

Шкіра Ю.Р.¹, Гевко Н.І.¹, Гавришків Н.Г.², Осадчук О.Я.¹

¹Тернопільський національний економічний університет

²Галицький коледж ім. В. Чорновола

АЛГОРИТМИ ТА ЗАСОБИ АВТОМАТИЧЕНОЇ ДЕТЕКЦІЇ ОБЛИЧЧЯ В ВІДЕОПОТОЦІ

Вступ. Розпізнавання обличчя - останній тренд в авторизації користувача. Apple використовує Face ID, OnePlus - технологію Face Unlock. Baidu використовує розпізнавання особи замість ID-карт для забезпечення доступу в офіс, а при повторному перетині кордону в ОАЕ вам потрібно тільки подивитися в камеру[1].

В роботі буде запропоновано, алгоритми для створення мережі розпізнавання осіб за допомогою використання FaceNet.

Мета: Дослідження можливості побудови системи розпізнавання образів за допомогою FaceNet.

1. Дослідження можливостей побудови систем розпізнавання образів за допомогою FaceNet

FaceNet - нейронна мережа, яка вчиться перетворювати зображення особи в компактний евклідовий простір, де дистанція відповідає мірі схожості осіб. Простіше кажучи, чим більше схожі особи, тим вони ближче[2].

FaceNet використовує особливу функцію втрат звану TripletLoss. Вона мінімізує дистанцію між якорем і зображеннями, які містять схожу зовнішність, і максимізує дистанції між різними згідно формули:

$$Loss = \sum_{i=1}^N [\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + a] \quad (1)$$

де: $f(a)$ це енкодинг якоря

$f(p)$ це енкодинг схожих осіб (positive)

$f(n)$ це енкодинг несхожих осіб (negative)

a - це константа, яка дозволяє бути впевненим, що мережа не буде намагатися оптимізувати безпосередньо

$$f(a) - f(p) = f(a) - f(n) = 0$$

[...] + еквівалентно $\max(0, sum)$

FaceNet - сіамська мережа. Сіамська мережа - тип архітектури нейромережі, який навчається дифференціюванню вхідних даних. Тобто, дозволяє навчитися розуміти які саме зображення схожі, а які ні(рисунок 1).

Сіамські мережі складаються з двох ідентичних нейронних мереж, кожна з яких має однакові точні ваги. По-перше, кожна мережа приймає одне з двох вхідних зображень в якості вхідних даних. Потім виходи останніх шарів кожної мережі відправляються в функцію, яка визначає, чи містять зображення однакові ідентифікатори. У FaceNet це робиться шляхом обчислення відстані між двома виходами[3].

Siamese network

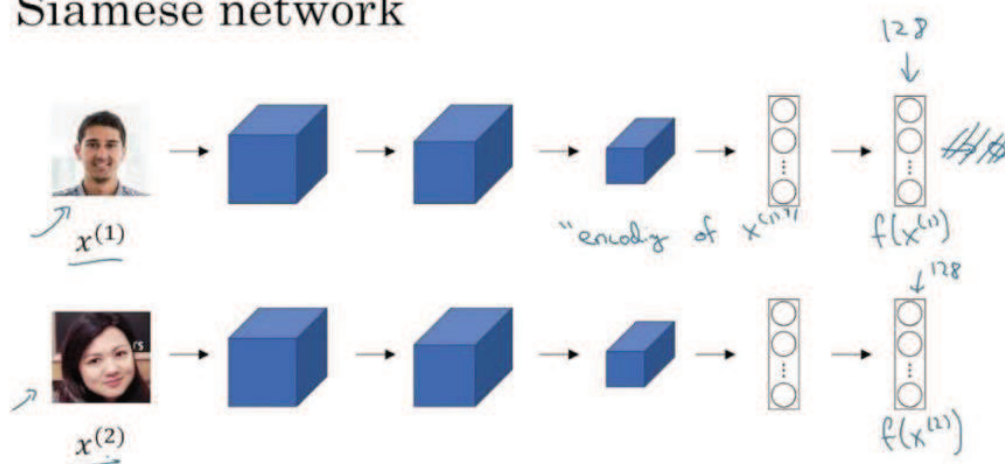


Рисунок 1 – Схема роботи Сіамських мереж

2. Реалізація систем розпізнавання образів за допомогою FaceNet

У реалізації будемо використовувати Keras і Tensorflow. Крім того, будемо використовувати два файли утиліти зі сховища [deeplaining.ai](https://github.com/davlatov/deeplaining.ai), щоб абстрагуватися від взаємодій з мережею FaceNet[4].

Модуль `fr_utils.py`, який містить функції для подачі зображень в мережу і отримання кодування зображень та `inception_blocks_v2.py`, котрий містить функції для приготування та компіляції мережі FaceNet[5].

Перше, що нам потрібно зробити, це зібрати мережу FaceNet для нашої системи розпізнавання осіб.

```
import os
import glob
import numpy as np
import cv2
import tensorflow as tf
from fr_utils import *
from inception_blocks_v2 import *
from keras import backend as K
K.set_image_data_format('channels_first')
FRmodel = faceRecoModel(input_shape = (3, 96, 96))
def triplet_loss(y_true, y_pred, alpha = 0.3):
    anchor, positive, negative = y_pred[0], y_pred[1], y_pred[2]
    pos_dist = tf.reduce_sum(tf.square(tf.subtract(anchor,
        positive)), axis = -1)
    neg_dist = tf.reduce_sum(tf.square(tf.subtract(anchor,
        negative)), axis = -1)
    basic_loss = tf.add(tf.subtract(pos_dist, neg_dist), alpha)
    loss = tf.reduce_sum(tf.maximum(basic_loss, 0.0))
    return loss
FRmodel.compile(optimizer = 'adam', loss = triplet_loss, metrics = ['accuracy'])
load_weights_from_FaceNet(FRmodel)
```

Почнемо ініціалізацію мережі зі входу розмірності (3, 96, 96). Це означає, що

картинка передається у вигляді трьох каналів RGB і розмірності 96×96 пікселів.

Тепер визначимо Triplet Loss функцію. Функція в сніпеті коду вище задовольняє рівняння Triplet Loss, яке було попередньо визначено[6].

Відразу після того, як визначили функцію втрат, можна скопіювати систему розпізнавання осіб за допомогою Keras. Будемо використовувати Adam optimizer для мінімізації втрат, підрахованих за допомогою функції Triplet Loss.

Тепер коли система скопійована, FaceNet потрібно підготувати базу даних особистостей, для яких мережа буде розпізнавати обличчя. Будемо використовувати всі зображення, які лежать в директорії images[7].

Будемо використовувати по одному зображення на людину в реалізації. FaceNet досить потужна, щоб розпізнати людину по одній фотографії.

```
def prepare_database():
    database = {}
    for file in glob.glob("images/*"):
        identity = os.path.splitext(os.path.basename(file))[0]
        database[identity] = img_path_to_encoding(file, FRmodel)
    return database
```

Для кожного зображення перетворимо дані зображення в 128 float числа. Цим займається функція `img_path_to_encoding`. Функція приймає на вхід шлях до зображення і опрацьовує зображення для мережі, після чого повертають результати роботи мережі[8].

Як тільки отримано закодоване зображення в базі даних, мережа готова приступити до розпізнавання[9].

Як вже обговорювалося раніше, FaceNet намагається мінімізувати відстань між схожими зображеннями і максимізувати між різними. Запропонована реалізація використовує дану інформацію для того, щоб визначити, чи є людина на новому зображенні.

```
def who_is_it(image, database, model):
    encoding = img_to_encoding(image, model)

    min_dist = 100
    identity = None

    # Loop over the database dictionary's names and encodings.
    for (name, db_enc) in database.items():
        dist = np.linalg.norm(db_enc - encoding)
        print('distance for %s is %s' % (name, dist))
        if dist < min_dist:
            min_dist = dist
            identity = name

    if min_dist > 0.52:
        return None
    else:
        return identity
```

Завантажуємо нове зображення в функцію `img_to_encoding`. Функція обробляє зображення, використовуючи FaceNet і повертає закодоване зображення[10]. Тепер можемо зробити припущення про найбільш імовірною особистості цієї людини.

Для цього підраховуємо відстань між отриманим новим зображенням і кожною людиною в нашій базі даних. Найменша дистанція вкаже на найбільш ймовірну особистість людини[11].

Необхідно визначити чи дійсно збігаються особистості на зображенні і в базі. Наступна частина коду для цього:

```
if min_dist > 0.52:  
    return None  
else:  
    return identity
```

Число 0.52 отримано методом проб і помилок. Для мережі це число може відрізнитись, залежно від реалізації і даних[12-13].

Висновок. Було проведено дослідження існуючих систем та технологій розпізнавання осіб, що дозволило розробити ефективну систему розпізнавання образів, використовуючи попередньо підготовлену версію алгоритму FaceNet на python.

Перелік використаних джерел

1. Гонсалес Р., Вудс Р. Цифровая обработка изображений/ Пер. с англ. – М.: Техносфера, 2005. – 1072 с.
2. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. - Москва: Техносфера, 2006. - 616с.
3. Ту Дж., Гонсалес Р. Принципы распознавания образов.- М.:Мир, 1998.-411 с.
4. Шлезингер М.И., Главач В. Десять лекций по статистическому и структурному распознаванию. – Киев: Наук. думка, 2004, 546 с.
5. Введение в MATLAB: Учеб. пособие/ Л. А. Мироновский, К. Ю. Петрова; ГУАП. – СПб., 2006. – 164 с.
6. Фурман Я.А. Введение в контурный анализ. Приложения к обработке изображений и сигналов/Я. А. Фурман, А. В. Кревецкий, А. К. Передреев, А. А. Роженцов, Р. Г. Хафизов, И. Л. Егошина, А. Н. Леухин/ М.: ФИЗМАТЛИТ, 2003, 592стр.
7. William K. Pratt Digital image processing/ Third Edition/ John Wiley & Sons, Inc. – 2001. – 723 с/
8. Поспелов Д.А. Искусственный интеллект. Справочник. Книга 2. Модели и методы/М.: Радио и связь, 1990. — 304 с.
9. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения/Харьков: Телетех, 2004. – 369с.
10. Голд Б., Рейдер Ч. Цифровая обработка сигналов.- М.: Сов. радио, 1973. - 568 с.
11. Крылов В.Н., Максимов М.В. Вторичные преобразователи сигналов и изображений. - Одесса, Астропринт, 1997. - 176 с.
12. Гмурман В.Е. Теория вероятностей и математическая статистика. 6-е изд.-М.: Высш. шк., 1998.- 479 с.
13. Федотов Н.Г. Методы стохастической геометрии в распознавании образов- М.:Радио и связь, 1990.- 144 с.