

*Гевко Н.І.¹, Меленчук Л.І.², Шкіра Ю.Р.¹**¹Тернопільський національний економічний університет**²Галицький коледж ім. В. Чорновола***ДОСЛІДЖЕННЯ ЗАСОБІВ СТВОРЕННЯ СИСТЕМ КОМП'ЮТЕРНОГО ЗОРУ**

Вступ. За останні роки комп'ютерний зір став досить популярним і виділився в окремий напрямок. Розробники створюють нові додатки, якими користуються по всьому світу. Технологічні гіганти готові ділитися новими відкриттями та інноваціями з усіма, щоб технології не залишалися привілеєм багатих. Одна з таких технологій - розпізнавання осіб. При правильному і етичному використанні ця технологія може застосовуватися в багатьох сферах життя[1].

Мета: На даний час широко розповсюджені системи комп'ютерного зору з використанням штучного інтелекту та різних алгоритмів опрацювання даних у відеопотоці.

1. Дослідження існуючих систем розпізнавання

Розпізнавання обличчя в соцмережах. Facebook замінив присвоєння тегів зображень вручну на автоматично генеровані пропозиції тегів для кожного зображення, що завантажується на платформу. Facebook використовує простий алгоритм розпізнавання осіб для аналізу пікселів на зображенні і порівняння його з відповідними користувачами.

Розпізнавання осіб у сфері безпеки. Простий приклад використання технології розпізнавання осіб для захисту особистих даних - розблокування смартфона «по обличчю». Таку технологію можна впровадити і в пропускну систему: людина дивиться в камеру, а вона визначає дозволити йому увійти чи ні [2].

Розпізнавання обличчя для підрахунку кількості людей. Технологію розпізнавання осіб можна використовувати при підрахунку кількості людей, які відвідують будь-який захід (наприклад, конференцію або концерт). Замість того щоб вручну підраховувати учасників, ми встановлюємо камеру, яка може захоплювати зображення осіб учасників і видавати загальну кількість відвідувачів. Це допоможе автоматизувати процес і заощадити час.

2. Налаштування системи: вимоги до апаратного та програмного забезпечення

Розглянемо, як можна використовувати технологію розпізнавання осіб, звернувшись до доступних інструментів з відкритим вихідним кодом[3].

Для побудови системи розпізнавання використаємо такі інструменти:

- Веб-камера (Logitech C920) для побудови моделі розпізнавання осіб в реальному часі на ноутбуці Lenovo E470 ThinkPad (Core i5 7th Gen). Можна також використовувати вбудовану камеру ноутбука або відеокамеру з будь-якої зручної системою для аналізу відео в режимі реального часу замість тих, які запропоновані.

- Переважно використовувати графічний процесор для більш швидкої обробки відео.

- Пропонується використовувати операційну систему Ubuntu 18.04 з усім необхідним ПЗ.

3. Налаштування програмного забезпечення

Необхідно перевірити, чи правильно налаштована камера. З Ubuntu це зробити просто: необхідно виявити чи розпізнано пристрій операційною системою. Для цього потрібно виконати кроки що на рисунку 1.

```
analyticsvidhya@analyticsvidhya-ThinkPad-E470:~$ ls /dev/video*  
/dev/video0
```

Рисунок 1. Перевірка апаратного забезпечення

Необхідно підключити веб-камеру і задати команду знову. Якщо веб-камера підключена правильно, новий пристрій буде відображено в результаті виконання команди, що на рисунку 2.

```
analyticsvidhya@analyticsvidhya-ThinkPad-E470:~$ ls /dev/video*  
/dev/video0  
analyticsvidhya@analyticsvidhya-ThinkPad-E470:~$ ls /dev/video*  
/dev/video0 /dev/video1
```

Рисунок 2. Підключення веб камери

Код, зазначений в даній статті, написаний з використанням Python (версія 3.5). Для установки Python рекомендується використовувати Anaconda - популярний дистрибутив Python для обробки і аналізу даних[4].

OpenCV - бібліотека з відкритим кодом, яка призначена для створення додатків комп'ютерного зору. Установка OpenCV проводиться за допомогою pip:

```
pip3 install opencv-python
```

Виконаємо встановлення face_recognition API.

Будемо використовувати face_recognition API, який вважається самим простим API для розпізнавання осіб на Python у всьому світі.

Для установки використовуємо:

```
pip install dlib
```

```
pip install face_recognition
```

Після настройки системи переходимо до впровадження. Для початку, ми створимо базовий код, на основі якого буде побудована система розпізнавання облич.

Необхідно створити файл face_detector.py:

```
# import libraries  
import cv2  
import face_recognition  
# Get a reference to webcam  
video_capture = cv2.VideoCapture("/dev/video1")  
# Initialize variables  
face_locations = []  
while True:  
    # Grab a single frame of video  
    ret, frame = video_capture.read()  
    # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition  
uses)  
    rgb_frame = frame[:, :, ::-1]  
    # Find all the faces in the current frame of video  
    face_locations = face_recognition.face_locations(rgb_frame)  
    # Display the results  
    for top, right, bottom, left in face_locations:  
        # Draw a box around the face
```



```
cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
# Display the resulting image
cv2.imshow('Video', frame)
# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()
```

Потім виконаємо код запусивши файл Python, надрукувавши:

```
python face_detector.py
```

Якщо все працює правильно, відкриється нове вікно з запусивним функцією розпізнавання осіб в реальному часі.

Для створення автоматизованої системи з використанням відеокамери для відстеження, де спікер перебуває в даний момент часу. Залежно від його положення, система повертає камеру так, що спікер завжди залишається в центрі кадру.

Перший крок – необхідно створити систему, яка ідентифікує людини або людей на відео і фокусується на місцезнаходження спікера.

Спочатку імпортуємо необхідні бібліотеки:

```
import cv2
import face_recognition
```

Потім зчитуємо відео і встановлюємо довжину:

```
input_movie = cv2.VideoCapture("sample_video.mp4")
length = int(input_movie.get(cv2.CAP_PROP_FRAME_COUNT))
```

Після цього створюємо файл виведення з необхідним дозволом і швидкістю передачі кадрів, аналогічної тій, що була в файлі введення.

Завантажуємо зображення спікера в якості зразка для розпізнавання його на відео:

```
image = face_recognition.load_image_file("sample_image.jpeg")
face_encoding = face_recognition.face_encodings(image)[0]
known_faces = [
    face_encoding,]
```

Закінчивши, запускаємо цикл, який буде:

- Витягувати кадр з відео.
- Знаходити все обличчя і ідентифікувати їх.
- Створювати нове відео, яке буде поєднувати в собі оригінал кадру із зазначенням місцезнаходження особи спікера з підписом.

Код, який буде це виконувати:

```
# Initialize variables
face_locations = []
face_encodings = []
face_names = []
frame_number = 0
while True:
    # Grab a single frame of video
    ret, frame = input_movie.read()
    frame_number += 1
    # Quit when the input video file ends
    if not ret:
        break
    # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition
```

uses)

```
rgb_frame = frame[:, :, :-1]
# Find all the faces and face encodings in the current frame of video
face_locations = face_recognition.face_locations(rgb_frame, model="cnn")
face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)
face_names = []
for face_encoding in face_encodings:
    # See if the face is a match for the known face(s)
    match = face_recognition.compare_faces(known_faces, face_encoding, tolerance=0.50)
    name = None
    if match[0]:
        name = "Phani Srikanth"
        face_names.append(name)
# Label the results
for (top, right, bottom, left), name in zip(face_locations, face_names):
    if not name:
        continue
    # Draw a box around the face
    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
    # Draw a label with a name below the face
    cv2.rectangle(frame, (left, bottom - 25), (right, bottom), (0, 0, 255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.5, (255, 255, 255), 1)
# Write the resulting image to the output video file
print("Writing frame {} / {}".format(frame_number, length))
output_movie.write(frame)
# All done!
input_movie.release()
cv2.destroyAllWindows()
```

Розроблена система дозволить за допомогою штучного інтелекту розпізнавати обличчя спікера з відео та детектувати його положення в фреймі кадру. Ці базові налаштування є підґрунтям до створення більш гнучких та складних систем забезпечення безпеки та розпізнавання образів.

Висновок. Дослідження існуючих систем розпізнавання образів дозволяє виявити ряд недоліків в існуючих алгоритмах та високу апаратну та програмну складність, що обумовлює необхідність розробки нового ефективного алгоритму розпізнавання образів.

Перелік використаних джерел

1. L.Juwei, N. P.Konstantinos, A. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms", IEEE Transactions On Neural Networks, vol.14, no. 1, pp.117–126, January 2003.
2. M. Lades, J. Vorbruggen, J. Buhmann, "Distortion invariant object recognition in the dynamic link architecture", IEEE Transactions on computers, 1993, vol. 42, no. 3, pp. 300 -310, March 1993.
3. P. Viola, "Robust realtime face detection", International Journal of Computer Vision, 2004, vol. 57, no. 2, pp. 137-154, 2004.
4. А. М. Лисенко, "Застосування біометричних систем для ідентифікації особи", Вісник Київського нац. ун.-ту ім. Т.Шевченка, Юридичні науки, 2004, №60/62, с. 87-91 . [5] Метод Виолы-Джонса (Viola-Jones) как основа для распознавания лиц, Электронный ресурс, Режим доступа:<https://habrahabr.ru/post/133826/>