обчисленні значення згортки має накладатися фільтр h, іншими словами $\mathbf{X}\bar{h}-\bar{r}$ є витягнутим у вектор результатом, що формується під першим квадратом норми Фробеніуса у виразі (1); $\mathbf{I}$ – одинична матриця; оператори $(\cdot)^T$ та $(\cdot)^{-1}$ позначають транспонування та обернення матриць.

Рішення задачі (1) у формі (2) є досить обчислювально-витратним і має асимптотичну складність $\mathcal{O}(N^3)$, де N – це кількість елементів у зображенні шаблону t, тому на практиці воно не застосовується. Натомість рішення (1) знаходять у частотній області [19,22,23], що завдяки застосуванню швидкого перетворення Фур'є та теореми про згортку дозволяє досягти асимптотичної складності вже на рівні $\mathcal{O}(N \cdot \log(N))$ операцій. Це рішення має вигляд [19]:

$$H_i = \frac{T_i \cdot R^*}{\sum_{j=1}^{d} T_j \cdot T_j^* + \lambda}, \tag{3}$$

де H_i та T_i – i-тий канал фільтра h_i та шаблону t_i у частотній області відповідно; R – відгук у частотній області; операція $(\cdot)^*$ – позначає комплексне спряження; операції ділення та множення є комплексними і виконуються поелементно; λ – скалярний параметр регуляризації; шаблон T та фільтр H у частотній області отримані шляхом застосування двовимірного перетворення Фур'є до кожного з каналів шаблону t та фільтра h, представлених у просторовій області.

Водночас, пізніше було встановлено, що обчислення фільтра в частотній області за допомогою (3) має важливий недолік: через періодичність та нескінченність функцій (синусів і косинусів), по яким здійснюється розклад у перетворенні Фур'є, принципово неможливо отримати фільтр, який у просторовій області містив би нульові значення у чітко визначених позиціях, зокрема скрізь за межами тієї частини, що кодує об'єкт [17, 25]. Це призводить до того, що у фільтр потрапляє фонова інформація, причому при збільшенні фільтра збільшуватиметься і обсяг цієї інформації. Як наслідок область кадру, яка може бути використана для обчислення фільтра обмежується, що з одного боку знижує дискримінантну спроможність фільтра (спроможність фільтра розрізняти фон та об'єкт), а з іншого – зменшується область пошуку, що не дозволяє відслідковувати об'єкти, які швидко рухаються.

Щоб подолати зазначений недолік в ряді робіт запропоновано внести додаткові обмеження в оригінальну оптимізаційну задачу. Ці обмеження передбачають або вирізання за допомогою маскуючої матриці непотрібної інформації із фільтра [25, 26], або заміну скалярного параметру регуляризації на регуляризаційну матрицю, яка придушує небажані елементи фільтра в заданих позиціях [17,18,20,24]. При цьому для вирішення самої оптимізаційної задачі найпоширенішими підходами є метод спряжених градієнтів [3,18,24] та метод множників Лагранжа, що змінюють напрямки (ADMM) [20,21,25,26]. У підходах, де використовується метод спряжених градієнтів [3,17,18,24] є можливість під час оптимізації додатково врахувати області розташування об'єкта, знайдені на попередніх кадрах, однак загалом для пошуку DCF фільтра метод спряжених градієнтів є дещо повільнішим за ADMM, а для ефективного застосування він ще й потребує виконання певних вимог щодо вигляду матриці регуляризації (в частотній області вона повинна представлятись в компактному вигляді). Таким чином, в даній роботі для пошуку DCF фільтра будемо використовувати метод ADMM.

Оптимізаційна задача в такому випадку матиме вигляд:

$$\begin{gathered}\arg\min_{h} \|\sum_{i=1}^{d} t_i * h_i - r\|^2 + \sum_{i=1}^{d} \|w \cdot g_i\|^2, \\ \text{s.t. } h_i - g_i = \mathbf{0}, \forall i = 1, ..., d\end{gathered} \tag{4}$$

де w – матриця регуляризації, за допомогою якої придушуються непотрібні складові фільтра; g – це інша змінна, що також задає фільтр, причому оптимізаційне обмеження вимагає, щоб $g = h$; оператор "$\cdot$" позначає поелементне множення між w та g_i; інші змінні та оператори є аналогічними формулі (1).

Для вирішення задачі (4) використаємо масштабовану форму методу ADMM [16], яка передбачає ітеративне вирішення таких оптимізаційних підзадач:

$$\begin{aligned}
h^{(k+1)} &= \arg\min_{h} \|\sum_{i=1}^{d} t_i * h_i - r\|^2 + \frac{\rho}{2}\sum_{i=1}^{d} \|h_i - g_i^{(k)} + u_i^{(k)}\|^2, \\
g^{(k+1)} &= \arg\min_{g} \sum_{i=1}^{d} \left(\|w \cdot g_i\|^2 + \frac{\rho}{2}\|h_i^{(k+1)} - g_i + u_i^{(k)}\|^2\right), \\
u^{(k+1)} &= u^{(k)} + \left(h^{(k+1)} - g^{(k+1)}\right),
\end{aligned} \tag{5}$$

де ρ – штрафний параметр; u – масштабована двоїста змінна, пропорційна множникам Лагранжа [16]; показник $(\cdot)^{(k)}$ визначає значення відповідних змінних на k-ій ітерації методу ADMM.

Стисло розглянемо розв'язок кожної з підзадач визначених у формулах (5).

Підзадача для змінної h. Через наявність згортки у даній підзадачі, її зручніше вирішувати у частотній області. При цьому згідно з теоремою Парсеваля та теоремою про згортку можна зробити заміну квадратів норм на квадрати модулів, а згортку на

поелементне множення, зокрема:

$$H^{(k+1)}=\arg\min_{H}\sum_{x}\sum_{y}|\sum_{i=1}^{d}T_i(x,y)\cdot H_i(x,y)-R(x,y)|^2+ \\ +\frac{\rho}{2}\sum_{x}\sum_{y}\sum_{i=1}^{d}|H_i(x,y)-G_i^{(k)}(x,y)+U_i^{(k)}(x,y)|^2 \tag{6}$$

де суми по x та y рахуються по всім елементам, які передбачали норми у початковому виразі, а всі дії множення, додавання та віднімання виконуються над скалярними комплексними значеннями елементів (x,y) відповідних матриць.

Пошук мінімуму у (6) можна виконувати незалежно по кожному елементу (x,y) матриць окремо. Для цього, враховуючи, що змінні у виразі є комплексними слід скористатись похідними Вітінгера – диференціювання можна здійснити по комплексно-спряженій змінній $H_i^*(x,y)$, а отриману систему рівнянь вирішувати відносно звичайної змінної $H_i(x,y)$ [22]. Зауважимо, що необхідність вирішення саме системи рівнянь виникає через наявність підсумовування по каналам. Кожна система при цьому міститиме d рівнянь, а самих систем буде стільки, скільки є комбінацій (x,y). На щастя рішення цих систем можна спростити за допомогою формули Шермана-Морісона [20,26], що в кінцевому рахунку дає наступне рішення оптимізаційної підзадачі для змінної H у частотній області:

$$H_i^{(k+1)}=[\tfrac{\rho}{2}RT_i^*+(G_i^{(k)}-U_i^{(k)})]- \\ -T_i^*\cdot\frac{\sum_{j=1}^{d}\left(T_j\cdot[\frac{\rho}{2}RT_j^*+(G_j^{(k)}-U_j^{(k)})]\right)}{\sum_{j=1}^{d}T_jT_j^*+\frac{\rho}{2}}, \tag{7}$$

де H, G, T, R, та U – фільтри h та g, шаблон t, бажаний відгук r та масштабована двоїста змінна u переведені в частотну область; T^* позначає комплексне-спряження шаблону t у частотній області; всі операції множення і ділення виконуються поелементно.

Підзадача для змінної g. Через наявність поелементного множення $w\cdot g_i$, на відміну від попередньої, цю підзадачу краще вирішувати в просторовій області. Для цього норми можна розписати у вигляді сум квадратів елементів матриць. Враховуючи, що всі операції над матрицями при цьому виконуватимуться поелементно, диференціювання можна проводити незалежно по кожному елементу окремо, а результат мінімізації матиме вигляд:

$$g_i^{(k+1)}=\frac{\rho\left(h_i^{(k+1)}+u_i^{(k)}\right)}{2w^2+\rho}. \tag{8}$$

Оскільки пошук h здійснюється в частотній області, а значення g з міркувань подальшого швидкого обчислення згортки доцільніше мати в частотній області, то і вираз (8) краще представити в частотній області:

$$G_i^{(k+1)}=\mathcal{F}\left[\mathcal{F}^{-1}\left[H_i^{(k+1)}+U_i^{(k)}\right]\cdot\frac{\frac{\rho}{2}}{w^2+\frac{\rho}{2}}\right], \tag{9}$$

де $\mathcal{F}[\cdot]$ та $\mathcal{F}^{-1}[\cdot]$ – позначає пряме та зворотне двовимірне перетворення Фур'є відповідно.

Оновлення масштабованої двоїстої змінної u. Оскільки у виразі для оновлення змінної u містяться тільки операції додавання та віднімання масивів, його обчислення може виконуватись безпосередньо в частотній області:

$$U^{(k+1)}=U^{(k)}+\left(H^{(k+1)}-G^{(k+1)}\right). \tag{10}$$

Для пришвидшення збіжності оптимізаційних процедур у ряді робіт [20, 21, 26] пропонується застосовувати адаптивну зміну штрафного параметру ρ (розмір кроку). Таке оновлення здійснюється на кожній ітерації алгоритму за формулою [26]:

$$\rho^{(k+1)}=\min\left(\beta\cdot\rho^{(k)},\rho_{\max}\right), \tag{11}$$

де β – коефіцієнт зміни штрафного параметра, а $\rho_{\max}$ – максимально допустиме значення ρ.

Водночас, слід врахувати, що при застосуванні методу ADMM у масштабованій формі (як у цій роботі), оновлення параметру ρ має супроводжуватись обернено пропорційною зміною масштабованої двоїстої змінної U, тобто при збільшенні ρ в β раз, U має бути зменшено відповідно, тобто: $\frac{U}{\beta}$ [16].

Після виконання необхідної кількості ітерацій методу ADMM, результат формується як рішення підзадачі g, тобто фільтр, пошук якого здійснюється, міститиметься у змінній $G^{(k+1)}$ на останній ітерації.

Пошук об'єкта за допомогою знайденого DCF фільтра. Після того, як DCF фільтр розраховано, з його допомогою пошук об'єкта на зображенні F виконується шляхом обчислення згортки у частотній області з подальшим переходом у просторову область:

$$c=\sum_{i=1}^{d}\mathcal{F}^{-1}\left[F_i\cdot G_i\right]=\mathcal{F}^{-1}\left[\sum_{i=1}^{d}F_i\cdot G_i\right], \tag{12}$$

де $\mathcal{F}^{-1}[\cdot]$ – зворотне двовимірне перетворення Фур'є; c – відгук фільтра (згідно з постановкою задачі (1) або (4)); F_i – i-й канал ознак області пошуку об'єкта в частотній області; G_i – i-й канал дискримінантного фільтра в частотній області; операція множення є поелементною.

Положення об'єкта визначається шляхом пошуку максимальної точки у відгуку c (за умови, що у якості бажаного відгуку r в оптимізаційній задачі була обрана функція також з чітко вираженим піком).

Оновлення фільтра. Оскільки під час відслідковування об'єкт може змінювати зовнішній вигляд, на кожному кадрі виконується оновлення DCF фільтра. Для цього, після визначення найбільш імовірного

положення об'єкта на поточному кадрі, його окіл розглядається як новий шаблон, за допомогою якого розраховується фільтр G для даного кадру, а далі виконується експоненціальне усереднення з фільтром, що використовувався на попередньому кадрі:

$$G^{(\tau+1)} = \alpha \cdot G + (1-\alpha) \cdot G^{(\tau)}, \qquad (13)$$

де $G^{(\tau+1)}$ та $G^{(\tau)}$ – усереднені фільтри, що використовуватиметься та використовувався для пошуку об'єкта на наступному $(\tau+1)$ і поточному (τ) кадрах відповідно; G – фільтр обчислений для околу об'єкта після визначення його положення на поточному кадрі; α – коефіцієнт експоненціальної фільтрації.

Пошук та пристосування до масштабу об'єкта. Для більш надійного відслідковування об'єктів, що змінюють розмір також використовується процедура оцінки масштабу [17, 19–21, 26]. В найпростішому випадку її суть зводиться до того, що фрагмент кадру, де здійснюється пошук об'єкта масштабується декілька разів як в сторону збільшення, так і в сторону зменшення (будується так звана піраміда зображень), а далі фільтр застосовується до кожного з масштабів. Масштаб, на якому буде отримано найвищий кореляційний відгук, вважається найбільш близьким до істинного розміру об'єкта, а обчислення розміру рамки навколо об'єкта та нового фільтра G для поточного кадру виконується з урахуванням цього масштабу. Для підвищення точності оцінки масштабу також може використовуватись інтерполяція.

У більш складних варіантах можуть застосовуватись додаткові нейронні мережі, зокрема IoUNet, що окремо оцінюють розмір та форму об'єкта у [3], процедура Alpha-refine [27], яка уточнює положення, розмір та форму об'єкта за допомогою сіамської нейронної мережі, або навіть сегментування – у найбільш останніх підходах.

З огляду на більш просту реалізацію, в даній роботі зупинимось на варіанті оцінювання масштабу на основі побудови піраміди зображень.

2 Ознаки

З початком використання згорткових нейронних мереж як генераторів ознак для систем відслідковування на основі DCF, в більшості робіт застосовувались ресурсоємні в обчисленні моделі такі, як VGG-2048/VGG-M [17, 18, 24], ResNet [3, 28] та дещо менш складна AlexNet [29]. З одного боку це дозволило підвищити якість відслідковування, але призвело до помітного зниження швидкодії, яка подекуди зменшилась до одиниць кадрів на секунду. Щоб досягти компромісу між швидкодією та надійністю відслідковування, в даній роботі пропонується, в якості мереж для отримання ознак дослідити нейронні архітектури, призначені для швидкого обчислення, такі як SqueezeNet, MobileNet, ShuffleNet та tiny версії детекторів YOLO.

Важливими аспектами при застосуванні у DCF фільтрі нейронних ознак є по-перше: визначення з якого саме шару(ів) нейронної мережі братимуться тензори ознак, по-друге: якої роздільної здатності мають бути вхідне зображення та вихідний тензор(и).

Вибір шару, з якого мають бути отримані ознаки виконується в даній роботі зокрема з таких міркувань. На Рис. 1 показано типову структуру тензорів ознак, що формуються при проходженні даних через згорткову нейронну мережу.

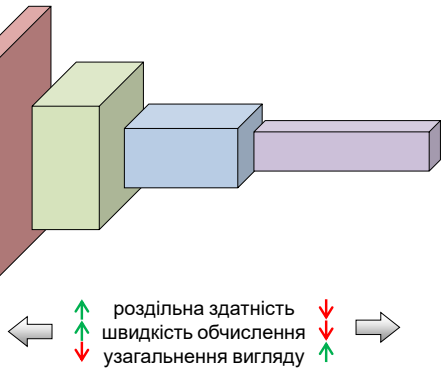

Fig. 1. Типова структура масивів (тензорів), що формуються на виходах шарів згорткової нейронної мережі: масиви ближчі до входу мають меншу кількість каналів (вужчі), але вищу роздільну здатність, тоді як масиви, що формуються більш глибокими шарами навпаки мають меншу роздільну здатність та більшу кількість каналів (є ширшими)

В ранніх шарах мережі утворюються масиви меншої глибини (мають меншу кількість каналів), проте вищої роздільної здатності (мають більшу ширину та висоту). Вважається, що ознаки на виході більш ранніх шарів мають меншу специфічність та інваріантність у представленні об'єктів і тому гірше узагальнюють зовнішній вигляд [30]. Обчислення ознак отриманих із ранніх шарів із очевидних причин є простішим.

Ознаки, отримані з більш глибоких шарів, навпаки, мають більшу кількість каналів та нижчу роздільну здатність (меншу ширину та висоту масиву). Ці ознаки є більш специфічними в сенсі того, що забезпечують краще представлення певних об'єктів (тих на яких навчалася мережа), однак при цьому мають вищу інваріантність завдяки кращому узагальненню зовнішнього вигляду. Для кращої ілюстрації підтвердження даного факту наведемо такий приклад: якщо у мережі для розпізнавання об'єктів в екстремальному випадку в якості найбільш глибокого шару розглядати безпосередньо її вихід, то можна вважати, що на ньому формується масив розміром $1 \times 1 \times C$ елементів, де C – це кількість класів, яку розпізнає дана мережа. Цей масив має роздільну здатність лише 1×1 та видає по третій розмірності дуже узагальнену

інформацію – імовірність приналежності об'єкта до певного класу, забезпечуючи при цьому максимальну (яку тільки може забезпечити дана мережа) інваріантність до зміни форми об'єкта, його зовнішнього вигляду та положення на вхідному зображенні.

При вирішенні задачі відслідковування, висока інваріантність ознак, отримуваних з більш глибоких шарів мережі начебто і є гарною властивістю, оскільки дозволяє забезпечити надійнішу роботу при поворотах об'єкта, зміні ракурсу камери, освітлення тощо. Але з іншого боку, якщо ознаки надмірно узагальнюватимуть зовнішній вигляд, це приводитиме до ускладнення *розрізнення схожих* об'єктів, що є небажаним. Крім того, для більш точного визначення просторового положення об'єкта за допомогою DCF фільтра масив ознак повинен мати достатню роздільну здатність, а з міркувань практичного застосування ще й якомога швидше обчислюватись. Таким чином, шар, з якого доцільно отримувати ознаки, має знаходитись десь у середині мережі, можливо навіть ближче до її входу.

З огляду на сказане вище, а також для повноти викладення, розглянемо особливості побудови обраних швидкодіючих нейронних мереж (SqueezeNet, MobileNet, ShuffleNet та tiny версії детекторів YOLO), обґрунтовуючи вибір шарів, з яких було б доцільно отримувати ознаки в контексті задачі відслідковування, що розглядається.

Мережа SqueezeNet. Основною особливістю цієї мережі є використання модулів Fire [10]. Дані модулі мають на вході стискаючий згортковий шар $s_{1\times1}$ з невеликою кількістю фільтрів розміром 1×1, які зменшують глибину вхідного тензора. За стискаючим шаром слідує шар активації ReLU, вихід якого паралельно подається на два згорткові шари розширення $e_{1\times1}$ та $e_{3\times3}$ з фільтрами розміром 1×1 та 3×3 відповідно. Виходи шарів розширення з'єднуються з шарами ReLU і далі конкатенуються, формуючи фінальний вихід модуля (Рис. 2). Використання стискаючого шару дозволяє зменшити глибину фільтрів у шарах розширення, що як наслідок призводить до зниження загальної кількості параметрів та обчислень.

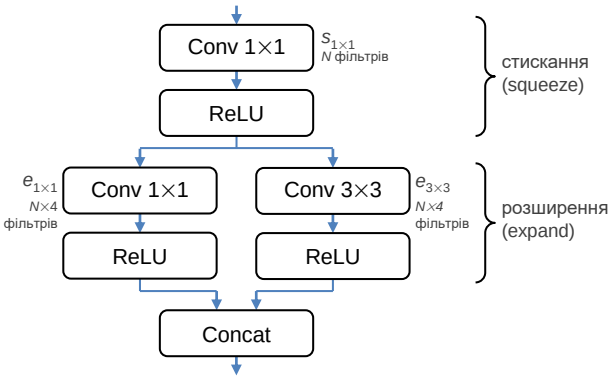

Fig. 2. Структура модуля Fire мережі SqueezeNet

Для застосування у задачі відслідковування, що розглядається, найбільш компромісними з огляду на швидкодію та узагальнюючу спроможність є ознаки, отримувані на виходах модулів fire2-fire3 мережі [10], при цьому особливий інтерес представляють виходи стискаючих шарів $s_{1\times1}$, оскільки вони значно зменшують глибину тензорів ознак.

Мережа MobileNet. Для підвищення швидкодії в мережі MobileNet використовуються згорткові шари розділювані по глибині (depthwise separable convolution), в основі яких лежить припущення щодо можливості застосування властивості асоціативності згортки для підвищення обчислень без суттєвої втрати якості розпізнавання. Завдяки цій властивості звичайну повну згортку замінюють на дві послідовні згортки з фільтрами меншого розміру, які давали б на виході тензор того ж розміру, що й звичайна згортка. А саме, звичайний згортковий шар з N фільтрами розміром $D \times D \times M$ замінюється на розділюваний по глибині шар, в якому спочатку використовуються M фільтрів $D \times D \times 1$, до результату обчислення яких застосовуються N фільтрів $1 \times 1 \times M$. Це дозволяє знизити кількість параметрів, що підлягають налаштуванню та, як наслідок – загальний обсяг обчислень. Розділюваний по каналам згортковий блок, що реалізує описану вище концепцію показано на Рис. 3.

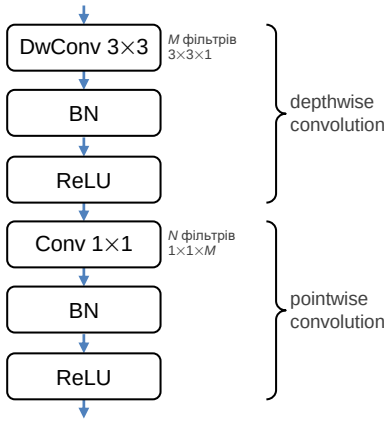

Fig. 3. Структура розділюваного по каналам згорткового блоку: DwConv – шар, що використовує фільтри глибиною 1 і виконує згортку по кожному каналу окремо, його застосування по обчисленням еквівалентно застосуванню лише одного фільтра в звичайному згортковому шарі; BN позначає шар нормалізації по навчальній підвибірці (Batch Normalization)

У мережах MobileNet новіших версій 2 та 3 входи та виходи розділюваного по каналам згорткового блоку зроблені більш звуженими по глибині (bottleneck) – тензори мають меншу кількість каналів. На вході блоку додатково вводиться згортковий шар розширення, що використовує фільтри 1×1 та збільшує глибину тензора перед DwConv шаром, а вихідний згортковий шар налаштований так, щоб приводити розмір вихідного тензора до вхідного. При цьому вхід та вихід

блоку з'єднуються обводом як у ResNet [5], а шари активації з ReLU замінюються на ReLU6. У мережі MobileNet версії 3 окрім цього до вихідного згорткового шару застосовується нормалізація на основі стиснення і збудження (Squeeze-and-Excite) [31].

Найбільш цікавими у застосуванні в якості ознак для системи відслідковування на основі DCF фільтра будуть виходи 2-3 розділюваних по каналам згорткових блоків мереж MobileNet.

Мережа ShuffleNet. Підвищення швидкодії у мережі цього типу досягається за рахунок використання по-канальних згорток та операції групової згортки (group convolution) [12, 32]. Суть цієї операції полягає у тому, що фільтри при обчисленні охоплюють не всі канали вхідного тензору, а певні підмножини каналів – групи. За рахунок цього фільтри мають меншу глибину, а отже меншу кількість параметрів та потребують менше обчислень. Оскільки результати обчислення згорток окремих груп є незалежними між собою, для підвищення взаємного врахування інформації між групами, автори ShuffleNet використовують операцію перемішування по каналам (Shuffle) [12]. У мережі ShuffleNet застосовується звужування аналогічно до того, як це виконується у розділюваному по каналам згортковому блоці мережі MobileNet версії 2 (див. опис вище). Мережа ShuffleNet має два види блоків – з та без усереднюючого об'єднання (Average Pooling). Структура цих блоків показана на Рис. 4.

Зауважимо, що по-канальна згортка DwConv мережі MobileNet може вважатись частинним випадком групової згортки GConv, в якій кожна група складається лише з одного каналу.

У ShuffleNet найбільш доцільними для використання у якості ознак будуть тензори, що формуються на другій ступені мережі, оскільки саме тут вони ще мають достатню для роботи системи відслідковування просторову роздільну здатність (див. структуру мережі у [12]).

Детектори YOLO. Існує ряд версій мереж даного типу. В цій роботі обмежимось розглядом спрощених (tiny та nano) варіантів мереж версій 3 [13] та 5 [14], ознаки з яких братимуться з шарів, що передують детекторним блокам – так званих backbone частин.

У випадку з YOLOv3-tiny частина мережі, що відповідає за генерування ознак (backbone) складається з повторюваних блоків, які містять згортковий шар з фільтрами 3×3, за яким слідує шар нормалізації по навчальній підвибірці (batch normalization), далі функція активації LeakyReLU і шар максимального об'єднання (max pooling) з розміром вікна 2×2 та кроком 2 (окрім передостаннього і останнього блоків). У передостанньому блоці шар максимального об'єднання використовує крок 1, а в останньому блоці шар об'єднання взагалі відсутній. Перший блок використовує 16 згорткових фільтрів, а їх кількість подвоюється у кожному наступному блоці. Всього генератор ознак має 7 блоків (в останньому 1024 згорткові фільтри).

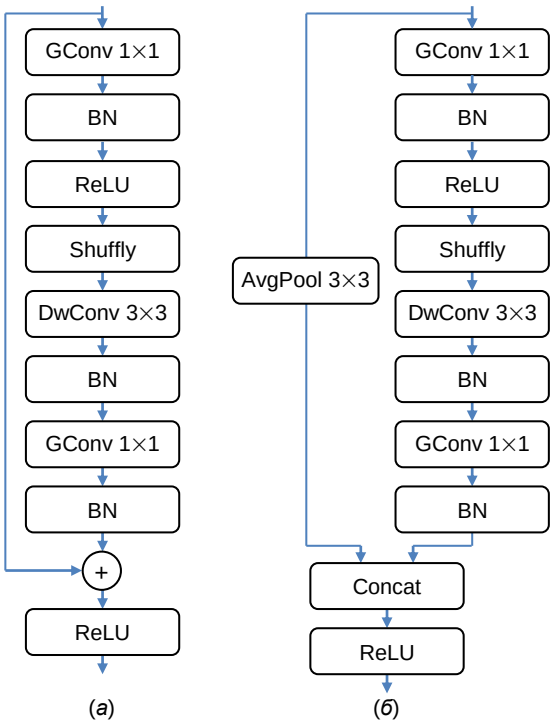

Fig. 4. Структура блоків використовуваних у мережі ShuffleNet: *а* – звичайний блок; *б* – блок з усереднюючим об'єднанням (Avg Pool). GConv – шар групової згортки; Shuffle – шар перемішування (на практиці реалізується шарами зміни розміру та транспонування); DwConv – згортка по глибині; BN – шар нормалізації по навчальній підвибірці (Batch Normalization)

Структура backbone частини мережі YOLOv5n (варіант nano) аналогічна повній версії мережі [14] за виключенням того, що в nano версії використовується менша кількість фільтрів у згорткових шарах. Згортковий модуль у мережі YOLOv5 складається зі згорткового шару, за яким іде нормалізація по навчальній підвибірці (Batch Normalization) та функція активації SiLU (Рис. 5, а). Мережа також містить модулі, що позначаються C3 (Рис. 5, б) і складаються з двох паралельно включених вхідних згорткових модулів, за другим із яких слідує серія блоків звужування (bottleneck, див. Рис. 5, в), причому вихід останнього з цих блоків конкатенується з виходом першого вхідного згорткового модуля і передається на вхід вихідного згорткового модуля.

Іншою відмінністю nano версії від більших варіантів мережі YOLOv5 є використання меншої кількості блоків звужування у модулях C3. Заради повноти викладення слід також зазначити, що мережа YOLOv5 містить модуль швидкого об'єднання просторової піраміди ознак (SPPF), за яким слідує проміжна частина мережі, яку зазвичай називають «neck». У проміжній частині також застосовуються модулі C3, однак у їхніх блоках звужування відсутні обводи на

відміну від того, що показано на Рис. 5. На модулі SPPF та проміжній частині мережі YOLOv5 детальніше зупинятись не будемо, оскільки з огляду на ефективність застосування в задачі відслідковування найбільш цікавими є виходи першого й другого C3 модулів backbone частини, що генерує первинні ознаки (виходи більш глибоких шарів є занадто предметно-специфічними та потребують більше обчислень).

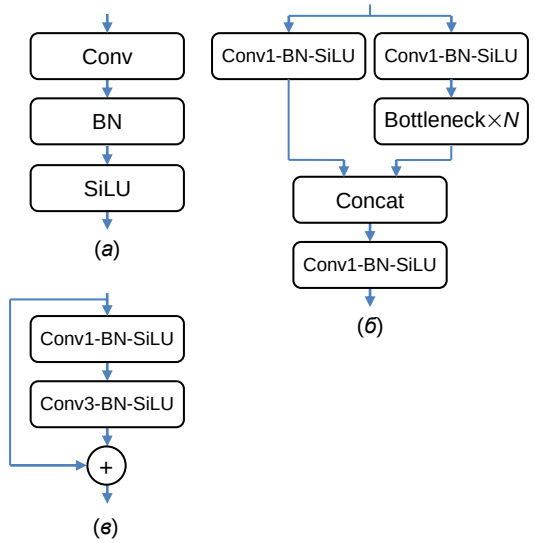

Fig. 5. Структура деяких модулів, що використовуються у мережі YOLOv5: *а* – згортковий модуль; *б* – модуль C3; *в* – блок звужування (bottleneck). Згорткові модулі у блоці звужування та модулі C3 позначені Convn-BN-SiLU (включають згортковий шар, шар нормалізації за навчальною підмножиною та функцію активації SiLU), причому число n одразу за Conv означає розмір фільтрів у відповідних згорткових шарах – число $n = 1$ означає, що використовуються фільтри розміром 1×1, число $n = 3$ – що використовуються фільтри розміром 3×3

3 Експериментальні випробовування

3.1 Налаштування методу оптимізації для отримання DCF фільтра та основні параметри системи відслідковування у експериментах

При проведенні усіх експериментів отримання DCF фільтра здійснювалось шляхом вирішення оптимізаційної задачі (4), для чого виконувались ітерації методу ADMM, кожна з яких передбачає обчислення формул (7)-(11). У якості бажаного відгуку в просторовій області r застосовувався Гаусіан із параметром стандартного відхилення $\sigma = 0,1$.

Матриця регуляризації w, що визначає наскільки сильно придушуються фонові складові фільтра, бралася такою, щоб її центральна частина, яка у просторі ознак має такий самий розмір що й об'єкт, містила б значення близькі до 0 (в експериментах використовувалось значення 0,001) – в цій області складові фільтра відповідають ознакам об'єкта і майже не придушувались. За межами області об'єкта елементи матриці w навпаки бралися із великими значеннями (в експериментах 100), що забезпечувало придушення складових фільтра, які відповідали фону. Для кращої адаптивності при зміні форми об'єкта, а також для збереження у фільтрі незначної кількості контекстно-фонової інформації границя між центральною областю, що містить малі значення та периферійною частиною w, що містить великі значення та відповідає за придушення фонових складових розмивалася фільтром Гауса розміром 7×7 пікселів та стандартним відхиленням $\sigma = 0,5$.

У формулах (7)-(11) застосовувався початковий крок оптимізації $\rho(0) = 1$, коефіцієнтом зміни $\beta = 30$ та максимально допустимим значенням $\rho_{\max} = 1000$. Кількість ітерацій методу ADMM при обчисленні фільтра на кожному кадрі бралася рівною $k_{\max} = 2$. Для підвищення збіжності оптимізаційної процедури, в якості початкового наближення фільтра $G^{(0)}$ та масштабованої двоїстої змінної $U^{(0)}$ на кадрах починаючи з другого бралися значення $G^{(k_{\max})}$ та $U^{(k_{\max})}$ із попереднього кадру.

В усіх експериментах використовувалась квадратна область пошуку збільшена в 4 рази по відношенню до середнього розміру описаного навколо об'єкта прямокутника. Перед переходом до частотної області з метою уникнення ефектів Гібса (розмиття спектру), а також для збільшення імовірності виявлення об'єкта поблизу його попереднього положення до кожного каналу ознак застосовувалось зважуюче вікно фон Хана. Крім того, до кожного каналу ознак застосовувалась нормалізація:

$$\hat{f}_i = \frac{f_i}{\|f_i - \bar{f}_i\|}, \forall i = 1, ..., d, \qquad (14)$$

де f_i – i-тий канал тензору ознак; $\bar{f}_i$ – середнє значення елементів у i-тому каналі тензору ознак; $\|\cdot\|$ – L_2-норма (норма Фробеніуса); d – як і раніше кількість каналів у тензорі ознак.

Оновлення фільтра на кожному кадрі (окрім початкового) виконувалось за формулою (13), в якій коефіцієнт експоненціальної фільтрації α був прийнятий рівним 0,025.

Оцінювання розміру об'єкта здійснювалось шляхом обчислення кореляції (12) на $n_s = 3$ масштабах, з коефіцієнтом масштабування $s = 1,03^p$, де $p = \{-(n_s - 1)/2, \ldots, +(n_s - 1)/2\}$. Масштабом об'єкта на поточному кадрі вважався той, на якому фіксувалось максимальне значення кореляції.

Оскільки зміщення об'єкта оцінювалось у просторі ознак, який має меншу роздільну здатність, ніж область пошуку у реальному зображенні, додатково

виконувалась апроксимація положення кореляційного піку, що дозволило більш точно оцінювати зміщення об'єкта.

3.2 Оцінювання якості роботи методу

Оцінювання якості відслідковування здійснювалось за допомогою тесту VOT Challenge 2019 [15] для короткотривалого відслідковування (short-term tracking). Тести VOT Challenge (починаючи з 2015 по 2022 рік) використовують одні й ті ж міри оцінювання якості при короткотривалому відслідковуванні. Між версіями по роках тести відрізнялися головним чином заміною деяких відеопослідовностей та порогових значень при обчисленні мір якості. При цьому щорічний перегляд тестів здійснювався з метою зміни складності та забезпечення актуальності для наявних на той період часу найновіших систем відслідковування. Оскільки метод, що розглядається в даній роботі належить до класу методів, що розвивались орієнтовно до 2019 року, було прийнято рішення використовувати саме тест 2019 року. При цьому варто зауважити, що так як під час експериментів сам метод відслідковування не змінювався, а використовувалися лише різні ознаки, можна вважати, що вибір варіанту тесту загалом *не повинен* мати значущого впливу на результат.

Для повноти викладення коротко наведемо основні характеристики тесту VOT Challenge 2019. Даний тест містить 60 відеопослідовностей з розміченими на них положеннями різних об'єктів. Розрахунок мір якості оснований на оцінюванні індексу Жакара, який також ще називають «перетином-над-об'єднанням»: $IoU = \frac{|r_t \cap r_{GT}|}{|r_t \cup r_{GT}|}$, де r_t – множина точок, яка задає положення об'єкта, знайдене системою відслідковування, r_{GT} – множина точок положення об'єкта взятого з еталонної розмітки відеопослідовності, $|\cdot|$ – кількість елементів множини. У тесті використовується три міри оцінювання якості:

- *точність* (A) – усереднене по кількості кадрів перекриття (IoU) на тих частинах відео, де відслідковування було визнано надійним;
- *надійність* (R) – величина пропорційна середній кількості зривів відслідковування (вважається, що зрив відбувся тоді, коли перекриття стає меншим за певний поріг);
- *очікуване усереднене перекриття* (EAO) – усереднене значення IoU, очікуване для даної системи відслідковування на великій кількості фрагментів відеопослідовностей приведених до певної однакової довжини [15].

Передбачається, що величина EAO інтегрує в собі перші дві міри (A та R), тому результати тесту впорядковуються саме за нею.

Результати оцінювання системи відслідковування на основі DCF фільтра, що використовує описану вище оптимізаційну процедуру ADMM з просторовою регуляризацією та ознаками отриманими з шарів нейронних мереж, розглянутих у розділі 2, наведено у Табл. 1. Всі тести проводились із програмною реалізацією системи відслідковування, що використовує бібліотеку OpenCV версії 4.9.0, в тому числі й для обчислення нейромережних ознак.

Як видно з Табл. 1 найкращий результат як по точності, так і по надійності відслідковування ($EAO = 0,1692$, $R = 154$) досягається при використанні ознак мережі YOLOv3 отриманих з шару `conv_4`. Окремо варто звернути увагу на можливість використання шару `conv_2` даної мережі. Ознаки з цього шару дають показники якості та надійності відслідковування вищі за середні (Табл. 1), однак при цьому дозволяють подавати на вхід мережі дуже мале зображення – лише 64×64 пікселі, що в результаті забезпечує найвищу серед протестованих мереж швидкодію (див. Табл. 2).

Наступними за показниками надійності та якості є ознаки мережі SqueezeNet. Тут цікавим є той факт, що згідно з експериментом як з огляду на якісні показники, так і на швидкодію (див. Табл. 2) кращим варіантом є використання ознак із шарів, в яких виконується ущільнення тензорів ознак, тобто шарів `squeeze1x1` (див. Рис. 2). Глибший шар `fire3` дає дещо кращі показники надійності R, однак забезпечує нижче очікуване усереднене перекриття EAO у порівнянні з шаром `fire2`.

Найгірші показники забезпечуються подібними між собою мережами MobileNetV3 та ShufflyNet-9. Імовірною причиною цього є структура даних мереж: у них з ростом глибини (якщо орієнтуватись за блоками шарів) швидко спадає роздільна здатність тензорів ознак. З огляду на це використовувати глибші шари недоцільно, оскільки при цьому значно знижується точність локалізації об'єкта, а менш глибокі шари не забезпечують достатню інваріантність до зміни зовнішнього вигляду об'єкта.

Для порівняння до Таблиці 1 (в останні рядки) також було додано результати відслідковування при використанні запрограмованих вручну ознак HOG та більш складної мережі ResNet-50. Варто зауважити, що ознаки з усіх розглянутих нейронних мереж, окрім мережі MobileNetV3 забезпечують кращі показники параметрів EAO та R, ніж при використанні ознак HOG.

Показники швидкодії наведені у Таблиці 2. З неї можна бачити, що найвища швидкодія на всіх тестових обчислювальних платформах забезпечується при використанні ознак мереж YOLOv3 та SqueezeNet. При цьому варто зауважити, що на платформі NVidia Jetson Nano застосування GPU далеко не завжди гарантує підвищення швидкодії, а інколи навіть

Table 1 Результати оцінювання на тесті VOT Challenge 2019. Позначення у таблиці: *шар* – назва шару у мережі, з якого брався тензор ознак для обробки DCF фільтром; *вх. зобр.* – роздільна здатність вхідного зображення у пікселях; *тензор ознак* – розмір тензору ознак; EAO, A, та R – міри оцінювання якості VOT Challenge (міра надійності R безпосередньо вказує кількість зривів відслідковування)

Мережа	**Шар**	**Вх. зобр.**	**Тензор ознак**	**EAO↑**	**A↑**	**R↓**
SqueezeNet	fire2_squeeze1x1	137 × 137	34 × 34 × 16	0,1586	0,4516	170
SqueezeNet	fire2_expand3x3	137 × 137	34 × 34 × 64	0,1454	0,4601	176
SqueezeNet	fire3_squeeze1x1	137 × 137	34 × 34 × 16	0,1584	0,4697	165
SqueezeNet	fire3_expand3x3	137 × 137	34 × 34 × 64	0,1435	**0,4790**	174
MobileNetV3	expanded_conv_2/add	224 × 224	28 × 28 × 24	0,1296	0,4345	251
MobileNetV3	expanded_conv_3	224 × 224	28 × 28 × 96	0,1296	0,4131	223
ShuffleNet	gconv1_7_1	224 × 224	28 × 28 × 136	0,1340	0,4463	204
YOLOv3	conv_4	128 × 128	32 × 32 × 64	**0,1692**	0,4724	**154**
YOLOv3	conv_2	64 × 64	32 × 32 × 32	0,1569	0,4682	173
YOLOv5n	mul_22	128 × 128	32 × 32 × 32	0,1448	0,4550	205
Ознаки HOG	-	136 × 136	32 × 32 × 31	0,1337	**0,5021**	221
ResNet-50	res3a	197 × 197	25 × 25 × 512	0,1601	0,4634	**153**

↑ – чим вище, тим краще

↓ – чим нижче, тим краще

червоним, синім та зеленим показані перший, другий та третій найкращий результат у колонці

Table 2 Середній час обробки кадру у мілісекундах на різних обчислювальних платформах (розміри вхідних зображень і тензорів ознак для всіх мереж відповідають розмірам наведеним у Табл. 1)

Мережа	**Шар**	**Мобільний ПК**	**NVidia Jetson Nano**		**Raspberry Pi 5**
		Intel x86, 3,6ГГц	**ARM A57, 1,46ГГц**	**ARM A57, 1,46ГГц+GPU**	**ARM A76, 2,4ГГц**
SqueezeNet	fire2_squeeze1x1	5,04	28,35	32,67	8,27
SqueezeNet	fire2_expand3x3	9,79	48,33	53,45	17,6
SqueezeNet	fire3_squeeze1x1	6,27	36,0	33,87	12,1
SqueezeNet	fire3_expand3x3	10,91	55,45	55,87	20,43
MobileNetV3	expanded_conv_2/add	11,85	48,69	67,81	19,27
MobileNetV3	expanded_conv_3	14,0	61,34	78,1	25,53
ShuffleNet	gconv1_7_1	129,98	174,16	150,17	67,98
YOLOv3	conv_4	10,15	48,46	48,0	15,13
YOLOv3	conv_2	4,24	20,0	27,7	4,73
YOLOv5n	mul_22	8,77	43,05	35,24	10,17
Ознаки HOG	-	2,54	13,39	-	3,12
ResNet-50	res3a	83,8	358,65	149,73	197,69

призводить до її зниження. Це обумовлюється тим, що ознаки в більшості випадків беруться з неглибоких шарів мереж. Кількість обчислень на CPU в такому випадку може бути меншою, ніж накладні витрати пов'язані із обміном даними між CPU та GPU. Коли кількість обчислень навпаки зростає, перевага використання GPU стає більш очевидною, що зокрема стає помітно для мереж ShufflyNet, YOLOv5n та особливо ResNet-50 (в останньому випадку при застосуванні GPU час обробки кадру зменшується більш ніж в 2 рази).

Висновки

В ході проведеного в роботі аналізу особливостей використання ознак швидкодіючих нейронних мереж щодо застосування у системах відслідковування на основі дискримінантних кореляційних фільтрів (DCF) встановлено, що найбільш доцільними з точки зору одночасного досягнення високих показників якості роботи та швидкодії є ознаки, що беруться з неглибоких шарів мережі YOLOv3 tiny (шар `conv_4`) або мережі SqueezeNet (шар `fire2_squeeze1x1`). Ознаки YOLOv3 tiny дають дещо вищі показники надійності та точності, але поступаються у швидкодії використанню ознак, отримуваних з мережі SqueezeNet. В будь-якому разі обидві мережі дозволять отримати якість відслідковування на рівні більш обчислювально-складної ResNet-50. Серед протестованих мереж найменш доцільними у використанні виявились ознаки з MobileNetV3 та ShuffleNet-9 – вони забезпечують найнижчі показники точності та

надійності, а також мають порівняно велику обчислювальну складність.

Не зважаючи на те, що дослідження виконувались для системи відслідковування, в якій дискримінантний фільтр обчислюється за допомогою методу ADMM з просторовою регуляризацією, отримані результати цілком можуть бути застосовані й для систем, що основані на інших методах оптимізації та використовують інші або додаткові способи регуляризації.

Напрямком подальших досліджень можуть бути використання інших методів оптимізації для отримання дискримінантного фільтра, а також аналіз можливостей ознак сучасніших мереж серії YOLO.

References

[1] Ma Z., Wang L., Zhang H., Lu W., Yin J. (2020). RPT: Learning Point Set Representation for Siamese Visual Tracking. *16th European Conference Computer Vision – ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science*, vol. 12539, pp. 653–665. DOI: 10.1007/978-3-030-68238-5_43.

[2] Yang Z., Miao J., Wei Y., Wang W., Wang X., Yang Y. (2024). Scalable Video Object Segmentation with Identification Mechanism. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 46, Iss. 9, pp. 6247–6262. DOI: 10.1109/TPAMI.2024.3383592.

[3] Danelljan M., Bhat G., Khan F. S., Felsberg M. (2019). ATOM: Accurate Tracking by Overlap Maximization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4655–4664. DOI: 10.1109/CVPR.2019.00479.

[4] Bhat G., Danelljan M., Gool L. V., Timofte R. (2019). Learning Discriminative Model Prediction for Tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6181–6190. DOI: 10.1109/ICCV.2019.00628.

[5] He K., Zhang X., Ren S., Sun J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[6] Cui Y., Song T., Wu G., Wang L. (2023). MixFormerV2: Efficient Fully Transformer Tracking. *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, vol. 36, pp. 58736–58751.

[7] Kristan M., Leonardis A., Matas J., Felsberg M. et al (2022). The Tenth Visual Object Tracking VOT2022 Challenge Results. *Computer Vision – ECCV 2022 Workshops. ECCV 2022. Lecture Notes in Computer Science*, vol. 13808, pp. 431–460. DOI: 10.1007/978-3-031-25085-9_25.

[8] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D. et al (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *International Conference on Learning Representations*. DOI: 10.48550/arXiv.2010.11929.

[9] Borsuk V., Vei R., Kupyn O., Martyniuk T., Krashenyi I., Matas J. Fast, Efficient, Accurate and Robust Visual Tracker (2022). *Computer Vision – ECCV 2022. ECCV 2022. Lecture Notes in Computer Science*, vol. 13682, pp. 644–663. DOI: 10.1007/978-3-031-20047-2_37.

[10] Iandola F. N., Han S., Moskewicz M. W., Ashraf K. et al (2016). AlexNet-level Accuracy with 50x Fewer Parameters and <0.5mb Model Size. *arXiv.org*. DOI: 10.48550/arXiv.1602.07360.

[11] Howard A., Sandler M., Chu G., Chen L.-C. et al (2019). Searching for MobileNetV3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[12] Zhang X., Zhou X., Lin M., Sun J. (2018). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856. DOI: 10.1109/CVPR.2018.00716.

[13] Redmon J., Farhadi A. (2018). YOLOv3: An Incremental Improvement. *arXiv.org*. DOI: 10.48550/arXiv.1804.02767.

[14] Jocher G. (2020). Ultralytics YOLOv5. DOI: 10.5281/zenodo.3908559.

[15] Kristan M., Matas J., Leonardis A., Felsberg M. et al (2019). The Seventh Visual Object Tracking VOT2019 Challenge Results. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. DOI: 10.1109/ICCVW.2019.00276.

[16] Boyd S., Parikh N., Chu E, Peleato B., Eckstein J. (2010). Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, Vol. 3, Iss. 1, pp. 1–122. DOI: 10.1561/2200000016.

[17] Danelljan M., Häger G., Khan F., Felsberg M. (2015). Learning spatially regularized correlation filters for visual tracking. *IEEE International Conference on Computer Vision (ICCV)*, pp. 4310–4318. DOI: 10.1109/ICCV.2015.490.

[18] Danelljan M., Bhat G., Khan F.S., Felsberg M. (2017). ECO: Efficient convolution operators for tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6931–6939. DOI: 10.1109/CVPR.2017.733.

[19] Danelljan M., Häger G., Khan F.S., Felsberg M. (2017). Discriminative Scale Space Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, Iss. 8, pp. 1561–1575. DOI: 10.1109/TPAMI.2016.2609928.

[20] Li F., Tian C., Zuo W., Zhang Lei, Yang M.-H. (2018). Learning spatial-temporal regularized correlation filters for visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4904–4913. DOI: 10.1109/CVPR.2018.00515.

[21] Lukežič A., Vojíř T., Zajc L. Č., Matas J., Kristan M. (2017). Discriminative correlation filter tracker with channel and spatial reliability. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6309–6318. DOI: 10.1007/s11263-017-1061-3.

[22] Bolme D. S., Beveridge R. J., Draper B. A., Lui Y. M. (2010) Visual Object Tracking using Adaptive Correlation Filters. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2544–2550. DOI: 10.1109/CVPR.2010.5539960.

[23] Henriques J. F., Caseiro R., Martins P., Batista J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, Vol. 37, Iss. 3, pp. 583–596. DOI: 10.1109/TPAMI.2014.2345390.

[24] Danelljan M., Robinson A., Khan F. S. and Felsberg M. (2016). Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. *14th European Conference on Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, Vol. 9909, pp. 472–488. DOI: 10.1007/978-3-319-46454-1_29.

[25] Galoogahi H. K., Sim T., Lucey S. (2015). Correlation Filters with Limited Boundaries. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4630–4638. DOI: 10.1109/CVPR.2015.7299094.

[26] Galoogahi H. K., Fagg A., Lucey S. (2017). Learning background-aware correlation filters for visual tracking. *IEEE International Conference on Computer Vision (ICCV)*, pp. 1144–1152. DOI: 10.1109/ICCV.2017.129.

[27] Yan B., Zhang X., Wang D., Lu H., Yang X. (2021) Alpha-refine: Boosting tracking performance by precise bounding box estimation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5289–5298. DOI: 10.1109/CVPR46437.2021.00525.

[28] Lukežič A., Matas J., Kristan M. (2020). D3S – A Discriminative Single Shot Segmentation Tracker. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7131–7140. DOI: 10.1109/CVPR42600.2020.00716.

[29] Xu T., Feng Z.-H., Wu X.-J., Kittler J. (2020). An Accelerated Correlation Filter Tracker. *Pattern Recognition*, vol. 102. DOI: 10.1016/j.patcog.2019.107172.

[30] Ma C., Huang J.-B., Yang X., Yang M.-H. (2015). Hierarchical Convolutional Features for Visual Tracking.*2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3074–3082. DOI: 10.1109/ICCV.2015.352.

[31] Hu J., Shen L., Albanie S., Sun G., Wu E. (2018). Squeeze-and-Excitation Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[32] Xie S., Girshick R., Dollár P., Tu Z., He K. (2017). Aggregated Residual Transformations for Deep Neural Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995. DOI: 10.1109/CVPR.2017.634.

Study of the Use of Features Extracted From the Lightweight and Fast Neural Networks in the Visual Object Trackers Based on Discriminative Correlation Filters

Varfolomieiev A. Yu.

Introduction. Visual object tracking is an important computer vision problem that has a wide range of applications. Although there are many tracking methods exist, not all of them can solve the problem in real-time, especially on mobile and embedded platforms. One of the approaches that permit achieving a trade-off between a high processing speed and tracking reliability is based on the discriminative correlation filters (DCF). The latest implementations of this approach use features, extracted from large and computationally intensive neural networks, such as VGG or ResNet. On the one hand, it enabled to improve tracking quality, but on the other, made trackers too computationally demanding and unacceptably slow for some applications. At the same time, features from lightweight and fast neural networks within the DCF-based tracking approach are practically little studied. This paper is partially intended to fill this gap.

Theoretic results. Using the tracker, which is based on the discriminative correlation filter (DCF) that utilizes the alternating direction methods of multipliers (ADMM) as the optimizer, we analyzed the features, extracted from the lightweight convolutional neural networks such as SqueezeNet, MobileNetV3, ShuffleNet-9, and the tiny/nano YOLO detectors. Particularly, we justified the selection of specific layers from which it is the most expedient to extract the features, and using the VOT Challenge 2019 as the benchmark, estimated the tracking robustness and precision. Using the mobile PC and two popular embedded platforms (Nvidia Jetson Nano and Raspberry Pi 5) we also measured the average frame processing time for all the tested features of the mentioned neural networks.

Conclusions. Our study found that the most efficient features, which provide the trade-off between processing speed and tracking quality, are the ones extracted from the YOLOv3 tiny and SqueezeNet neural networks.

Key words: visual object tracking; neural network features; discriminative correlation filters (DCF); alternating direction method of multipliers (ADMM)