

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Звіт

Лабораторна робота № 4 з дисципліни
«Штучний інтелект в задачах обробки зображень»

«Розпізнавання людини на фото з використанням бібліотеки Dlib»

Виконав(ла)

ІП-01 Черпак А. В.

(шифр, прізвище, ім'я, по батькові)

Перевірів(ла)

Нікітін В. А.

(прізвище, ім'я, по батькові)

Київ 2023

Завдання

1. Зробити розпізнавання будь-якої “зірки”
2. Зробити розпізнавання себе.

Мета

Навчитися розпізнавати обличчя на фото використовуючи навчені нейронні мережі

Хід роботи

1. Імпортуємо необхідні бібліотеки та створимо допоміжні функції для пошуку усіх файлів, що належать заданим класам, пошуку та обробки обличчя, а також аналізу співпадінь:

```
import os
import cv2
import glob
import dlib
import numpy

face_detector = dlib.get_frontal_face_detector()
shape_predictor =
dlib.shape_predictor("models/shape_predictor_68_face_landmarks.dat")
face_encoder =
dlib.face_recognition_model_v1("models/dlib_face_recognition_resnet_model_v1.dat"
)

VALID_EXTENSIONS = ['.jpg', 'jpeg']

def get_image_paths(root_dir, class_names):
    image_paths = []
    for class_name in class_names:
        class_dir = os.path.sep.join([root_dir, class_name])
        class_file_paths = glob.glob(os.path.sep.join([class_dir, '*.*']))
        for file_path in class_file_paths:
            ext = os.path.splitext(file_path)[1]
            if ext.lower() not in VALID_EXTENSIONS:
                print("Skipping file: {}".format(file_path))
                continue
            image_paths.append(file_path)
    return image_paths

def face_rects(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    rects = face_detector(gray, 1)
    return rects

def face_landmarks(image):
    return [shape_predictor(image, face_rect) for face_rect in face_rects(image)]

def face_encodings(image):
    return [numpy.array(face_encoder.compute_face_descriptor(image,
face_landmark)) for face_landmark in
face_landmarks(image)]

def nb_of_matches(known_encodings, unknown_encoding):
    distances = numpy.linalg.norm(known_encodings - unknown_encoding, axis=1)
    small_distances = distances <= 0.6
    return sum(small_distances)
```

2. Створимо функцію для генерації вкладень до кожного обличчя:

```
import os
import cv2
from Lab4_utils import get_image_paths, face_encodings
import pickle

root_dir = 'dataset'
class_names = os.listdir(root_dir)
image_paths = get_image_paths(root_dir, class_names)
name_encoding_dict = {}
```

```

nb_current_image = 1
for image_path in image_paths:
    print(f"Image processed {nb_current_image}/{len(image_paths)}")
    image = cv2.imread(image_path)
    encodings = face_encodings(image)
    name = image_path.split(os.path.sep)[-2]
    e = name_encoding_dict.get(name, [])
    e.extend(encodings)
    name_encoding_dict[name] = e
    nb_current_image += 1
with open("encodings.pickle", "wb") as f:
    pickle.dump(name_encoding_dict, f)

```

Запустимо описану вище функцію, і поглянемо на вивід:

```

D:\Programming\py\python.exe "D:\Education\6 sem\Shizoid\Lab4\Lab4_faceEncoding.py"
Image processed 1/18
Image processed 2/18
Image processed 3/18
Image processed 4/18
Image processed 5/18
Image processed 6/18
Image processed 7/18
Image processed 8/18
Image processed 9/18
Image processed 10/18
Image processed 11/18
Image processed 12/18
Image processed 13/18
Image processed 14/18
Image processed 15/18
Image processed 16/18
Image processed 17/18
Image processed 18/18

Process finished with exit code 0

```

Рисунок 1 – Результати генерації вкладень

Як бачимо, все відпрацювало без помилок. Тепер створимо за стосунок для розпізнавання людей по фото.

```

from Lab4_utils import *
import pickle
with open("encodings.pickle", "rb") as f:
    name_encodings_dict = pickle.load(f)

imagePoroshenko = cv2.imread("examples/1.jpg")
imageArnold = cv2.imread("examples/2.jpg")
imageMultiple = cv2.imread("examples/3.jpg")
imagePoroh = cv2.imread("examples/5.jpg")
imageMe = cv2.imread("examples/4.jpg")

images = [imagePoroshenko, imageArnold, imageMultiple, imageMe, imagePoroh]

for i in range(len(images)):
    encodings = face_encodings(images[i])
    names = []
    for encoding in encodings:

```

```

counts = {}
for (name, encodings) in name_encodings_dict.items():
    counts[name] = nb_of_matches(encodings, encoding)
if all(count == 0 for count in counts.values()):
    name = "Unknown"
else: name = max(counts, key=counts.get)
names.append(name)
for rect, name in zip(face_rects(images[i]), names):
    x1, y1, x2, y2 = rect.left(), rect.top(), rect.right(), rect.bottom()
    cv2.rectangle(images[i], (x1, y1), (x2, y2), (0, 255, 0), 2)
    cv2.putText(images[i], name, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0, 255, 0), 2)
cv2.imshow("image", images[i])
cv2.waitKey(0)

```

При навчанні були використані фотографії Порошенка, а також мої фото. Тому протестуємо систему спершу за допомогою фотографії Петра Олексійовича, потім – зображення Шварцнегера, далі – спільного зображення Порошенка, Зеленського та Тимошенко. Очевидно, система повинна розпізнати лише Петра Олексійовича, а інших позначити як Unknown.

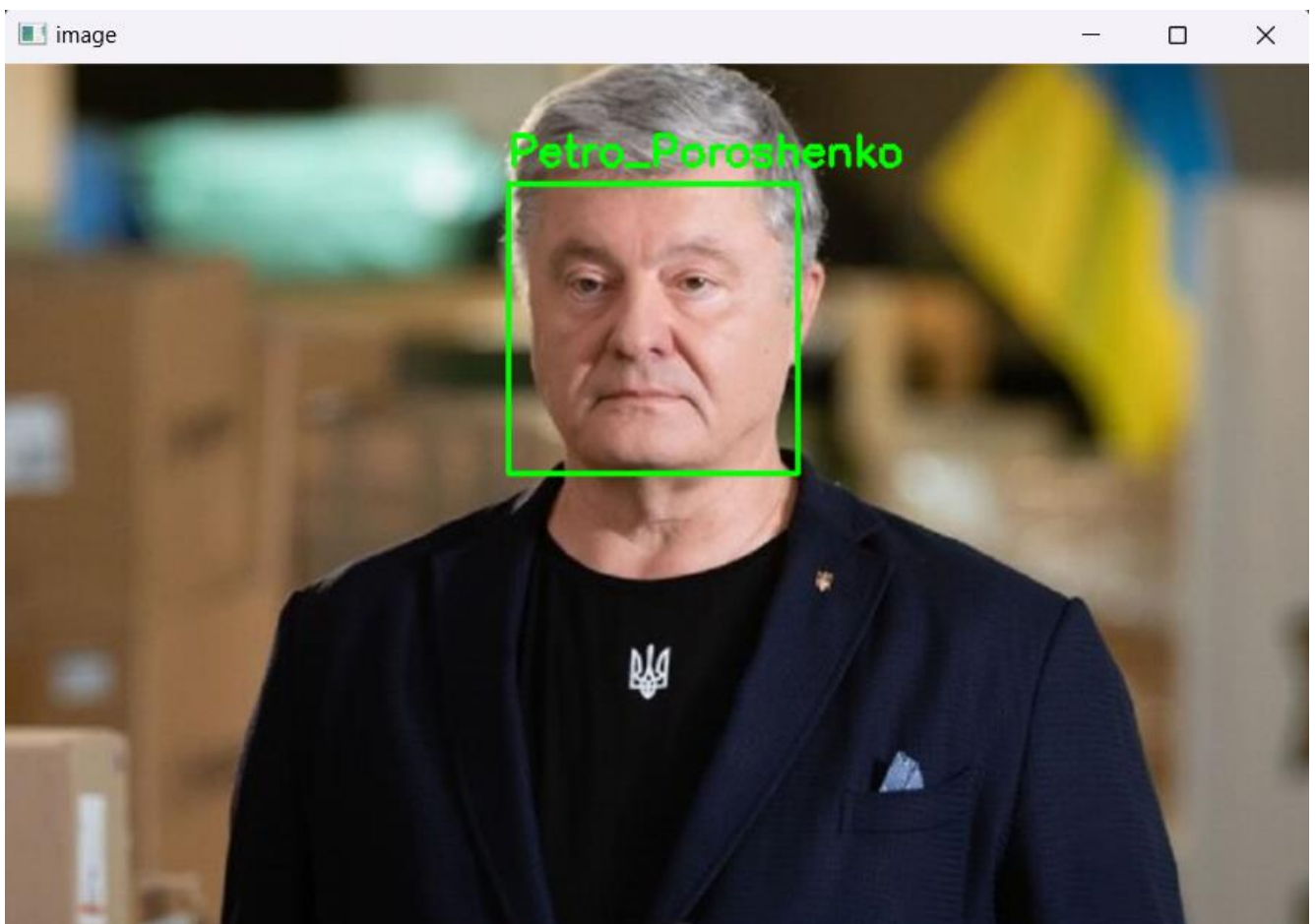


Рисунок 2 – Результат розпізнавання Петра Олексійовича

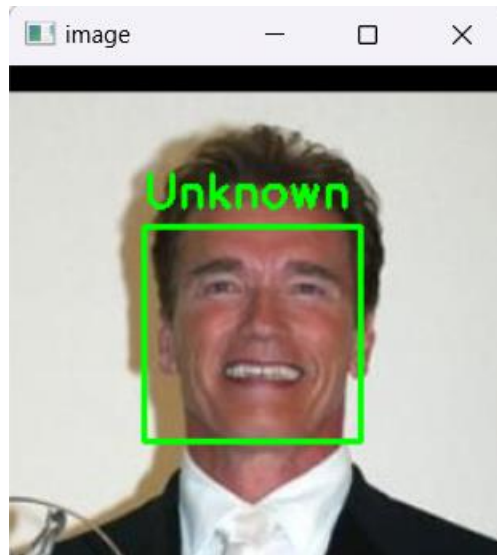


Рисунок 3 – Результат розпізнавання Шварцнегера – не розпізнано

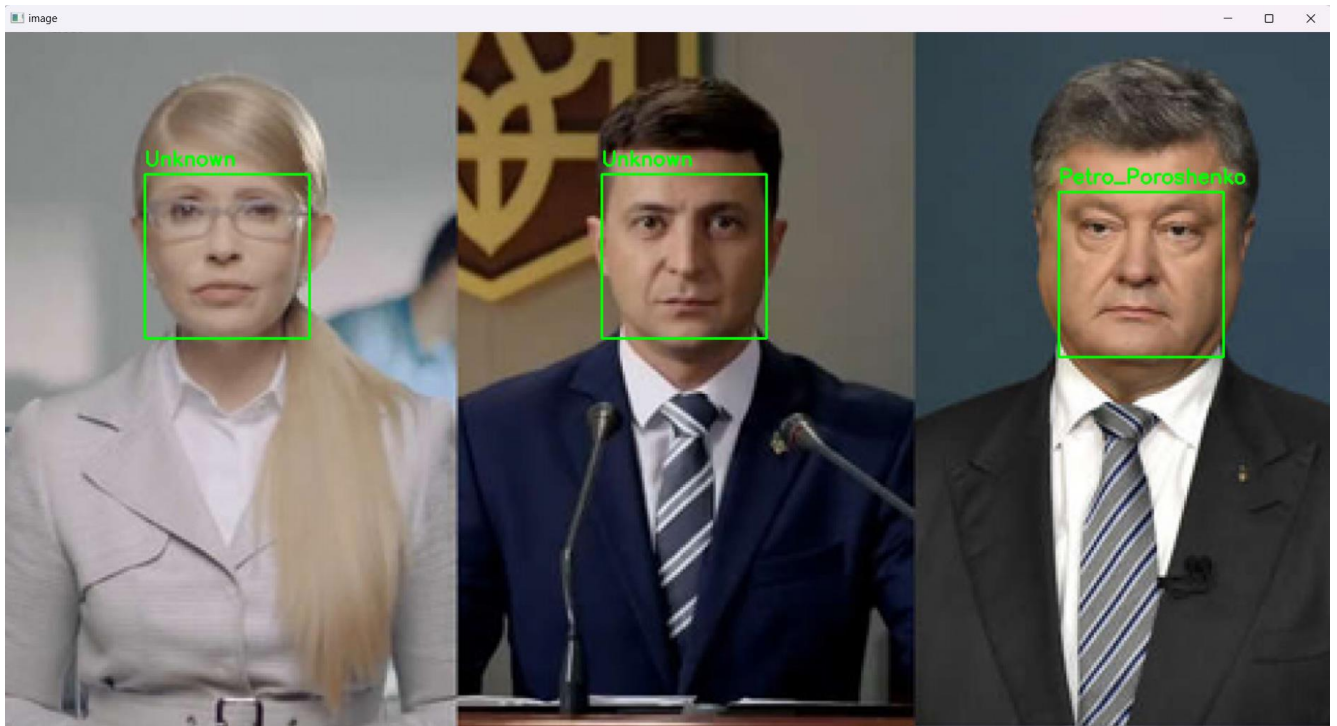


Рисунок 4 – Результат розпізнавання на фото з групою осіб – розпізнано лише тих, на кому система навчалася.

Після цього передамо у систему нашу з дівчиною спільну фотографію. Знову ж, мене повинно розпізнати, а дівчину – ні.

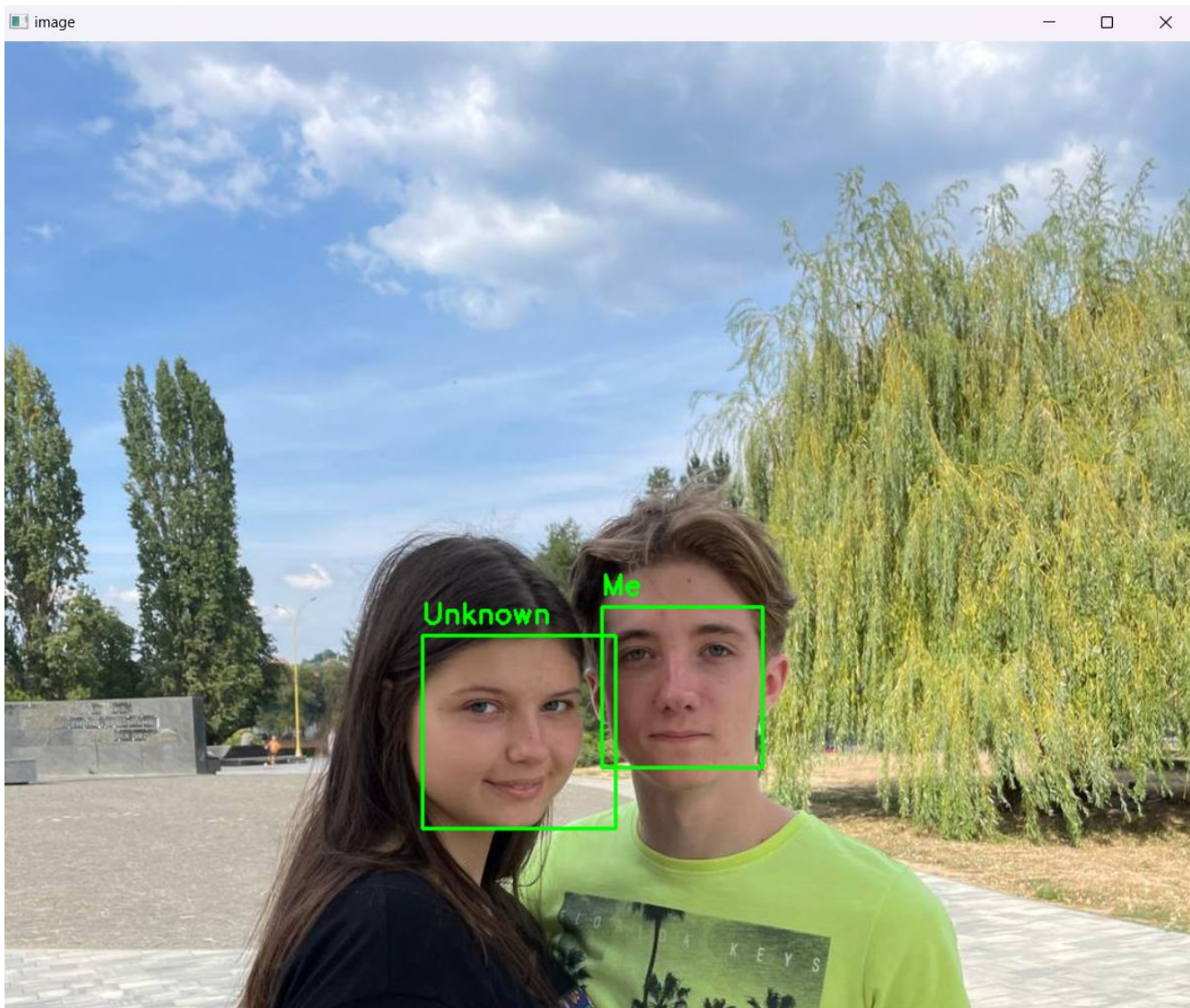


Рисунок 5 – Результати розпізнавання нас з дівчиною

Тепер цікавості заради передамо викривлене зображення і спробуємо його класифікувати.

Хто переможе? порошороп чи окнешенко

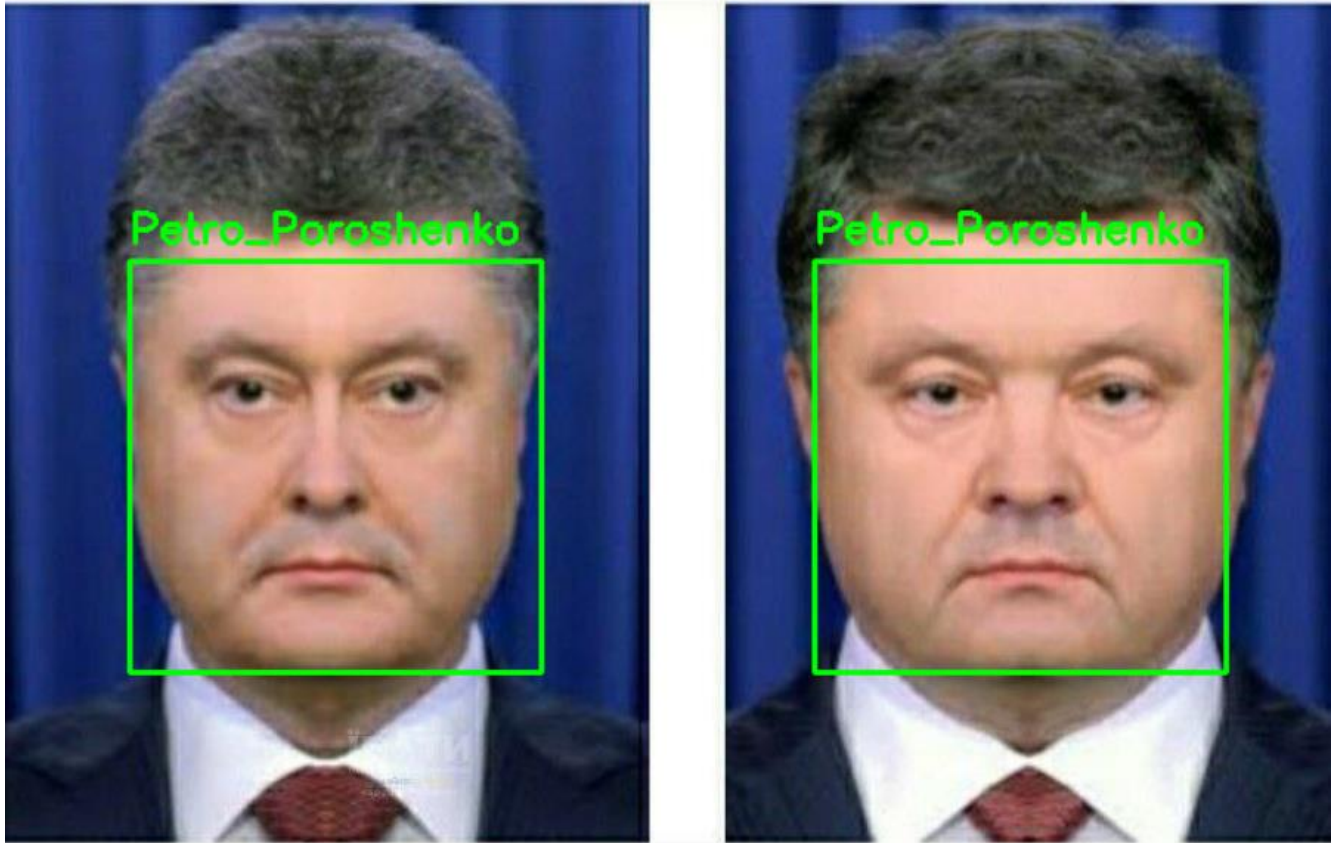


Рисунок 6 – Розпізнавання викривлених зображень

Як бачимо, навіть на частково викривлених фотографіях система здатна розпізнавати обличчя людини.

Контрольні запитання:

1. Розпізнавання обличчя – це процес ідентифікації людини за рисами її обличчя. Ця технологія використовується при біометричному розблокуванні екрану, ідентифікації при накладанні електронного підпису, а також у криміналістиці для визначення особи правопорушника чи потерпілого.
2. Для розпізнавання обличчя спершу потрібно знайти область зображення, у якій воно розташоване, потім вирівняти його за допомогою спеціальних орієнтирів, далі кодувати, а вже лиш тоді .
3. Згорткові нейронні мережі – це різновид нейронних мереж, що містять у тому числі згорткові шари для обробки зображень (двовимірних чи багатовимірних). У даній лабораторній розглядалися сіамські мережі, тобто система з двох абсолютно однакових мереж, через які пропускаються пари зображень і порівнюються результати.
4. Для навчання мережі необхідно подати на вхід ціле дерево зображень, що міститиме як позитивні, так і негативні семпли. Один з позитивних обирається «якірним», тобто подальше навчання відштовхуватиметься саме від нього. При кодуванні для кожного зображення створюється вкладення, і вже керуючись даними з цих вкладень мережа намагається віднайти такі ваги параметрів, при яких відстань між вкладеннями однієї людини буде мінімальною, а для різних людей – максимальною.
5. Dlib – кросплатформна бібліотека, написана на C++, що складається з багатьох компонент різного призначення. У даній роботі ми використовували її для отримання детектора облич, вирівнювання за орієнтирами, а також безпосередньо для розпізнавання облич.