

- API для взаємодії з веб додатком, яке має надавати набір функцій, що дозволяють читати, записувати в базу даних або відправляти повідомлення до Telegram-боту та бути надійно захищене від зловмисників;

- npm пакет node-telegram бот API для взаємодії з Telegram, що дозволяє встановити web-socket з'єднання, слухати повідомлення, що були відправлені Telegram-боту, а також сповіщення користувачам, що підписані на Telegram-бот. Функціонал даної бібліотеки не структурує користувацький код, тому поверх даної бібліотеки була написана обгортка, що надає можливість використовувати патерн middleware.

Третім компонентом системи є веб додаток, що використовує такі технології: мову програмування TypeScript, фреймворк ReactJS, Material UI [6] – бібліотека компонентів React заснована на Google Material Design [7], Redux [8] для управління станом додатку.

Telegram-бот доступний для ЗВО з вересня 2019 року. Проте задовольнити всі потреби користувача такою малою командою досить важко. Також з плином часу будуть з'являтися нові кейси, що потребуватимуть більше функціонала системи комунікації. Тому дана система буде у вільному доступі і кожен бажаючий зможе покращувати її. Такий підхід дозволить не тільки покращити взаємодію між Telegram ЗВО та адміністрацією університету, але й усіма бажаючим брати участь у розробці та впровадженні своїх ідей в працюючий проект. Можливе додавання таких функцій, як перегляд розкладу заліків та екзаменів, можливість дізнатися який зараз екзамен або який наступний, сповіщення про події, заходи та конференції.

Список використаних джерел

- 1.Telegram APIs. [Електронний ресурс]. – Режим доступу: <https://core.telegram.org/> (дата звернення: 12.03.20). – Назва з екрана.
- 2.TypeScript language. [Електронний ресурс]. – Режим доступу: <https://www.typescriptlang.org/> (дата звернення: 12.03.20). – Назва з екрана.
- 3.Node JS. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/uk/>(дата звернення: 12.03.20). – Назва з екрана.
- 4.MongoDB. [Електронний ресурс]. – Режим доступу: <https://www.mongodb.com/>(дата звернення: 12.03.20). – Назва з екрана.
- 5.ReactJS. [Електронний ресурс]. – Режим доступу: <https://reactjs.org/>(дата звернення: 12.03.20). – Назва з екрана.
- 6.Material UI. [Електронний ресурс]. – Режим доступу: <https://material-ui.com/>(дата звернення: 12.03.20). – Назва з екрана.
- 7.Google Material Design. [Електронний ресурс]. – Режим доступу: <https://material.io/design/>(дата звернення: 12.03.20). – Назва з екрана.
- 8.Redux. [Електронний ресурс]. – Режим доступу: <https://redux.js.org/>(дата звернення: 12.03.20). – Назва з екрана.

УДК 004.93'1

УДОСКОНАЛЕНИЙ АЛГОРИТМ ПОШУКУ ОБЛИЧЧЯ ТА ОЧЕЙ НА БАЗІ КАСКАДНОГО КЛАСИФІКАТОРА ХААРА

Бугай Ю.В., студ. гр. ПІ-161; **Войтенко В.П.**, к.т.н., доцент
Національний університет «Чернігівська політехніка»

Розпізнавання облич – це один із найважливіших напрямів розпізнавання образів, яке, зазвичай, має за мету віднесення вихідних даних до певного класу за допомогою виділення істотних ознак, що характеризують ці дані, із загальної маси даних [1]. Перспективи розвитку методів розпізнавання облич досить значні [2, 3]. Так, сьогодні їхнє використання можна знайти у різних сферах – від безпеки до комерції.

Наразі відомі алгоритми розпізнавання облич мають ряд недоліків, а саме:

- можуть виникати проблеми з виявленням усіх видів облич;
- існують певні труднощі з пошуком людей в окулярах;
- зміни ракурсу призводять до незадовільних результатів.

З цих причин постає мета побудови удосконалених алгоритмів, які би допомагали виправляти вищезгадані недоліки. Завданнями даної роботи є виявлення обличчя за допомогою каскадного класифікатора Хаара та поширення його можливостей на виявлення очей на зображенні.

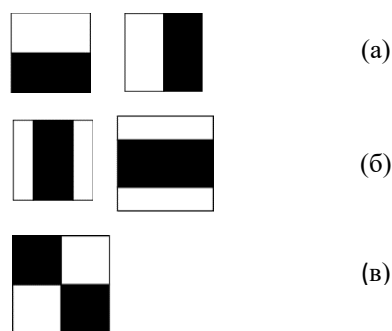
Нині існує велика кількість підходів до виявлення облич. Найкращі результати розпізнавання на сьогодні дає застосування згорткових нейронних мереж. Але розробка їх досить трудомістка, і конкурентоспроможні рішення у змозі запропонувати великі корпорації (як, наприклад, AdaBoost [4]). В той самий час існують алгоритми й рішення, які є сучасними та ефективними для впровадження без значних зусиль.

Виявлення об'єктів за допомогою каскадних класифікаторів на основі класифікатора Хаара є ефективним методом виявлення об'єктів, запропонованим Полом Віолою та Майклом Джонсом у 2001 році [1]. Цей підхід на основі машинного навчання передбачає навчання функції каскаду з використанням великої кількості позитивних і негативних образів. Результат потім використовується для виявлення об'єктів на інших зображеннях.

Даний метод набув великої популярності завдяки високій точності та ґрунтовності теоретичної основи [5]. Основні принципи, на яких базується метод:

- використовується інтегральне уявлення зображення, що дозволяє швидко обчислювати необхідні об'єкти;
- використовуються ознаки Хаара, за допомогою яких відбувається пошук потрібного об'єкта (в цьому контексті, – обличчя та його рис);
- використовується бустінг (комплекс методів) для вибору найбільш вірогідних ознак для шуканого об'єкта на цій частині зображення;
- всі ознаки надходять на вхід класифікатора, який дає результат «так» або «ні»;
- використовуються каскади ознак для швидкого відкидання вікон, де знайдено обличчя.

Основною категорією алгоритму є примітиви Хаара – це ознаки цифрового зображення, які використовуються для розпізнавання об'єктів та працюють на основі вейвлету Хаара. Примітиви являють собою розділені на декілька частин прямокутники, які позиціонують у зображенні (рис. 1). Далі обраховується інтенсивність пікселів кожної області та знаходиться різниця між обрахованими сумами, яка називається значенням даної ознаки Хаара.



а) крайові особливості; б) лінійні особливості; в) кутові особливості

Рисунок 1 – Примітиви Хаара

Процес розпізнавання людини по зображенню обличчя наведено на рис. 2. Він містить такі кроки [4, 5]:

1. Локалізація обличчя на зображенні (знаходження ділянки зображення, що містить обличчя, її розмірів і положення).
2. Вирівнювання знайденого обличчя (за геометрією та яскравістю).
3. Витяг основних ознак обличчя із зображення.
4. Розпізнавання шляхом порівняння вилучених ознак із наявними у базі даних (БД).

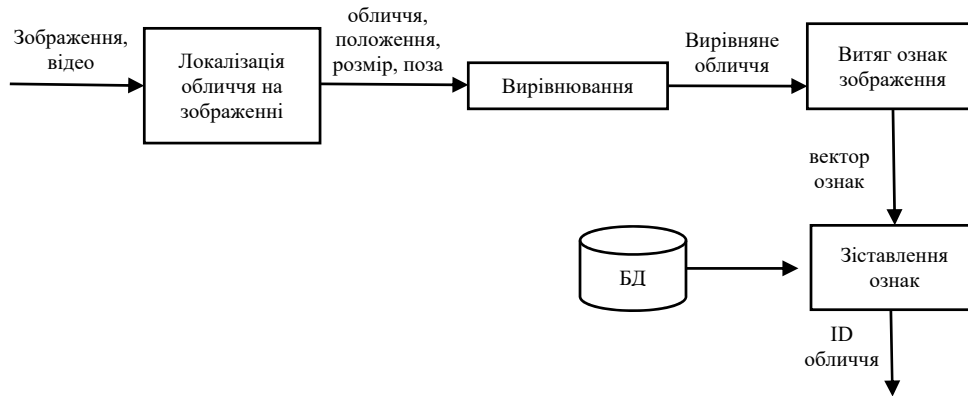


Рисунок 2 – Загальна схема процесу розпізнавання людини за зображенням обличчя

За основу розробки програми розпізнавання обличч було обрано бібліотеку OpenCV, оскільки в ній вже міститься багато заздалегідь підготовлених класифікаторів для обличчя, очей, посмішок тощо. Спочатку було виконане підключення бібліотек OpenCV та NumPy (розширення мови Python). Далі були завантажені необхідні класифікатори XML (haarcascade_frontalface_default.xml – для обличчя, haarcascade_eye.xml – для очей), а також вхідне зображення в режимі відтінків сірого. Перед зчитуванням довелося налаштувати розміри зображення. Після цього знайдені грані на зображенні. Якщо обличчя знайдено, розроблена функція повертає позиції виявлених граней як Rect (x, y, w, h). Як тільки отримані ці місця, можна здійснити виявлення очей на обличчі.

Результат роботи застосунку розпізнавання обличчя та очей представлено на рис. 3.

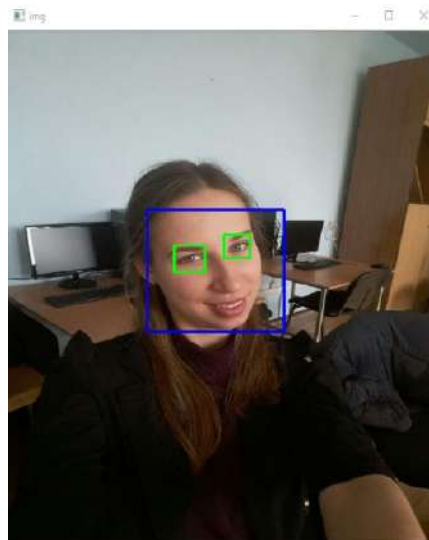


Рисунок 3 – Демонстрація роботи застосунку розпізнавання обличчя та очей

Отже, алгоритм Віоли-Джонса на базі ознак Хаара показав стабільно гарні результати пошуку об'єктів на зображенні та був взятий за основу створення власного застосунку для розпізнавання людей по зображенню їх обличч і очей. Інструментом реалізації було обрано бібліотеку OpenCV з відкритим кодом та мову Python. Це дозволило знайти грані обличчя на зображенні, а після цього визначити розташування очей в межах встановленого квадрату, що допомагає значно прискорити розпізнавання образів.

Список використаних джерел

1. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M.J. Jones // IEEE Conf. On Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001. – V. 1. – P. 511–518.
2. KeyLemon – Face recognition software [Електронний ресурс] – Режим доступу: <http://www.keylemon.com/product/>

3. In the future, can you remain anonymous? [Электронный ресурс] – Режим доступа: http://money.cnn.com/2012/01/13/technology/face_recognition/index.htm

4. Y. Freund, R. E. Schapire. A Short Introduction to Boosting // Journal of Japanese Society for Artificial Intelligence. 1999, №14(5), С. 771-780.

5. DeepFace: Closing the Gap to Human-Level Performance in Face Verification [Электронный ресурс] – Режим доступа: <https://research.facebook.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/>
