

2. Гушко С.В. Управлінські інформаційні системи. Львів : Магнолія Плус, 2006. 320 с.
3. Основи інформаційних систем / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єрємін, О. С. Краєва; За ред. В. Ф. Ситника. К.: КНЕУ, 2001. 420 с.
4. Пономаренко В.С. Інформаційні системи і технології в економіці. К. : Академія, 2005. 456 с.
5. Татарчук М.І. Корпоративні інформаційні системи. К. : КНЕУ, 2006. 560 с.
6. В. М. Михайленко, Т. В. Січко. Моделі і методи автоматизованої системи управління регіонального університетського центру. Вінниця, ВНАУ, 2014.

УДК 004.4(043.2)

*Дикальчук Є.В., студент 2 курсу СО
«Магістр» спеціальності 122 «Комп'ютерні
науки»*

*Парамонов А.І., к.т.н., доцент, доцент
кафедри комп'ютерних наук та
інформаційних технологій*

ПРОГРАМНИЙ ЗАСІБ ІДЕНТИФІКАЦІЇ ОСІБ У ВІДЕОПОТОЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Розпізнавання обличчя – це біометричний процес, який математично відображає риси обличчя людини та зберігає дані у вигляді відбитка обличчя. Він здатний ідентифікувати та розпізнавати людей за цифровим зображенням або відеокадром із джерела відео. Перспективним підходом до розпізнавання обличчя є порівняння вибраних рис обличчя з даного зображення із базою даних відбитків обличчя, тобто, це додаток, який однозначно ідентифікує людину, аналізуючи візерунки на основі текстури та форми обличчя людини. Метою даної роботи є дослідження алгоритмів, методів пошуку та розпізнавання об'єктів у відеопотоці, а саме пошук та розпізнавання обличчя.

При виконанні роботи створено програмний засіб, який виконує пошук та ідентифікацію обличчя у відеопотоці на основі ефективних алгоритмів виявлення і розпізнавання.

На підставі аналізу різноманітних алгоритмів та методів ідентифікації можна виділити загальну структуру процесу розпізнавання обличчя, зображену на рисунку 1.

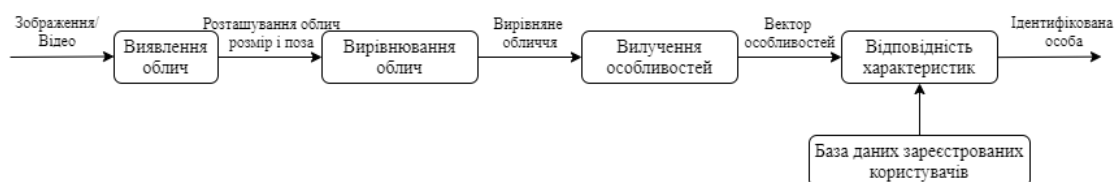


Рисунок 1 – Загальний процес розпізнавання обличчя

Перший етап – це виявлення і локалізація обличчя на кадрі з потокового відео. На другому етапі розпізнавання проводиться вирівнювання зображення обличчя, центрування очей і роту. Далі відбувається вилучення особливих ознак, тобто кодувань обличчя. Останній етап – порівняння вилучених кодувань з наявними в базі даних. Основною відмінністю між алгоритмами розпізнавання буде обчислення ознак і порівняння їх сукупностей між собою.

Для виконання ідентифікації програмою виконуються такі кроки:

- 1) зчитування кадру за кадром з веб камер (бібліотека OpenCV);
- 2) пошук облич у кадрі (гістограма напрямлених градієнтів);
- 3) вирівнювання знайдених облич (лицеві орієнтири (facial landmarks));
- 4) перетворення зображення у 128-мірне значення;
- 5) порівняння обличчя з кадру з наявними обличчями;
- 6) перевірка ідентичності особи.

Спочатку зчитуємо кадр за кадром із веб-камер. Для цього використовуємо бібліотеку OpenCV [1].

Після цього відбувається процес знаходження всіх облич. Для ідентифікації використовується HOG – гістограма направленого градієнта.

Гістограма напрямлених градієнтів (надалі HOG) – це дескриптор функції, що використовується для виявлення об'єктів в комп'ютерному зорі і обробці зображень. Метод дескриптора HOG підраховує входження градієнтної орієнтації в локалізованих частинах вікна виявлення зображення або області інтересу [2].

Переходимо до наступного етапу – нормалізація облич, центрування очей та роту. Отже, після виділення обличчя на зображенні з'являється проблема: одне і те ж обличчя, яке береться до уваги з різних сторін має різний вигляд і для комп'ютера. Щоб взяти це до уваги, необхідно перетворити всі зображення таким чином, щоб окремі риси обличчя (очі, губи і ніс) завжди були відцентрованими на зображеннях. Щоб виконати даний етап беремо за основу алгоритм, який має назву «оцінка антропометричних точок». Основна ідея в тому, що виділяється 68 специфічних точок (міток), що є на кожному обличчі, а саме: внутрішній кут кожної із брів, зовнішні краї очей, частина підборіддя тощо. В такому випадку не беремо до уваги, в яку сторону обличчя повернуте, і можемо очі та рот відцентрувати таким чином, щоб на зображенні вони були фактично в однакових положеннях. Це зробить точність ідентифікації набагато вищою.

Передостанній етап – кодування зображення обличчя. Схема вивчення згорткової нейронної мережі з ціллю вивести сукупність характеристик облич потребує чималої кількості даних і значної продуктивності комп'ютера. У роботі використовується навчена мережа OpenFace [3]. В результаті, все, що потрібно – це провести зображення облич через уже навчену нейронну мережу й у висновку отримати 128 характеристик для різних облич.

Останній етап – ідентифікація обличчя. Для ідентифікації використовуємо Евклідову відстань між двома точками. Щоб порівняти два зображення, створюємо кодування для обох зображень, пропускаючи через модель окремо. Потім можемо використовувати Евклідову відстань, щоб знайти дистанцію, яка

буде меншою для схожих граней і більшою для різних. Для кожного зображення обчислюємо Евклідову відстань між вхідним зображенням і кожного із зображень, представлених в наборі даних. Яке з них має найменшу Евклідову відстань, може розглядатися як зображення тієї ж людини, що й у вхідному зображенні.

Отже, після проведення експериментів та тестування програми слід зауважити, що для досягнення позитивних результатів роботи необхідно хоча б 10 фотографій особи для навчання класифікатора. З такою кількістю фото програма вірно ідентифікує особи при різних поворотах і положеннях голови, будь-якому освітленні, при цьому достатньо лише трохи світла, аби алгоритм зміг визначити краї обличчя.

Список літературних джерел.

1. OpenCV: Home. [Електронний ресурс] – Режим доступу: <https://opencv.org/>.
2. Mark Nixon. Feature Extraction and Image Processing for Computer Vision 3rd Edition. Academic Press, 2012. - 632 pages
3. OpenFace - GitHub Pages. [Електронний ресурс] – Режим доступу: <https://cmusatyalab.github.io/openface/>

УДК 004.4(043.2)

*Зінченко Б.В., студент 3 курсу спеціальності
122 «Комп'ютерні науки»
Січко Т.В., к.т.н., доцент кафедри
комп'ютерних наук та інформаційних
технологій*

ВИКОРИСТАННЯ ДАНИХ ДЛЯ ОПТИМІЗАЦІЇ ПРОЕКТІВ ТА КОДУ .NET НА БАЗІ ІГРОВОГО РУШІЯ UNITY

Донецький національний університет імені Василя Стуса, м. Вінниця

На сьогодні в умовах великої конкуренції у сфері ігор важливим фактором є швидкодія та розмір програмного коду. Проекти оцінюються не тільки в залежності від ідеї та імені компанії, що розробила гру, але й від реалізації. Реалізація може вважатися успішною при умові, що розробнику вдалося втілити усі ідеї проекту та оминати проблеми з помилками у коді.

Першим і найважливішим видом оптимізації програмного продукту є робота з пам'яттю. Скрипти, які пишуться у міжплатформенному середовищі розробки ігор Unity використовують автоматичне керування пам'яттю. Мови програмування, такі як C і C++, навпаки, використовують ручне управління пам'яттю - розробник може безпосередньо зчитувати і записувати дані за вказаними адресами пам'яті і він відповідальний за видалення будь-якого створюваного ним об'єкта. Наприклад, якщо створюються об'єкти в C++, то після