

РИБАК ЛЕВ-ВОЛОДИМИР

Національний лісотехнічний університет України

<https://orcid.org/0000-0002-5915-5520>e-mail: rybaklevko@ukr.net

ДЕНДЮК МИХАЙЛО

Національний лісотехнічний університет України

<https://orcid.org/0000-0002-7631-022X>e-mail: dendyuk@nlту.edu.ua

ДОСЛІДЖЕННЯ ПОРІВНЯЛЬНИХ ХАРАКТЕРИСТИК РОЗПІЗНАВАННЯ ОБЛИЧ НА РІЗНИХ ПЛАТФОРМАХ

В роботі наведено результати порівняння розпізнавання облич методом власних граней з використанням різного апаратного забезпечення. Виконано опис загальної проблематики пов'язаної з розпізнаванням облич, вплив цієї технології на наукові дослідження і практичні завдання в сучасному світі. Здійснено аналітичний огляд підходів до розпізнавання облич з використанням нейронних мереж та методу власних граней. Проаналізовано теоретичні переваги та недоліки кожного із методів. Описано апаратне забезпечення, яке буде використано для кожного із підходів та який вплив на експеримент воно матиме. Обґрунтовано вибір саме методу власних граней для здійснення експерименту. Коротко описано математичну модель методу власних граней, її параметри та можливості практичного застосування. На основі цієї математичної моделі реалізоване програмне забезпечення для здійснення експериментального дослідження по розпізнаванню облич. Для реалізації програмного забезпечення використано мову програмування Python та засоби бібліотеки OpenCV. Проведено навчання програмної моделі на основі вхідного відео із людським обличчям. Представлено візуальний приклад використання натренованої моделі. Експериментальним шляхом виміряно характеристики розпізнавання облич на різному апаратному забезпеченні. Здійснено порівняння якості розпізнавання з використанням таких характеристик: кількість вхідних фреймів, розпізнанні обличчя з відомим іменем, розпізнанні обличчя з невідомим іменем та середній час розпізнавання в мілісекундах. Представлено ці результати у вигляді текстового та графічного відображення. За результатами експериментів встановлено, що запропонований метод є актуальним для пристроїв з низькою потужністю або з низьким запитом до продуктивності.

Ключові слова: апаратне забезпечення, графічний процесор, ЦП, власні грані, нейронні мережі.

RYBAK LEV-VOLODYMYR, DENDIUK MYKHAILO

Ukrainian National Forestry University

RESEARCH OF COMPARATIVE CHARACTERISTICS OF FACE RECOGNITION ON DIFFERENT PLATFORMS

The article presents the results of a comparison of face recognition by the method of eigenfaces using different hardware. A characterization of the general problems associated with face recognition, the impact of this technology on scientific research and practical tasks in the modern world is described. An analytical review of approaches to face recognition using neural networks and the method of eigenfaces was carried out. The theoretical advantages and disadvantages of each of the methods are analyzed. The hardware that will be used for each of the approaches and the impact it will have on the experiment are described. The choice of the eigenface method for the experiment is justified. The mathematical model of the method of eigenfaces, its parameters and possibilities of practical application are briefly described. On the basis of this mathematical model, software was implemented for the implementation of experimental research on face recognition. The Python programming language and OpenCV library tools were used to implement the software. A software model was trained based on an input video with a human face. The training results are demonstrated in the form of formatted output images and the trained model. A visual example of using the trained model is presented. The characteristics of face recognition on different hardware were experimentally measured. A comparison of the recognition quality was made using the following characteristics: the number of incoming frames, face recognition with a known name, face recognition with an unknown name, and the average recognition time in milliseconds. These results are presented in the form of textual and graphic display. According to the results of the experiments, it was established that the proposed method is relevant for devices with low power or low demand for performance.

Keywords: hardware, Graphics processing unit, Central processing unit, Eigenface algorithm, CNN.

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

Технологія розпізнавання обличчя стала фундаментальним компонентом у різних секторах, починаючи від систем безпеки та закінчуючи споживчими програмами, що забезпечує розширену автентифікацію, спостереження та персоналізований досвід користувачів. Протягом останнього десятиліття відбувся стрімкий розвиток апаратного забезпечення, що привело до його більшої доступності та обширнішого використання.

З все більшою інтеграцією алгоритмів розпізнавання облич у повсякденне життя виникло питання стосовно порівняльних характеристик такого роду алгоритмів з використанням різних архітектур процесорів, операційних систем, графічних процесорів та інших компонентів комп'ютерної техніки. Знаходження необхідного балансу між використовуваним алгоритмом та відповідних апаратних забезпеченням і є одним з ключових викликів. Якість виконання роботи на різних платформах залежить від багатьох умов, таких як освітлення, зміна пози або включення різних демографічних груп.

Це дослідження спрямоване на вивчення порівняльних характеристик розпізнавання обличчя на різних платформах. Провівши ретельний аналіз використання алгоритму розпізнавання обличчя описаного

математичною формулою на основі таких факторів, як швидкість, якість, впевненість та точність, це дослідження дозволить краще зрозуміти як можна використати, де можна використати та яким чином оптимізувати системи розпізнавання облич [1].

Аналіз досліджень та публікацій

Розглянемо алгоритми на основі нейронних мереж(CNN) та алгоритм Eigenface, оскільки саме вони є широкопоширеними та актуальними для апаратного забезпечення з обмеженими ресурсами. Розглянемо існуючі публікації, нещодавні дослідження та висновки щодо використання цих алгоритмів з використанням різного платформ.

Нещодавні дослідження демонструють, що графічні процесори(GPU) є більш ефективними з точки зору паралелізму обчислень порівняно з стандартним центральним комп'ютерним процесором(CPU). Це в свою чергу приводить до пріоритетності їх вибору для роботи з алгоритмами на основі нейронних мереж. Створюються бібліотеки, такі як cuDNN від NVIDIA, які займаються оптимізацією обчислень і роблять GPU навіть ефективнішим [2,4].

Альтернативою використанню графічних процесорів з алгоритмами CNN є FPGA (програмованих вентильних матриць) і ASIC (спеціальних інтегральних схемах), які можуть перевершувати їх з точки зору енергоефективності та швидкості, особливо в периферійних пристроях з обмеженими ресурсами [4]. Використовуючи при цьому менше пам'яті, уникаючи використання глобальних буферів даних, робить їх ідеальними для розпізнавання обличчя на основі CNN у мобільних або вбудованих системах [2].

Алгоритм Eigenface передував підходам до розпізнавання облич з використанням нейронних мереж. Він покладається на математичні операції(лінійна алгебра), зокрема множення матриць, що означає з точки зору апаратного забезпечення більш цільове використання саме центрального процесора та його ресурсів для ефективних розрахунків.

Він також може бути реалізований на графічному процесорі, тоді використовуються альтернативні математичні підходи, так як скалярні добутки для проектування зображень на простір власних граней. Це дозволяє використати паралелізацію, яка є основною перевагою графічного процесора. Це покращення особливо помітно під час обробки великих наборів даних або роботи в системах реального часу [3].

Є можливість динамічно розподіляти завдання на відповідний апаратний засіб, використовуючи сучасні реалізації цього алгоритму, тобто вибирати або CPU, або GPU для тих чи інших завдань. До прикладу, це алгоритми, які використовують OpenCL. Це приводить до оптимізованого та гнучкого використання паралелізму та потужностей платформ [3].

Ще одним варіантом є специфікована оптимізація під апаратне забезпечення. Обидва алгоритми значно виграють від цільової апаратної оптимізації. З використанням квантування, скорочення та пошуку нейронної архітектури (NAS), нейронні мережі стали краще застосовуваними для пристроїв з обмеженими ресурсами [2]. Для методу власних граней використання оптимізованого ядра дозволяє стати більше ефективним для програм реального часу [3].

Підсумовуючи, можна сказати, що ефективність використання цих алгоритмів для розпізнавання облич є відмінним для різних платформ. Метод власних граней є продуктивнішим на пристроях з обмеженим ресурсом, а CNN на пристроях з потужним графічним процесором.

Формулювання цілей статті

Метою роботи є: дослідження та порівняння характеристик розпізнавання облич на різному апаратному забезпеченні.

Виклад основного матеріалу

Метод власних граней використовується для розпізнавання облич. Основна ідея цього методу полягає у визначенні власних векторів (власних граней) із коваріаційної матриці навчальних зображень облич. Ці власні вектори є ключовими характеристиками, що описують спільні риси облич у наборі зображень. Зображення обличчя можна представити як лінійну комбінацію цих власних векторів, що дозволяє ефективно зменшити розмірність даних і використовувати їх для розпізнавання[5, 6].

Тепер сформулюємо формулу обчислення власних граней [5]:

$$\hat{X} = \bar{F} + A^T v_k ((A^T \cdot v_k) \cdot (x - \bar{F})), \quad (1)$$

де \bar{F} - середнє значення матриць облич,

A – матриця відхилень,

v – власні грані,

x - нове зображення.

Здійснимо практичну реалізацію методу власних граней та проведемо тестування на різному апаратному забезпеченні. Для цього використаємо мову програмування Python та засоби бібліотеки OpenCV.

Для того, щоб почати роботу безпосередньо з розпізнаванням облич спершу потрібно провести тренування моделей для розпізнавання. Виділимо декілька ключових компонентів такого тренування: каскадний класифікатор, модель, вхідні відео матеріали та вихідні зображення.

Для каскадного класифікатора використаємо cv2.CascadeClassifier з бібліотеки OpenCV. Клас може бути використаний для знаходження об'єктів на вхідних зображеннях та відеоматеріалах на основі

попередньо навченої моделі. В нашому випадку нас цікавить знаходження облич. Каскадний класифікатор працює, застосовуючи ряд фільтрів (каскадів) до різних частин зображення. Кожен регіон, який пройшов усі етапи класифікатора, вважається таким, що містить об'єкт.

Класифікатор здійснює процес навчання базуючись на позитивних і негативних прикладах об'єкта інтересу. Далі відбувається фільтрація кандидатів, які не підпадають обраним критеріям, що призводить до навчання моделі та більш точного виявлення [7].

Проведемо навчання моделі на основі вхідного відео зі чітко зображеним обличчям та ідентифікатором у якості імені(рис.1).

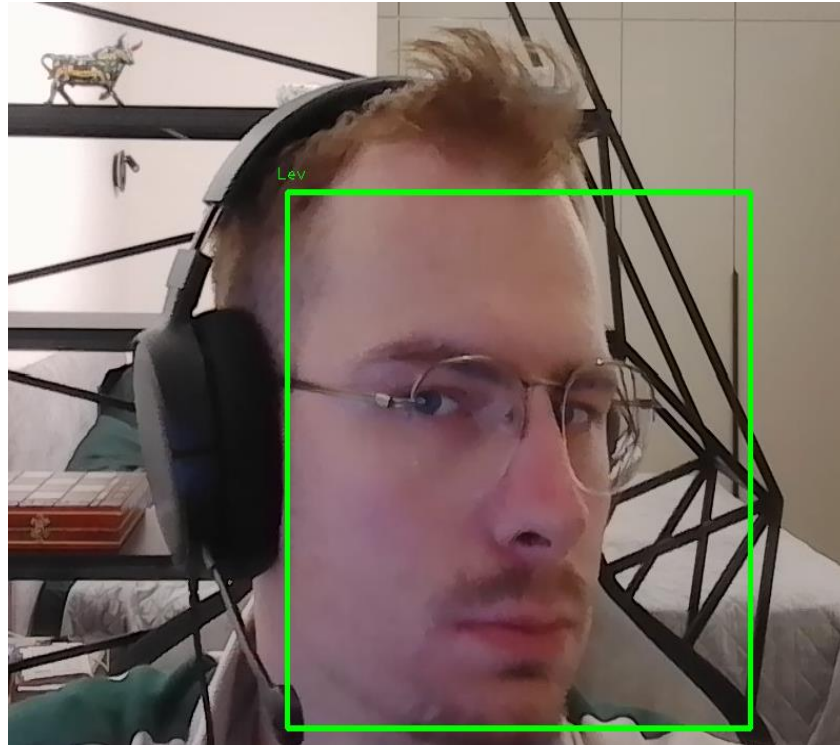


Рис. 1. Вхідне відео для тренування моделі

В результаті аналізу вхідного відео відбувається форматування, обробка з використанням каскадного класифікатора та збереження зображень із вхідним обличчям, яке в результаті буде використане для тренування моделі на розпізнавання. З повного вхідного зображення вирізано зменшене у форматі 20x20 в якому було знайдено обличчя. Це зроблено з метою економії пам'яті та пришвидшення навчання. Ці зображення збережені за відповідним ідентифікатором, а саме – іменем користувача(рис.2).



Рис. 2. Зображення для тренування моделі на основі вхідного відео

Далі здійснюється тренування моделі на основі використанням вхідних зображень. Результатом цієї роботи є натренована модель представлено у форматі .xml файлу(рис.3).

```
eigen_trained_data.xml × +
File Edit View
<?xml version="1.0"?>
<opencv_storage>
<opencv_eigenfaces>
  <threshold>1.7976931348623157e+308</threshold>
  <num_components>1139</num_components>
  <mean_type_id="opencv-matrix">
    <rows>1</rows>
    <cols>4000</cols>
    <dt>d</dt>
    <data>
      108.95434591747147 108.62949956101843 108.30553116769096
      108.87796312554873 109.78050921861282 109.18261633011413
      108.59350307287094 107.5548726953468 106.02282704126426
      103.84108867427568 102.94995610184372 102.69622475856013
      101.01053555750659 99.71641791044776 101.17910447761194
      102.30201931518876 102.52414398595259 102.31870061457418
      101.85338015803336 98.943810359964885 95.329236172080769
      91.208077260755047 87.275680421422294 84.149253731343279
      82.180860403863036 82.050921861281822 82.884108867427571
      84.380158033362605 86.14310798946444 88.101843722563657
      91.291483757682172 94.438103599648812 97.390693590869191
      99.955223880597018 102.20105355575066 103.585601404741
      104.56277436347673 105.39069359086919 106.54082528533802
      107.51185250219491 108.81474978050922 110.57067603160667
      112.91220368744513 115.78753292361721 117.33538191395961
      118.62862159789289 118.58735733099211 118.61545215100966
      118.08691834942933 117.58208955223881 117.08340649692713
```

Рис. 3. Результат тренування моделі

Тепер проведемо аналіз результатів роботи моделі подаючи на вхід відео з тією ж людиною. Розпізнавання покаже ім'я та показник впевненості в цифровому еквіваленті(рис.4).

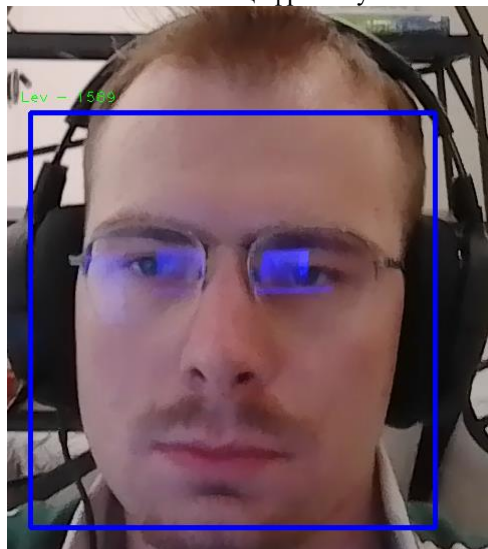


Рис. 4. Демонстрація роботи натренованої моделі

Згідно поставленої мети статті проведемо аналіз та порівняння роботи за характеристиками на різному апаратному забезпеченні. Для цього використаємо Raspberry Pi 4 Model B 2GB та Asus ROG Zephyrus G14 GA402N. Проведемо короткий огляд ключових технічних характеристик цих пристроїв.

Raspberry Pi 4 Model B (2 ГБ) оснащений такими засобами: чотирьохядерний центральний процесор ARM Cortex-A72 (64-розрядний) @ 1.5 ГГц, 2 ГБ оперативної пам'яті LPDDR4-3200 SDRAM та графічний процесор Broadcom VideoCore VI [8]. Важливим є те, що графічний процесор не має можливостей паралельної обробки зображень, що робить його достатньо неефективним для CNN алгоритмів. Натомість алгоритм Eigenface є менш залежним від паралелізації, а більш від потужностей центрального процесора разом з графічними характеристиками яких є достатніми для виконання завдань по розпізнаванню облич.

Розглянемо на противагу більш потужний Asus ROG Zephyrus G14 GA402N та його технічні характеристики, а саме: центральний процесор AMD Ryzen 9 7940HS (8 ядер, 16 потоків, до 5.2 ГГц),

оперативна пам'ять 16 ГБ DDR5-4800 та графічний процесор NVIDIA GeForce RTX 4060 (з 8 ГБ GDDR6 VRAM) [9]. Для пристрою з такими висококласними характеристиками не буде проблемою використати CNN алгоритм з врахуванням ядра CUDA графічного процесора. Для алгоритму Eigenface продуктивність буде аж надто високою і як наслідок таких же результатів можна добитися і на менш потужних пристроях.

Якщо порівнювати продуктивність цих обох пристроїв, то Asus значно перевищує Raspberry Pi 4 з точки зору обчислювальних можливостей, продуктивності та швидкої. Він буде більш ефективним при використанні CNN алгоритму, однак не очікується великої відмінності при використанні методу власних граней. Raspberry Pi більше підходить для легких невеликих завдань розпізнавання облич або для освітніх і експериментальних цілей. Також, слід пам'ятати про значну дешевизну, розмір та доступність такого роду одноплатних комп'ютерів у порівнянні з повноцінними персональними комп'ютерами. Часто при практичному чи науковому використанні таких пристроїв саме такі обмеження є ключовими. Тобто при здійснюваному експерименті ноутбук Asus буде використаний як «ідеальний» орієнтир відносно якого можна буде побачити вплив змін апаратного забезпечення на якість розпізнавання, оцінюючи різні характеристики цього зображення.

Перейдемо безпосередньо до проведення експерименту. Під час його виконання буде порівнювати такі характеристики як: кількість вхідних фреймів, розпізнанні обличчя з відомим іменем, розпізнанні обличчя з невідомим іменем та середній час розпізнавання в мілісекундах. Підхід до тренування та використання моделей для розпізнавання облич був попередньо описаний. Тренування проведемо для 2 різних облич на основі ряду відеоматеріалу для аналізів. Для аналізу вхідне відео розбивається на фрейми(окремі зображення) та кожне з них аналізується. Натренована модель класифікує обличчя й відповідно розподіляє результати на категорії з відомими та невідомими іменами.

Проведемо описаний експеримент та виведемо дані у текстовому й графічному форматі(рис.5).

1) Результати Asus:

Lev. Frame counter: 681, frames without detection 40, Known names: 607, Unknown names: 34, Average time: 70.93 ms

Gaidai. Frame counter: 873, frames without detection 0, Known names: 21, Unknown names: 852, Average time: 54.06 ms

2) Результати Raspberry PI

Lev. Frame counter: 681, frames without detection 39, Known names: 605, Unknown names: 37, Average time: 256.35 ms

Gaidai. Frame counter: 873, frames without detection 0, Known names: 20, Unknown names: 853, Average time: 119.38 ms

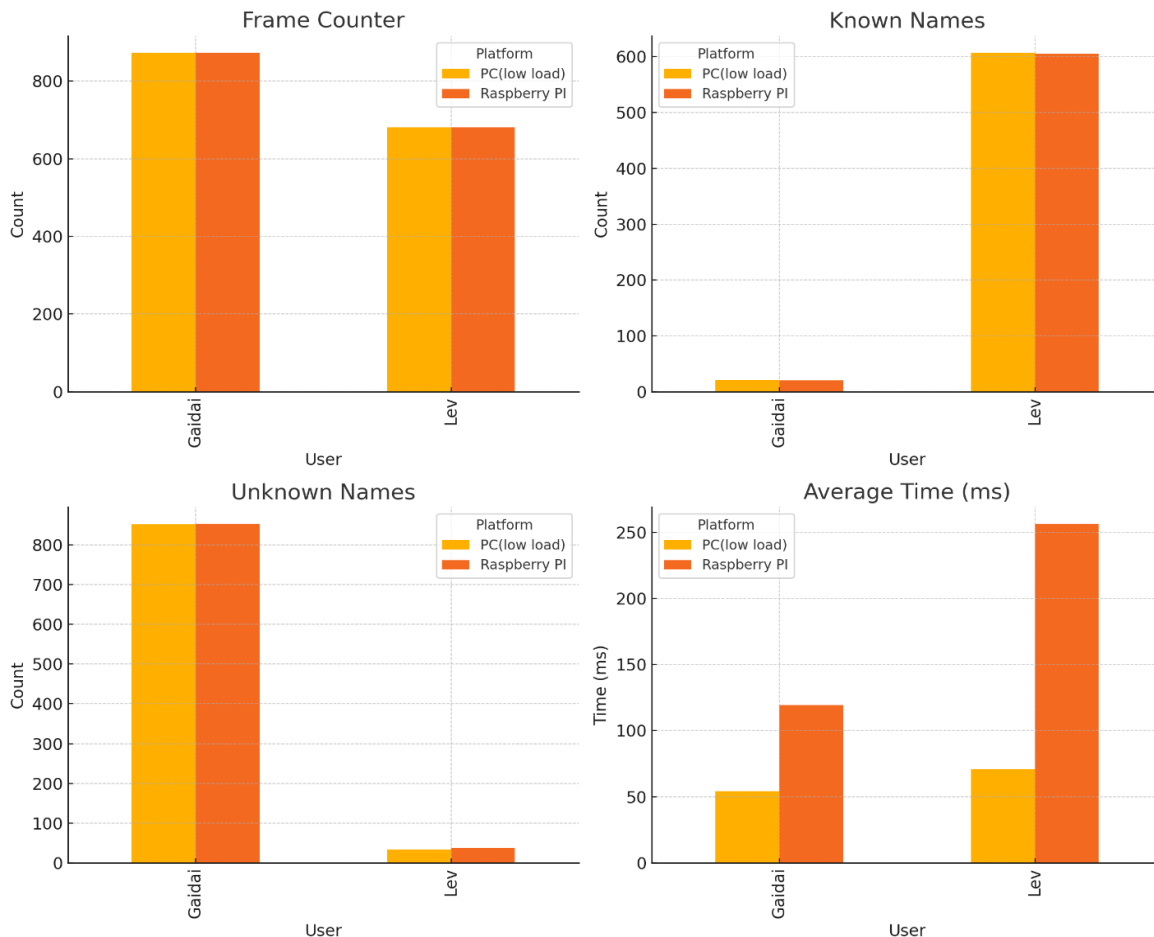


Рис. 5. Результати аналізу розпізнавання облич на різному апаратному забезпеченні

В результаті цього експерименту видно, що на пристрої з набагато більш обмеженими технічними характеристиками кількість розпізнаваних облич з відомими й невідомими іменами є практично ідентичним, ключовою відмінністю є лише час цього розпізнавання. Це свідчить про коректність гіпотези, що метод власних граней буде ефективним для розпізнавання облич на апаратному забезпеченні з порівняно низькими потужностями.

Висновки

Експериментальним шляхом досліджено та порівняно характеристики розпізнавання облич на різному апаратному забезпеченні. Продуктивність алгоритмів розпізнавання осіб оцінювалася на двох платформах: Asus і Raspberry Pi. Ключовими записаними показниками були загальна кількість кадрів, кадри без виявлення, виявлені відомі та невідомі імена та середній час на кадр. Продуктивність оцінювалася у двох користувачів. Asus перевершив Raspberry Pi за швидкістю обробки, маючи значно нижчий середній час обробки. Однак, з точки зору точності виявлення облич на зображеннях, обидві платформи показали подібні результати, підкреслюючи потенціал Raspberry Pi для програм по розпізнаванню облич з використанням методу власних граней на пристроях з низькою потужністю або з низьким запитом до продуктивності.

Література

- . Deng, J. Li, J. T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 8604–8608.
14. D. Ghimire, D. Kil, and S. Kim, "A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration," *Electronics*, vol. 11, no. 6, 2022. [Online]. Available: <https://doi.org/10.3390/electronics11060945>.
15. "GPU Programming - Face Detection Using the Eigenfaces Algorithm on the GPU," *Microsoft Learn*. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2018/october/gpu-programming-face-detection-using-the-eigenfaces-algorithm-on-the-gpu>
16. "Convolutional Neural Networks and Processing on CPU and GPU," *web.cs.ucla.edu*. [Online]. Available: https://web.cs.ucla.edu/~miodrag/CS259_Winter2018/Reports/JieWang_report.pdf
17. Рибак Л., Дендюк М. Математична модель для удосконалення методів розпізнавання обличчя в системі безпеки аутентифікації користувачів. Комп'ютерне моделювання та інформаційні технології: матеріали п'ятої науково-практичної конференції студентів, аспірантів та молодих вчених (Львів, 19-21 жовтня 2023 р.). – Львів: ННІ КНІТ НЛТУ України, 2023. – С. 29-33.
18. Turk, M., Pentland, A. Eigenfaces for recognition. *J.Cogn. Neurosci.* 1991, 3, 71-86 <https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>
19. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
20. Raspberry Pi Foundation. "Raspberry Pi 4 Model B Product Brief." [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
21. ASUS. "ASUS ROG Zephyrus G14 (2024) Product Specifications." [Online]. Available: <https://rog.asus.com/laptops/rog-zephyrus/rog-zephyrus-g14-series/>

References

- . Deng, J. Li, J. T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at microsoft," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 8604–8608.
2. D. Ghimire, D. Kil, and S. Kim, "A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration," *Electronics*, vol. 11, no. 6, 2022. [Online]. Available: <https://doi.org/10.3390/electronics11060945>.
3. "GPU Programming - Face Detection Using the Eigenfaces Algorithm on the GPU," *Microsoft Learn*. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2018/october/gpu-programming-face-detection-using-the-eigenfaces-algorithm-on-the-gpu>
4. "Convolutional Neural Networks and Processing on CPU and GPU," *web.cs.ucla.edu*. [Online]. Available: https://web.cs.ucla.edu/~miodrag/CS259_Winter2018/Reports/JieWang_report.pdf
5. Rybak L., Dendiuk M. Matematychna model dlia udoskonalennia metodiv rozpiznavannia oblychchia v systemi bezpeky autentyfikatsii korystuvachiv. Kompiuterne modeliuвання ta informatsiini tekhnolohii: materialy piatoi naukovo-praktychnoi konferentsii studentiv, aspirantiv ta molodykh vchenykh (Lviv, 19-21 zhovtnia 2023 r.). – Lviv: NNI KNIT NLTU Ukrainy, 2023. – S. 29-33.
6. Turk, M., Pentland, A. Eigenfaces for recognition. *J.Cogn. Neurosci.* 1991, 3, 71-86 <https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>
7. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.
8. Raspberry Pi Foundation. "Raspberry Pi 4 Model B Product Brief." [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
9. ASUS. "ASUS ROG Zephyrus G14 (2024) Product Specifications." [Online]. Available: <https://rog.asus.com/laptops/rog-zephyrus/rog-zephyrus-g14-series/>