педагогічний виклик. Від здобувачів освіти вимагається опанування комплексних систем прийняття рішень щодо оцінки ризиків та контролю фізичного доступу. Хоча гейміфікація широко застосовується для формування базової обізнаності з питань безпеки (наприклад, антифішингу), глибоке вивчення стандартів вимагає інтеграції складних компетенцій.

Для подолання цих викликів у межах пілотної ініціативи було здійснено перехід від статичних, перевантажених текстом матеріалів до гейміфікованого середовища в системі управління навчанням ATutor. Інтеграція інструментарію H5P дозволила створити нелінійні «сценарії розгалуження» (Branching Scenarios) на основі методології проблемно-орієнтованого навчання (PBL). Виступаючи в ролі аудиторів фізичної безпеки, студенти виявляли потенційні вразливості та приймали рішення, базуючись виключно на протоколах ISO 27001. Такі сценарії надійно занурюють здобувачів у ситуації, де вони повинні зважувати наслідки своїх дій, що стимулює критичне мислення. Емпіричні дані, отримані через навчальну аналітику з ATutor та H5P, продемонстрували важливу закономірність розвитку аналітичних навичок. Студенти, які спочатку обирали помилкові варіанти, були змушені аналізувати та виправляти змодельовані порушення і продемонстрували значно кращі результати у підсумкових комплексних оцінюваннях на критичне мислення порівняно з тими, хто пройшов сценарії з першої спроби.

Успішна реалізація цього проєкту слугує надійною моделлю для розбудови інституційного потенціалу, підтверджуючи, що інтерактивна та доступна цифрова педагогіка [1] може успішно функціонувати навіть в умовах серйозних зовнішніх викликів, ефективно озброюючи студентів критичними навичками, необхідними для майбутнього.

1. Zagorodna N., Skorenkyy Y., Kunanets N., Baran I., Stadnyk M., Augmented Reality-enhanced learning tools development for cybersecurity major. *CEUR Workshop Proceedings.* – 2022. – V. 3309. – p. 25–32.

Високопродуктивне розпізнавання облич на базі CUDA та Dlib у структурі комплексних систем забезпечення кібербезпеки

УДК 004.93

Олексій Смірнов[1], Віктор Заріцкий[2],
Костянтин Буравченко[3], Сергій Смірнов[4]

Центральноукраїнський національний технічний університет,
[1]dr.smirnovoa@gmail.com, [2]viktorzarickiy@gmail.com,
[3]buravchenkok@gmail.com, [4]smirnov.ser.81@gmail.com

Стрімкий прогрес у галузі розпізнавання образів зумовлений глибокою інтеграцією технологій штучного інтелекту в стратегічні сектори: від військової ідентифікації об’єктів БПЛА в умовах російсько-української війни до автоматизації медицини та промисловості. Це дослідження присвячене вдосконаленню систем безпеки, де детекція облич виступає фундаментальним компонентом інфраструктури "розумних міст", банківських установ та інформаційно-телекомунікаційних систем. Ключовим викликом для таких

систем є необхідність синтезу високої точності розпізнавання та обробки відеоданих у режимі реального часу. Ця вимога детермінує перехід від класичних алгоритмів до методів глибокого навчання (Deep Learning), які демонструють високу інваріантність до змін освітлення, ракурсних поворотів та оклюзій. Бібліотека dlib надає інструментарій для реалізації двох різних концепцій: Метод HOG (Histogram of Oriented Gradients): орієнтований на CPU, відрізняється швидкодією для фронтальних зображень, проте має низьку стійкість до нетипових ракурсів; Архітектура MMOD CNN (Maximum-Margin Object Detection): забезпечує прецизійну точність у динамічних сценаріях за рахунок аналізу багаторівневих ознак.

Проблема високої ресурсомісткості CNN-моделей (мільйони операцій на кадр), що створює "пляшкове горлечко" при використанні центральних процесорів, вирішується впровадженням технології NVIDIA CUDA. Паралелізація обчислень на графічних прискорювачах (GPU) дозволяє досягти оптимального балансу між точністю і частотою кадрів (FPS). Наукова новизна роботи полягає у комплексному порівняльному аналізі та кількісній оцінці ефективності переходу від класичного детектора HOG до GPU-прискореного методу MMOD CNN за допомогою розробленої методики бенчмаркінгу

Алгоритми інтелектуального аналізу даних та їх інтеграція з ШІ. Інструментарій dlib базується на двох підходах: класичному детектуванні та методах глибокого навчання. Алгоритм HOG, оптимізований для CPU, демонструє стабільну роботу з фронтальними обличчями, проте втрачає ефективність при зміні ракурсів. На противагу йому, архітектура MMOD CNN забезпечує прецизійну точність у динамічних умовах шляхом аналізу ієрархічних ознак. Висока обчислювальна складність нейромережі нівелюється застосуванням GPU-прискорення, що дозволяє виконувати масивні матричні операції паралельно та з мінімальними затримками.

Технологія CUDA для прискорення обчислень. Використання архітектури NVIDIA CUDA дозволяє перетворити графічний процесор на потужний обчислювальний вузол для виконання паралельних завдань. На відміну від центральних процесорів (CPU), спроектованих для послідовної обробки команд, GPU з тисячами спеціалізованих ядер забезпечує одночасне оперування великими масивами даних. Завдяки інтеграції бібліотеки dlib із CUDA, ресурсомісткі тензорні операції переносяться на відеокарту, що мінімізує завантаження CPU та гарантує роботу нейромережевих моделей у реальному часі. Для підтвердження ефективності цього методу було створено авторське програмне забезпечення мовою Python, яке за допомогою інструментів NumPy та Matplotlib здійснює комплексний бенчмаркінг та статистичну оцінку алгоритмів детекції

Методика експерименту та програмна реалізація. З метою забезпечення високої достовірності вимірювань розроблено інструментальний модуль FaceDetectionBenchmark. Алгоритм випробувань базується на кешуванні вхідних даних у RAM та обов’язковій фазі "warm-up" (прогріву) графічного процесора, що нівелює вплив ініціалізації CUDA-контексту на результат. Застосування прецизійних таймерів у поєднанні зі статистичним аналізом

середньоквадратичного відхилення дало змогу кількісно оцінити стабільність інференсу та виміряти рівень джиттеру в динамічному відео потоці.

Результати роботи методу HOG (CPU). Апробація класичного детектора на базі CPU продемонструвала високу прогнозованість часових показників. Середня латентність обробки кадру склала 25-30 мс (30-40 FPS), що варіюється залежно від вхідної роздільної здатності. Головною перевагою методу HOG визначено економічну доступність та відсутність потреби у спеціалізованих обчислювачах. Водночас виявлено суттєве обмеження точності: алгоритм втрачає працездатність при значних ракурсних відхиленнях обличчя та в умовах динамічної зміни освітленості.

Результати роботи методу CNN (GPU/CUDA). Використання CNN-детектора без апаратного прискорення виявило критичну нестачу продуктивності: затримка інференсу понад 800 мс (~1.2 FPS) робить застосування центрального процесора (CPU) недоцільним для систем реального часу. Впровадження технології NVIDIA CUDA забезпечило радикальний приріст швидкодії, скоротивши час обробки до 15–20 мс, що відповідає частоті понад 50 FPS. Окрім високої продуктивності, нейромережевий підхід продемонстрував вищу повноту детекції (recall), стабільно ідентифікуючи обличчя в складних ракурсах, де класичний метод HOG виявився неефективним

Порівняльна характеристика. Отримані дані підтверджують ефективність залучення GPU-ресурсів: показник інтенсифікації обчислень (speedup) склав 2,28 відносно базової архітектури.

Висновок. У роботі проведено оцінку та оптимізацію інструментів dlib для детекції облич. Доведено, що обмежена продуктивність CPU не дозволяє використовувати нейромережі (CNN) у режимі реального часу. Натомість впровадження технології NVIDIA CUDA забезпечило кратне прискорення: GPU-орієнтований CNN-підхід перевершив класичний метод HOG як за точністю, так і за швидкістю інференсу.

Security vulnerabilities at the Python LLM frameworks boundary

UDC 004.896:004.056.5

Oleksandr Karnaukhov[1], Nataliya Zagorodna[2], Oleh Yarema[3], Oleksandr Revniuk[4]

Ternopil Ivan Puluj National Technical University,
[1]karnaukhov@live.com, [2]zagorodna_n@tntu.edu.ua, [3]yarema.oleh.m@gmail.com, [4]revo0708@gmail.com

To execute complex workflows such as dynamic data analysis, file system management, and database querying, popular Python orchestration frameworks, including LangChain, LlamaIndex, and Ollama, frequently grant LLMs direct access to runtime environments [1]. This architecture often relies on underlying Python utilities like “exec()” or “eval()” to translate model outputs into system actions.

However, this design introduces a critical conflict between traditional deterministic software security and the probabilistic nature of linguistic models. While standard application security relies on strict input sanitization at the system boundary, AI agents inherently process untrusted natural language payloads directly from users.