Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 2

По дисциплине СМЗКС

Тема: «Биометрическая аутентификация»

Выполнил:

Студент 4 курса

Группы ИИ-16

Журавлёв В.А.

Проверил:

Хацкевич М.В.

Брест, 2022

**Цель работы:** приобретение практических навыков использования биометрической аутентификации.

**Вариант:**

По лицу.

**Код программы:**

import face\_recognition

from PIL import Image, ImageDraw

import pickle

from cv2 import cv2

def face\_rec():

gal\_face\_img = face\_recognition.load\_image\_file("img/gal\_gadot.jpg")

gal\_face\_location = face\_recognition.face\_locations(gal\_face\_img)

justice\_league\_img = face\_recognition.load\_image\_file("img/justice\_league\_actors.jpg")

justice\_league\_faces\_locations = face\_recognition.face\_locations(justice\_league\_img)

# print(gal\_face\_location)

# print(justice\_league\_faces\_locations)

# print(f"Found {len(gal\_face\_location)} face(s) in this image")

# print(f"Found {len(justice\_league\_faces\_locations)} face(s) in this image")

pil\_img1 = Image.fromarray(gal\_face\_img)

draw1 = ImageDraw.Draw(pil\_img1)

for(top, right, bottom, left) in gal\_face\_location:

draw1.rectangle(((left, top), (right, bottom)), outline=(255, 255, 0), width=4)

del draw1

pil\_img1.save("img/new\_gal1.jpg")

pil\_img2 = Image.fromarray(justice\_league\_img)

draw2 = ImageDraw.Draw(pil\_img2)

for(top, right, bottom, left) in justice\_league\_faces\_locations:

draw2.rectangle(((left, top), (right, bottom)), outline=(255, 255, 0), width=4)

del draw2

pil\_img2.save("img/new\_justice\_league.jpg")

def extracting\_faces(img\_path):

count = 0

faces = face\_recognition.load\_image\_file(img\_path)

faces\_locations = face\_recognition.face\_locations(faces)

for face\_location in faces\_locations:

top, right, bottom, left = face\_location

face\_img = faces[top:bottom, left:right]

pil\_img = Image.fromarray(face\_img)

pil\_img.save(f"img/{count}\_face\_img.jpg")

count += 1

return f"Found {count} face(s) in this photo"

def compare\_faces(img2\_path):

users = {"img/zhuravlev\_vlad.jpeg": "Zhuravlev Vlad",

"img/gal\_gadot.jpg": "Gal Gadot",

"img/dicaprio.jpg": "Dicaprio"}

img2 = face\_recognition.load\_image\_file(img2\_path)

img2\_encodings = face\_recognition.face\_encodings(img2)[0]

is\_found = False

for key in users:

img1 = face\_recognition.load\_image\_file(key)

img1\_encodings = face\_recognition.face\_encodings(img1)[0]

# print(img1\_encodings)

result = face\_recognition.compare\_faces([img1\_encodings], img2\_encodings)

if result[0]:

print("Hello, " + users[key] + "!!!")

is\_found = True

break

# print(result)

if not is\_found:

print("Sorry, such user not found...")

def main():

# face\_rec()

# print(extracting\_faces("img/justice\_league\_actors.jpg"))

print("Enter path to image: ")

file\_name = input()

compare\_faces(file\_name)

if \_\_name\_\_ == '\_\_main\_\_':

main()

Вывод: приобрел практические навыки использования биометрической аутентификации.