

ОСОБЛИВОСТІ РОБОТИ МЕТОДІВ ПОШУКУ ОБЛИЧ НА МОБІЛЬНИХ ПРИСТРОЯХ

Анотація. Все більша кількість застосунків використовує пошук облич безпосередньо на мобільних пристроях. Однак, фокусом сучасних досліджень за напрямком є підвищення якості пошуку, незважаючи на час роботи, що робить ряд цікавих та ефективних алгоритмів незастосовними у практичних сценаріях. У даній роботі розроблено адаптації п'яти «класичних» та нейромережових алгоритмів пошуку облич до виконання на мобільних пристроях. Проаналізовано час та якість роботи алгоритмів в залежності від ряду факторів (швидкості процесора мобільного пристрою, типового сценарію використання, тощо), показано неочевидну залежність часу виконання від змісту зображення. Сподіваємось, що наведені практичні рекомендації щодо використання алгоритмів поширять їх застосовність у мобільних застосунках та підштовхнуть подальший розвиток їх модифікацій.

Ключові слова: пошук облич, час виконання, мобільні пристрої, нейронні мережі, крайові обчислення.

Постановка проблеми. Попередньо було показано значущість швидкої роботи алгоритмів пошуку облич на мобільних пристроях для практичних застосунків [1] таких, як відстежування сонливості водія під час керування, визначення емоцій або системи контролю доступу [2]. Однак, загальним фокусом сучасних робіт є підвищення якості, незважаючи на час роботи, що робить багато новітніх якісних розробок незастосовними у практичних сценаріях.

В даній роботі: 1) проведено адаптацію низки «класичних» та нейромережових алгоритмів пошуку облич для мобільних пристроїв; 2) проаналізовано якість та час виконання таких алгоритмів в залежності від ряду факторів; 3) показано неочевидну закономірність, що час виконання ряду алгоритмів залежить від змісту зображення (кількість облич, чи вони великі або маленькі). Завдяки цьому аналізу, буде зроблено рекомендації щодо того, які алгоритми слід використовувати в залежності від швидкості процесора мобільного пристрою та типового сценарію використання.

Аналіз останніх досліджень і публікацій. Для початку коротко розглянемо вже усталені методи пошуку облич. Першим з таких методів є **каскадний класифікатор облич Хаара** [3], також відомий як метод Віоли-Джонса. Для свого часу цей метод пропонував краще співвідношення якості та часу. Автори запропонували швидкий метод обчислення фільтрів Хаару, на основі яких алгоритм градієнтного бустінгу AdaBoost передбачає належність кожного із регіонів зображення до класів «обличчя» чи «не обличчя». Каскадна архітектура дозволяє витратити більше часу на регіони, схожі на обличчя, та швидко відкидати частини зображення, що не є обличчям. Метод пошуку облич на основі **локальних бінарних шаблонів (LBP)** продовжує ідею Віоли та Джонса, але використовує LBP ознаки, що є ще простішими до обчислення. Як правило, цей алгоритм працює швидше, але дещо гірше попереднього. Загальнодоступні реалізації обох алгоритмів можна знайти в бібліотеці OpenCV [4]. Метод на основі **гістограми орієнтовних градієнтів (HOG)**, початково запропонований в [5] для пошуку людей та пізніше адаптований до пошуку облич. В цьому методі HOG дескриптори поєднуються із SVM класифікатором. Даний алгоритм реалізовано в [6].

Тепер більш детально розглянемо сучасні нейромережеві методи пошуку облич. Multi-task Cascaded Convolutional Networks (**MTCNN**) [7] – це перший із обраних для подальших досліджень методів, що заснований на нейронних мережах. Згідно авторам, кращу якість навчання можна отримати, якщо навчати нейронну мережу відразу декільком задачам: 1) пошук рамок облич; 2) передбачення вірогідності, що кожна з рамок містить обличчя; 3) виділення п'яти ключових точок обличчя, таких як: очі, кінчик носу, кути рота. Також замість однієї нейронної мережі авторами запропоновано 3, які виконуються одна після іншої, а саме: 1) мережа пропозицій (P-Net) – швидка мережа, що опрацьовує подане зображення у декількох масштабах; 2) мережа уточнення (R-Net); 3) мережа виводу (O-Net). Останні дві мережі опрацьовують передбачення, отримані із попереднього кроку, та відфільтровують або уточнюють їх. Для фільтрації передбачень використовується так званий алгоритм подавлення немаксимумів (NMS). Слід зауважити, що для багатьох методів роботи із обличчями передбачається вирівнювання обличчя (коли необхідно повернути зображення так, щоб очі були розташовані горизонтально). Для таких випадків ключових точок, передбачених алгоритмом MTCNN, буде достатньо. А, отже, додатковий час на пошук ключових точок іншими алгоритмами не витратиметься.

BlazeFace [8] – новітня розробка для пошуку облич, єдина з розглянутих, яку було спеціально розроблено для мобільних пристроїв. Авторами було взято за основу відомому архітектурі для пошуку довільних об'єктів SSD [9] із підмережею MobileNetV2 [10]. Незважаючи на те, що MobileNetV2 – це мережа, орієнтована на мобільні пристрої, автори помітили, що ускладнення так званих Bottleneck блоків (основних будівних блоків мережі MobileNetV2) одночасно із зменшенням їх кількості має переваги за відношенням часу до якості. Новий блок було названо BlazeBlock. Наступним кроком було спрощення архітектури SSD саме для пошуку облич, прибравши модулі, що не додавали якості саме для цієї задачі. І на останок було помічено, що алгоритм NMS (який було використано і в MTCNN) дає помітні «стрибки» у передбаченнях рамок обличчя на сусідніх кадрах, що може зменшувати практичну застосовність алгоритму на відео. Для вирішення проблеми, було запропоновано окремий невеличкий модуль регресії, що значно стабілізує передбачення. Окрім облич, даний алгоритм також передбачає ключові точки (у даному випадку це: очі, вуха, центр рота, кінчик носу). Слід зауважити, що незважаючи на значні нововведення даного алгоритму, авторами не було порівняно його із альтернативними методами. Крім того, навчання відбувалось на закритому наборі даних. У даній роботі таке порівняння буде виконано. Також на відміну від розглянутих алгоритмів, BlazeFace працює не із зображеннями довільного розміру, а має дві фіксовані модифікації: для фотографій на передню камеру із роздільною здатністю 128x128 та на задню – 256x256. Авторами було запропоновано саме 2 різні мережі, оскільки для передньої камери типовим досить великі обличчя, а для задньої вони, як правило, малі.

Метою даної роботи є розробка мобільних адаптацій описаних алгоритмів, аналіз якості та часу їх виконання, а також надання практичних рекомендацій щодо їх застосування.

Основна частина. Для проведення експериментів нами було розроблено мобільний застосунок для операційної системи Android для пошуку облич (рис. 1). У меню конфігурації передбачено широкий спектр налаштувань (рис. 1, б). Першим є роздільна здатність зображення, що подається до алгоритму пошуку облич. Тут є і типові значення для виконання алгоритмів: 128x128, 256x256, 480x360, 640x480; й наднизькі: 32x32, 64x64 – що дозволить проаналізувати, чи є алгоритми здатними до роботи із зображеннями із дуже малою деталізацією. Наступним пунктом є конфігурація моделі, де можна обрати алгоритм пошуку облич, та чи хочемо ми отримувати з нього інформацію про ключові точки обличчя.

чові точки (якщо вона наявна). Далі доступна конфігурація діагностики, яка відображається під час виконання алгоритму у таблиці згори (рис. 1, а). Діагностика може відображатися у вигляді середнього та вибіркової середньоквадратичного відхилення, розрахованого за заданий проміжок часу, або ж у вигляді масиву спостережень із окремими значеннями часу роботи алгоритмів.

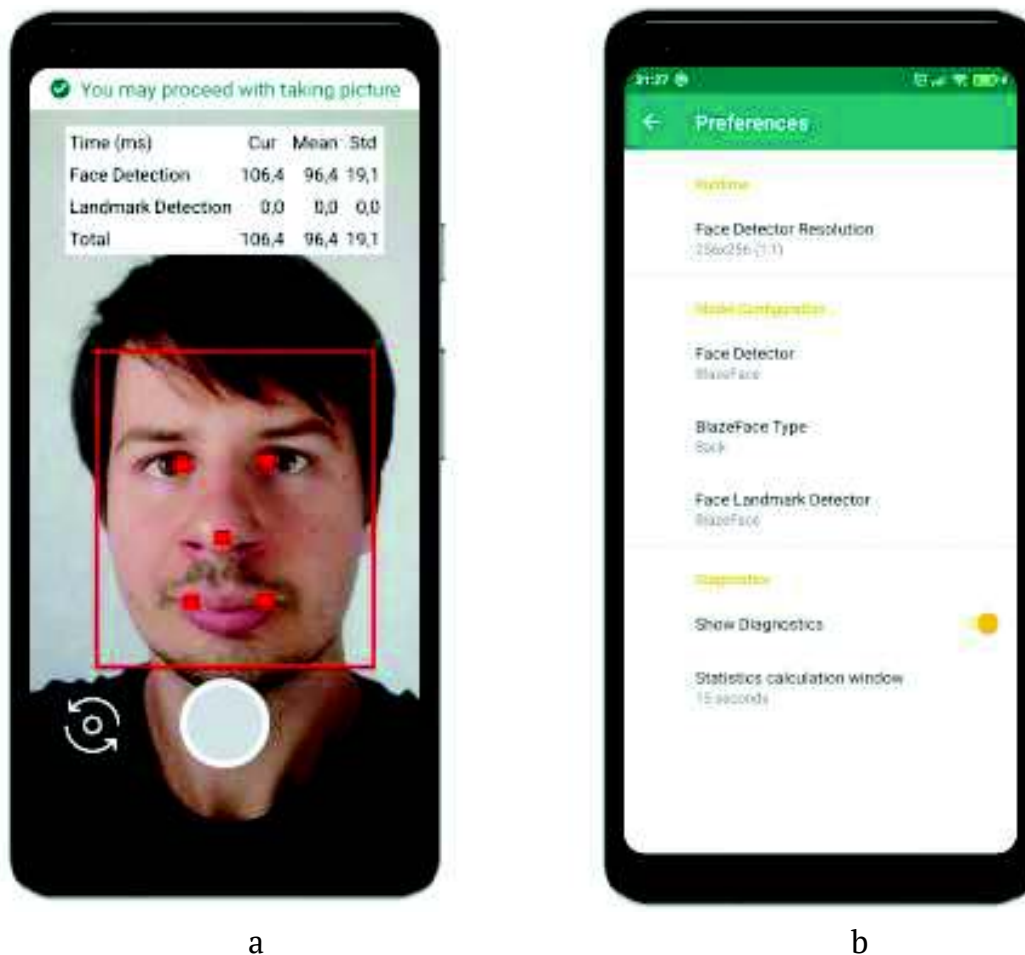


Рисунок 1 – Інтерфейс мобільного застосунку: а – пошук обличчя та ключових точок; б – панель налаштувань

Маючи на увазі операційну систему Android, у якості апаратної платформи нами було відібрано 2 смартфони із процесорами компанії Qualcomm серії Snapdragon. Процесори даної серії широко представлені на ринку, а отже, проведені нами експерименти будуть затребувані та легко відтворювані. Перший процесор – Snapdragon 800 – це флагманський процесор 2013-го року випуску. Враховуючи зріст швидкості роботи процесорів, станом на 2021-й рік такий процесор виступає на рівному із процесорами нижнього цінового сегменту. Другий процесор – Snapdragon 845 – флагманський процесор 2018-го року.

Шляхом проведення експериментів на ньому ми покриваємо сучасний верхній ціновий сегмент.

Кожен із представлених алгоритмів буде оцінено за трьома критеріями: 1) час виконання алгоритму на мобільному пристрою за заданих налаштувань та процесора; 2) успішність пошуку облич на одному із тестових сценаріїв; 3) візуальні особливості передбачень алгоритму, такі як нестабільність («тремтіння») передбачення на відео.

Експерименти. Перший експеримент було проведено з метою оцінки середнього часу виконання алгоритмів для обох процесорів, а також вибіркового середньоквадратичного відхилення часу передбачень. В табл. 1 показано результати експерименту для зображень із роздільною здатністю 256x256. Загалом, на більш старому процесорі алгоритми виконуються у 2,13 – 5,64 разів довше. Цікавість викликає той факт, що відносний зріст часу для нейромережевих алгоритмів є меншим, ніж для «класичних», незважаючи на те, що нейромережеві алгоритми є обчислювально більш складними. Це призвело до того, що, наприклад, каскади Хаара почали виконуватись довше, ніж нейромережеві алгоритми, хоча на Snapdragon 845 каскади Хаара виконувались швидше.

Таблиця 1

Швидкість виконання алгоритмів на мобільних процесорах.

μ – середнє, sd – вибіркве середньоквадратичне відхилення

Алгоритм	Snapdragon 845		Snapdragon 800		Швидше (разів)
	μ (мс)	sd	μ (мс)	sd	
MTCNN	98,9	24,1	210,3	100,4	2,13
BlazeFace	83,0	18,8	253,2	57,4	3,05
LBP	14,5	4,2	72,8	19,0	5,02
HAAR	71,2	7,2	401,8	87,6	5,64
HOG	25,2	4,0	90,6	28,1	3,60

Попередньо було відмічено, що більшість відібраних алгоритмів можуть виконуватись на зображеннях із різною роздільною здатністю. На рис. 2 показано час виконання в залежності від розміру вхідного зображення для процесора Snapdragon 845, а на рис. 3 – для Snapdragon 800. Рожевою горизонтальною лінією позначено орієнтир виконання алгоритмів у реальному часі (25 кадрів на секунду). Зауважимо, що оскільки для BlazeFace є дві моделі із різною архітектурою: для зображень, що сфотографовано на фронтальну камеру і мають

розмір 128x128 (авторами зазначено, що обличчя на таких знімках, зазвичай, досить великі), та для зображень на задню камеру (де обличчя менші) та розміру 256x256, результати між цими двома експериментами інтерполювати не можна, тому заміри на рис. 2, 3 показані окремими точками, а проміжні значення не мають сенсу.

Із рис. 2, 3 можна зробити висновок, що алгоритм на основі каскадів Хаару у всіх сценаріях працює досить повільно. Інші ж алгоритми можуть виконуватись достатньо швидко для зображень 128x128 або у деяких випадках (LBP, HOG) 256x256 на новому процесорі, та лише деякі з них (BlazeFace, LBP, HOG) для зображень не більше 128x128 на старому.

Як було показано, не всі алгоритми є застосовними (за часових обмежень) для зображень великого розміру. Однак, нами не було вказано, як залежить якість розпізнавання від розміру зображення. Оскільки алгоритми, що представлено в даній роботі, було навчено на різних наборах даних, тестування буде проведено для нашого цільового сценарію, а саме мобільного застосування у системі контролю доступу [2].

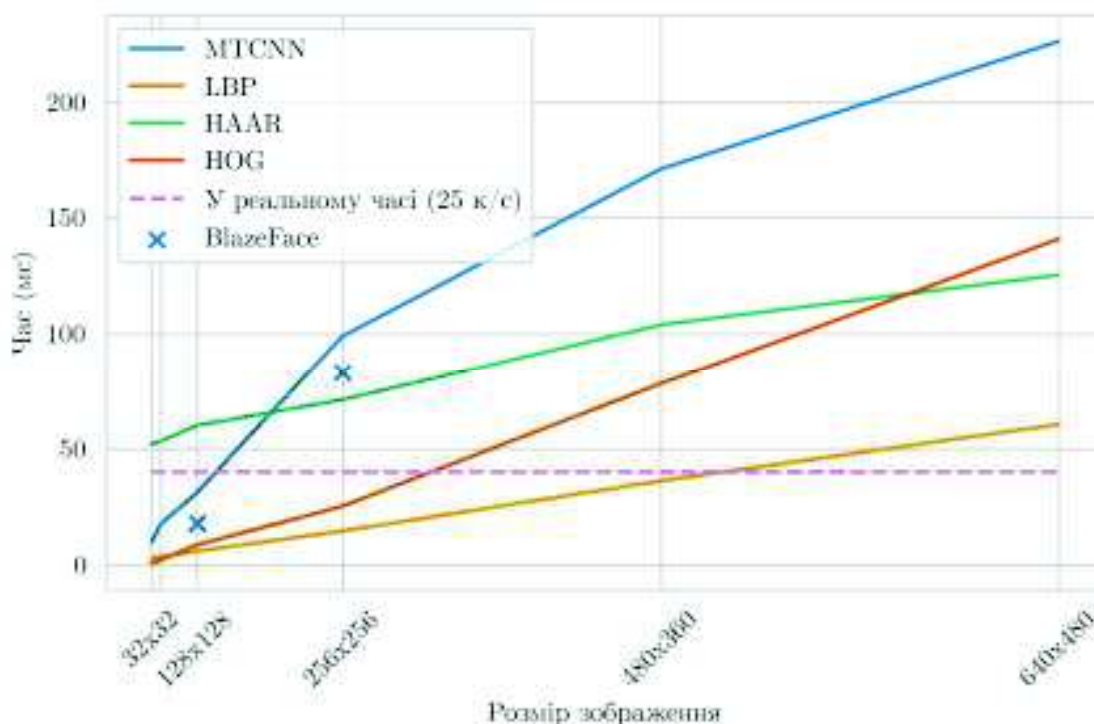


Рисунок 2 – Залежність часу виконання алгоритмів від розміру вхідного зображення для процесора Snapdragon 845

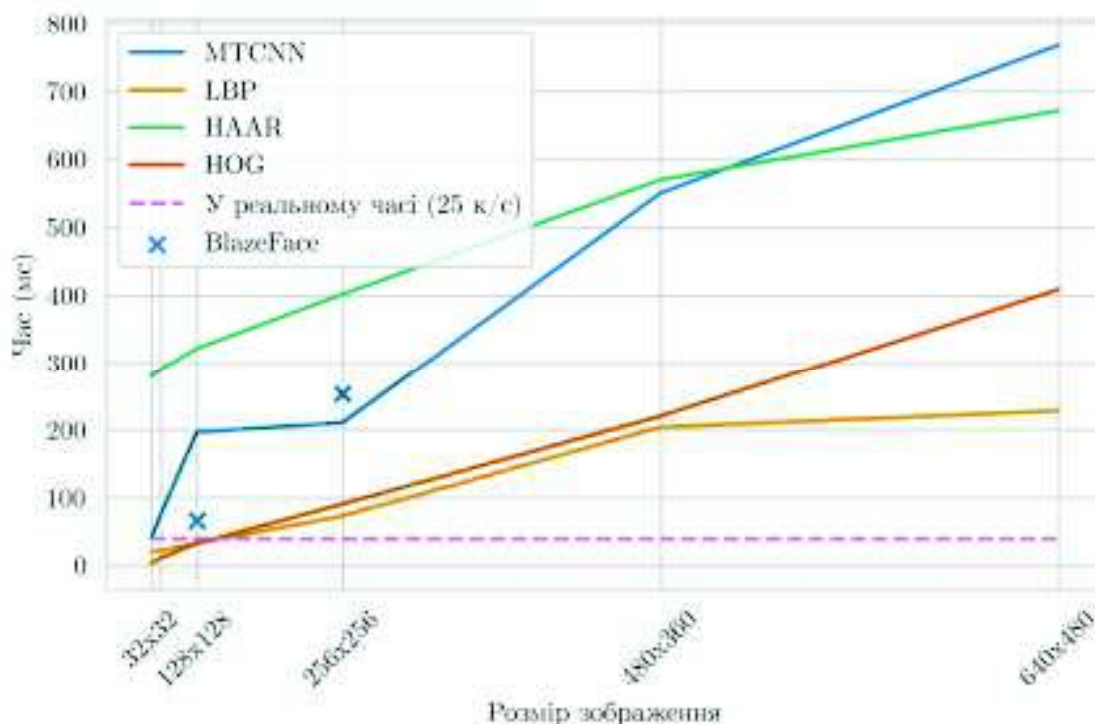


Рисунок 3 – Залежність часу виконання алгоритмів від розміру вхідного зображення для процесора Snapdragon 800

Отже, у другому експерименті кожному із алгоритмів ми виставлятимемо оцінку від нуля до трьох по одному балу за: 1) на знімку, що не містить облич, алгоритм не знаходить неіснуючі обличчя; 2) алгоритм знаходить одне велике обличчя, що присутнє на знімку; 3) алгоритм знаходить обидва невеличкі обличчя. Результати тестування показано на рис. 4, де зеленим кольором показане успішне виконання усіх сценаріїв, а жовтим і червоним – двох і одного відповідно. Зауважимо, що в нашому тестовому середовищі жоден із алгоритмів не передбачував обличчя на порожньому (майже білому знімку), отже кожен із алгоритмів отримав якнайменше один бал.

Візуальні зауваження алгоритмам. Каскади Хаара: на зображеннях розміру 480x360 і вище даний алгоритм знаходить неіснуючі обличчя, в більшості випадків такі «обличчя» то з'являються, то зникають на сусідніх кадрах тестових сценаріїв. LBP: на зображеннях розміру 256x256 або менше алгоритм не знаходить одне або обидва обличчя. HOG: досить якісно знаходить обличчя для знімків 256x256 або вище. MTCNN: знаходить велике обличчя навіть на знімках 32x32, що робить його найбільш адаптивним серед обраних. BlazeFace має дві фіксовані конфігурації та на обох відпрацьовує відмінно. Також слід зауважи-

ти, що завдяки розробленому модулю регресії, що стабілізує передбачення, «тремтіння» рамки обличчя на сусідніх кадрах майже відсутнє.

32x32		1	1	1	2
64x64		1	1	1	2
128x128	3	3	2	2	3
256x256	3	3	3	2	3
480x360		3	3	3	3
640x480		3	3	3	3
	BlazeFace	HAAR	HOG	LBP	MTCNN

Розмір зображення

Алгоритм

Рисунок 4 – Оцінка роботи алгоритмів на тестових сценаріях

В табл. 1 нами було помічено, що стандартне відхилення для різних алгоритмів суттєво відрізняється – цікаво зрозуміти таку поведінку. На рис. 5 показано час виконання кожного із алгоритмів в залежності від тестової сцени. Можна побачити, що для усіх каскадних алгоритмів, а саме: MTCNN, каскади Хаара, LBP наявний суттєвий зріст часу виконання алгоритму в залежності від типу сцени, тобто алгоритм «економить» час на обробці зображень, що не містять регіонів, схожих на обличчя. Така особливість алгоритмів корелює із тим, як працює мозок людини – ми витрачаємо значно менше часу на розпізнавання зображень із знайомими нам об'єктами, ніж із чимось невідомим або нечітким. Також цікаве те, що найсучасніший із обраних алгоритмів – BlazeFace – майже не має залежності від часу для моделі BlazeFace (front) та має обернену залежність для BlazeFace (back). Тобто коли кадр не містить облич, алгоритм витрачає більше часу на їх більш детальний пошук, аніж, коли наявне хоча б одне обличчя. Така відмінність робить цей алгоритм менш застосовним через вищу обчислювальну складність у випадках, коли більшість кадрів не буде містити обличчя.

Вдячності. Автор виражає вдячність Коряшкіній Ларисі Сергіївні к.ф.-м.н., доценту Національного технічного університету «Дніпровська політехніка» за плідну дискусію роботи та корисні зауваження. Роботу виконано у рамках державної бюджетної тематики за номером: 0121U109787.

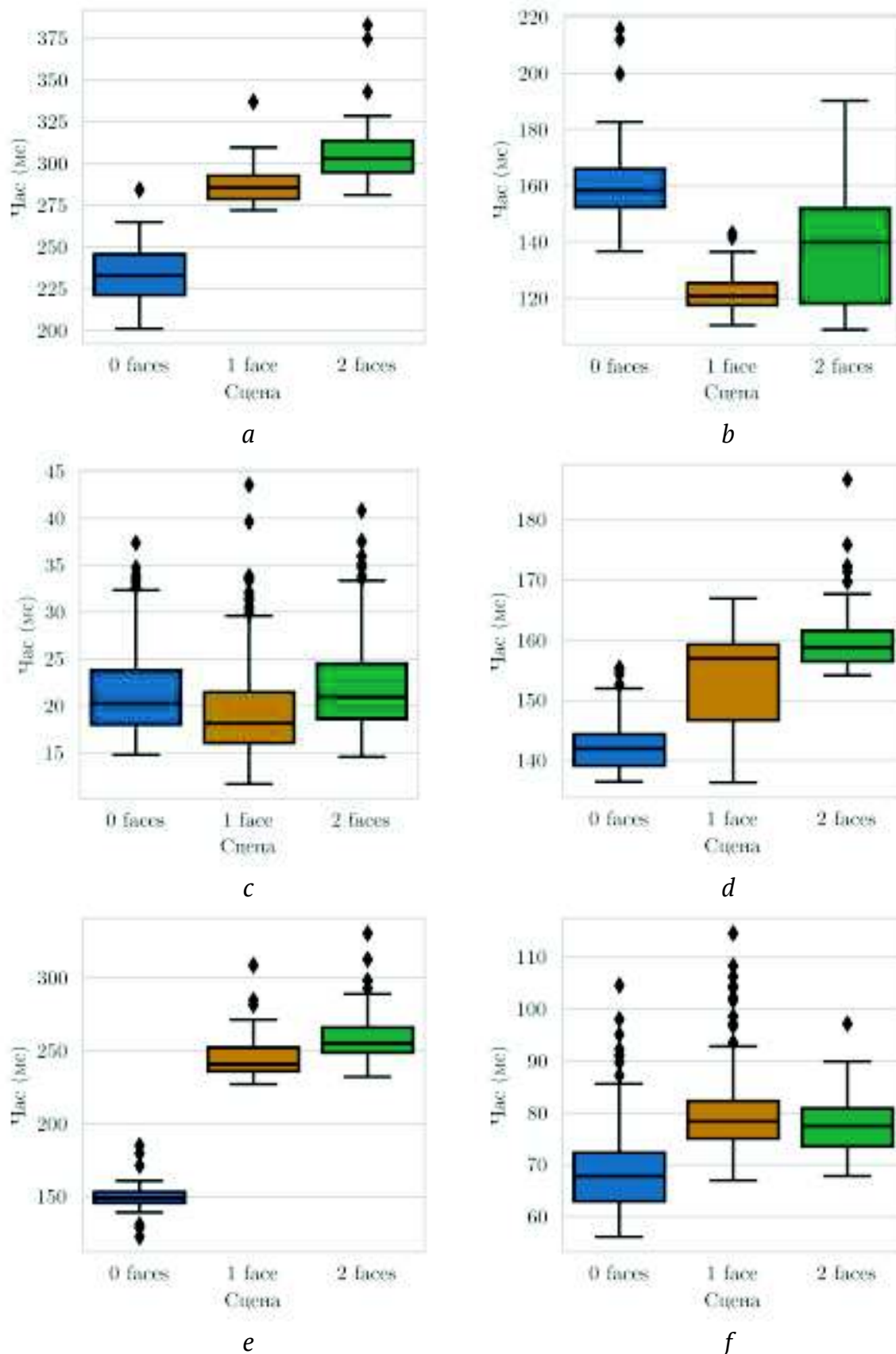


Рисунок 5 – Час виконання: *a* – MTCNN, *b* – BlazeFace (back),
c – BlazeFace (front), *d* – HOG, *e* – каскади Хаара, *f* – LBP

Висновки. В даній роботі було розроблено мобільні адаптації для 5 алгоритмів пошуку облич. З проведеного аналізу можна зробити висновок, що для практичного застосування найкращими є: 1) BlazeFace, який має стабільні та точні передбачення, однак, метод приймає на вхід зображення лише двох фіксованих розмірів, нетиповим є і більший час виконання для порожніх картинок порівняно із зображеннями з обличчями; 2) MTCNN економить обчислювальні ресурси, коли облич немає, та є найбільш адаптивним до різних розмірів вхідних зображень; 3) якщо ж найбільше значення має час виконання, то може бути використаний метод на основі HOG дескрипторів. Також в роботі було показано, що використання каскадів із детекторів дозволяє адаптивно змінювати час виконання методів в залежності від складності та змісту зображення. Автор сподівається, що отримані нові практичні результати поширять використання описаних методів у мобільних застосунках та підштовхнуть подальший розвиток їх модифікацій.

REFERENCES

1. Khabarлак K. Fast Facial Landmark Detection and Applications: A Survey / K. Khabarлак, L. Koriashkina // arXiv:2101.10808 [cs]. – 2021.
2. Khabarлак K.S. Mobile Access Control System Based on RFID Tags and Facial Information / K.S. Khabarлак, L.S. Koriashkina // Bulletin of National Technical University “KhPI”. Series: System Analysis, Control and Information Technologies. – 2020. – № 2 (4). – P. 69-74.
3. Viola P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. – 2001. – Vol. 1. – P. I-I.
4. Bradski G. The OpenCV Library / G. Bradski // Dr. Dobb’s Journal of Software Tools. – 2000.
5. Dalal N. Histograms of oriented gradients for human detection / N. Dalal, B. Triggs // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05). – 2005. – Vol. 1. – P. 886-893 vol. 1.
6. King D.E. Dlib-ml: A Machine Learning Toolkit / D.E. King // Journal of Machine Learning Research. – 2009. – Vol. 10. – P. 1755-1758.
7. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks / K. Zhang [et al.] // IEEE Signal Processing Letters. – 2016. – Vol. 23. – № 10. – P. 1499-1503.
8. BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs / V. Bazarevsky [et al.] // arXiv:1907.05047 [cs]. – 2019.

9. SSD: Single Shot MultiBox Detector / W. Liu [et al.] // Computer Vision – ECCV 2016. – Cham: Springer International Publishing, 2016. – P. 21-37.
10. MobileNetV2: Inverted Residuals and Linear Bottlenecks / M. Sandler [et al.] // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2018. – P. 4510-4520.

Received 11.09.2021.

Accepted 20.09.2021.

Особенности работы методов поиска лиц на мобильных устройствах

Всё большее количество приложений использует поиск лиц непосредственно на мобильных устройствах. Однако, фокусом современных исследований по направлению является увеличение качества поиска, вне зависимости от времени работы, что делает ряд интересных и эффективных алгоритмов неприменимыми в практических сценариях. В данной работе разработано адаптации 5 «классических» и нейросетевых алгоритмов поиска лиц для выполнения на мобильных устройствах. Проанализировано время и качество работы алгоритмов в зависимости от ряда факторов (скорости процессора мобильного устройства, типичного сценария использования, и др.), показано неочевидную зависимость времени выполнения от содержимого изображения. Надеемся, что приведенные практические рекомендации касательно использования алгоритмов увеличат их применимость в мобильных приложениях и подтолкнут дальнейшее развитие их модификаций.

Mobile Face Detection Algorithm Inference Traits

An ever-growing number of applications uses mobile face detection. However, most of the modern research papers focus on increasing detection quality while paying no attention to detection time. This means that many of the state-of-the-art algorithms are inapplicable on mobile due to excessively large detection time.

The goal of this this paper is to adapt 5 of the face detection algorithms for inference on mobile devices and analyze their performance characteristics. These algorithms include established methods: Haar Cascades, LBP, HOG, as well as, novel neural-network-based algorithms: MTCNN, BlazeFace.

The main research material. We conduct the experiments on three scenes typical for mobile face recognition systems: when there are no faces, 1 or 2 faces. For testing we have implemented an Android application. 2 widespread processors, namely Snapdragon 800 and 845, were selected for time measurements. Having tested the algorithms, we note that all them can run at real-time speeds for images of size 128x128 and only 2 of them (LBP, HOG) on 256x256 on the faster Snapdragon 845. On the slower Snapdragon 800 only BlazeFace, LBP, HOG can run at resolutions not higher than 128x128. We suggest not using Haar or LBP cascades in practice as their accuracy is quite low.

Conclusions. Based on the research conducted, we suggest that for the practical use-cases the best algorithms are: 1) BlazeFace, which has stable and accurate predictions, however, the method accepts only two image resolutions as input, in addition, higher inference time for empty

images than for images with faces is untypical; 2) MTCNN, thanks to the cascaded architecture, conserves the resources when input frames have no faces. This algorithm is also the most adaptive and can run at resolutions as low as 32x32 given that the faces are quite large; 3) in case if inference time is of the most importance, we suggest using HOG-based algorithm. In this paper we have also shown that cascaded algorithm architecture dynamically changes execution time depending on image content and its complexity, which follows how we, humans, think. We hope that the novel practical results obtained, will increase the use of the above-described methods in mobile applications and will boost the development of the algorithm modifications.

Хабарлак Костянтин Сергійович – аспірант, Національний технічний університет «Дніпровська політехніка», Україна.

Хабарлак Константин Сергеевич – аспірант, Национальный технический университет «Днепропетровская политехника», Украина.

Kostiantyn Khabarlak – Postgraduate student, Dnipro University of Technology, Ukraine.