МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем і мереж

Лабораторна робота №1

на тему:

“Введення в Python”

з дисципліни

“Екстремальне програмування”

Виконала:

студентка групи КН-311

Рибалкіна Марина

Перевірив:

к.т.н.

Щербак С.С.

Львів 2020

**Мета роботи**: поставити середовище розробки та ознайомитись з Python.

**Теоретичний матеріал**:

Python (найчастіше вживане прочитання — «Па́йтон», запозичено назву з британського шоу Монті Пайтон) — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

**Хід роботи**:

1. Спершу було встановлено на персональний комп’ютер інтерпретатор Python 3.6. Було завантажено за посиланням: <https://www.python.org/downloads/>
2. Далі було завантажено декілька необхідних модулів у терміналі Python у середовищі pep8. Команди є наступними:

pip install imageai

pip install tensorflow=1.8

pip install opencv-python

pip install keras

pip install scipy

pip install numpy

pip install matplotlib

У результаті модулі були успішно завантажені.

Було розроблено UI частину:

import time

import tkinter

import PIL.Image

import PIL.ImageTk

import cv2

from tkinter.filedialog import askopenfilename

from MyVideoRecongnition import MyVideoRecognition

from \_VideoCapture import \_VideoCapture

import threading

class FileType:

def \_\_init\_\_(self, extension=".mp4", file\_types=[('Video files', '\*.mp4'), ('All files', '\*.\*')]):

self.default\_extension = extension

self.file\_types = file\_types

class HUD:

def \_\_init\_\_(self, window, window\_title, b\_start\_loop=True):

self.window = window

self.window.title(window\_title)

self.init(window)

self.delay = 1

self.is\_video\_paused = False

self.video = None

if b\_start\_loop:

self.start\_loop()

def start\_loop(self):

self.update()

self.window.mainloop()

def init(self, window):

self.canvas = tkinter.Canvas(window, width=700, height=400)

self.canvas.pack()

self.create\_text(window)

self.create\_buttons(window)

def create\_text(self, window):

self.text = tkinter.Label(window, text='')

self.text.pack()

def create\_buttons(self, window):

button\_size = 15

self.button\_play = tkinter.Button(window, text='Play', width=button\_size)#, command=self.play\_video\_clicked)

self.button\_play.pack(side=tkinter.LEFT) # , expand=True)

self.button\_pause = tkinter.Button(window, text='Pause', width=button\_size)#, command=self.pause\_video\_clicked)

self.button\_pause.pack(side=tkinter.LEFT) # , expand=True)

self.button\_choose\_file = tkinter.Button(window, text='Choose video-file', width=button\_size)

self.button\_choose\_file.pack(side=tkinter.LEFT)

self.button\_choose\_builtin\_camera = tkinter.Button(window, text='Choose my webcam', width=button\_size)

self.button\_choose\_builtin\_camera.pack(side=tkinter.LEFT)

self.button\_analysis = tkinter.Button(window, text='Analysis', width=button\_size)

self.button\_analysis.pack(side=tkinter.LEFT)

self.button\_play\_analyzed = tkinter.Button(window, text='Play Analyzed', width=button\_size)

self.button\_play\_analyzed.pack(side=tkinter.LEFT)

self.button\_snapshot = tkinter.Button(window, text='Snapshot!', width=button\_size, command=self.take\_snapshot)

self.button\_snapshot.pack(side=tkinter.LEFT)

def update(self):

if not self.is\_video\_paused:

self.capture\_video()

self.window.after(self.delay, self.update)

def capture\_video(self):

if self.video is not None:

ret, frame = self.video.get\_frame()

if ret:

image = PIL.Image.fromarray(frame)

image.thumbnail((750, 750), PIL.Image.ANTIALIAS)

self.photo = PIL.ImageTk.PhotoImage(image=image)

self.canvas.create\_image(0, 0, image=self.photo, anchor=tkinter.NW)

def stop\_recognition(self):

self.video = None

print("stop\_recognition")

def play\_video(self, path=0):

self.video = \_VideoCapture(path)

print("play\_video")

def set\_is\_video\_paused(self, is\_paused):

self.is\_video\_paused = is\_paused

def take\_snapshot(self):

if self.video is not None:

ret, frame = self.video.get\_frame()

if ret:

cv2.imwrite("frame-" + time.strftime("%d-%m-%Y-%H-%M-%S") + ".jpg", cv2.cvtColor(frame, cv2.COLOR\_RGB2BGR))

class MyApp:

def \_\_init\_\_(self, b\_should\_start\_looping=False):

self.analysed\_video\_name = ""

self.count\_analyzed\_video = 0

self.default\_dialog = FileType()

self.path\_to\_opened\_video = ""

self.thread1 = None

self.my\_hud = HUD(tkinter.Tk(), 'VideoAnalysis', False)

self.init\_buttons\_behavior()

if b\_should\_start\_looping:

self.my\_hud.start\_loop()

def choose\_file\_clicked(self):

self.open\_dialog(self.default\_dialog)

# OMP\_NUM\_THREADS

def analyse\_clicked(self):

self.increment\_analysed\_video\_counter()

self.stop\_thread()

if self.path\_to\_opened\_video:

self.thread1 = threading.Thread(target=MyVideoRecognition,

args=(self.analysed\_video\_name, self.path\_to\_opened\_video, 'fromVideo', self.stop\_thread))

self.thread1.start()

else:

self.thread1 = threading.Thread(target=MyVideoRecognition, args=(self.analysed\_video\_name, 0, 'fromCamera', self.stop\_thread))

self.thread1.start()

print("analyse\_clicked")

def increment\_analysed\_video\_counter(self):

self.count\_analyzed\_video += 1

self.analysed\_video\_name = 'AnalyzedVideo' + str(self.count\_analyzed\_video)

def play\_analysed\_video(self):

if len(self.analysed\_video\_name) != 0:

self.my\_hud.play\_video(self.analysed\_video\_name + ".avi")

print("play\_analysed\_video")

def init\_buttons\_behavior(self):

self.my\_hud.button\_choose\_file['command'] = self.choose\_file\_clicked

self.my\_hud.button\_analysis['command'] = self.analyse\_clicked

self.my\_hud.button\_play\_analyzed['command'] = self.play\_analysed\_video

self.my\_hud.button\_choose\_builtin\_camera['command'] = self.turn\_camera\_on\_clicked

self.my\_hud.button\_play['command'] = self.play\_video\_clicked

self.my\_hud.button\_pause['command'] = self.pause\_video\_clicked

def turn\_camera\_on\_clicked(self):

self.path\_to\_opened\_video = ""

self.my\_hud.play\_video()

if self.my\_hud.is\_video\_paused:

self.my\_hud.capture\_video()

def open\_dialog(self, file\_type):

self.path\_to\_opened\_video = askopenfilename(defaultextension=file\_type.default\_extension,

filetypes=file\_type.file\_types)

if self.path\_to\_opened\_video is None:

self.path\_to\_opened\_video = ""

if len(self.path\_to\_opened\_video) > 0:

self.my\_hud.play\_video(self.path\_to\_opened\_video)

if self.my\_hud.is\_video\_paused:

self.my\_hud.capture\_video()

def pause\_video\_clicked(self):

self.my\_hud.set\_is\_video\_paused(True)

self.stop\_thread()

def play\_video\_clicked(self):

self.my\_hud.set\_is\_video\_paused(False)

if len(self.path\_to\_opened\_video) <= 0:

self.turn\_camera\_on\_clicked()

def stop\_thread(self):

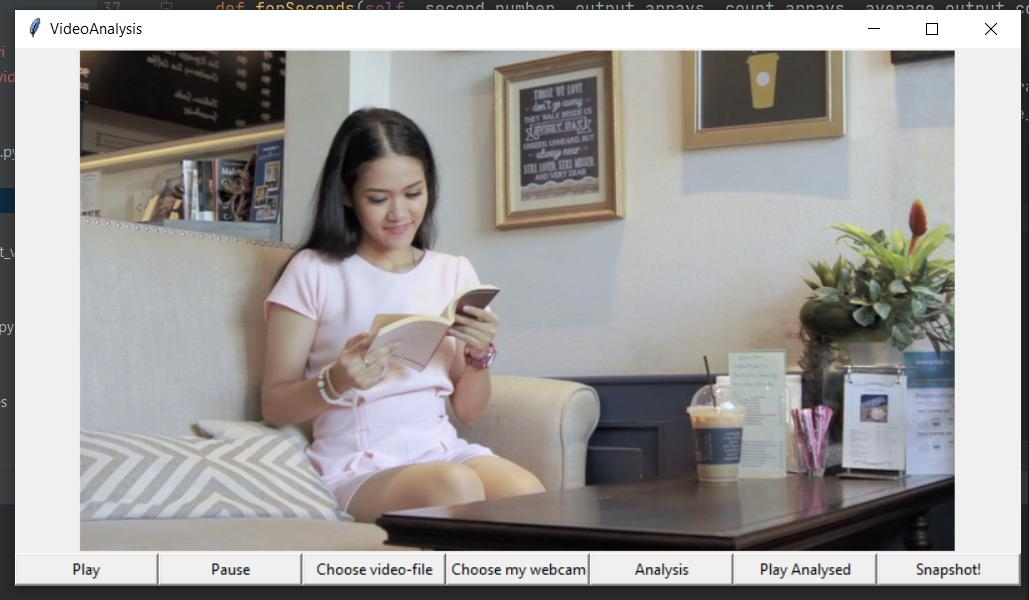
if self.thread1 is not None and self.thread1.is\_alive():

self.thread1.join

self.thread1 = None

#my\_app = MyApp(True)

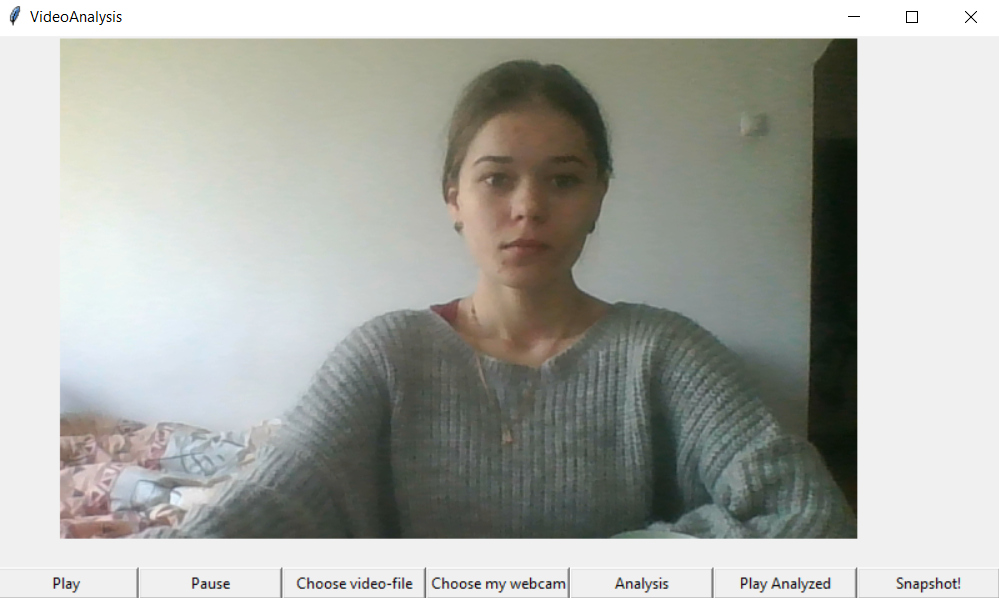
У результаті роботи було створено зручний програмний продукт, який представлено на рисунку 1.

**

*Рис. 1. Вигляд програмного продукту для користувача*

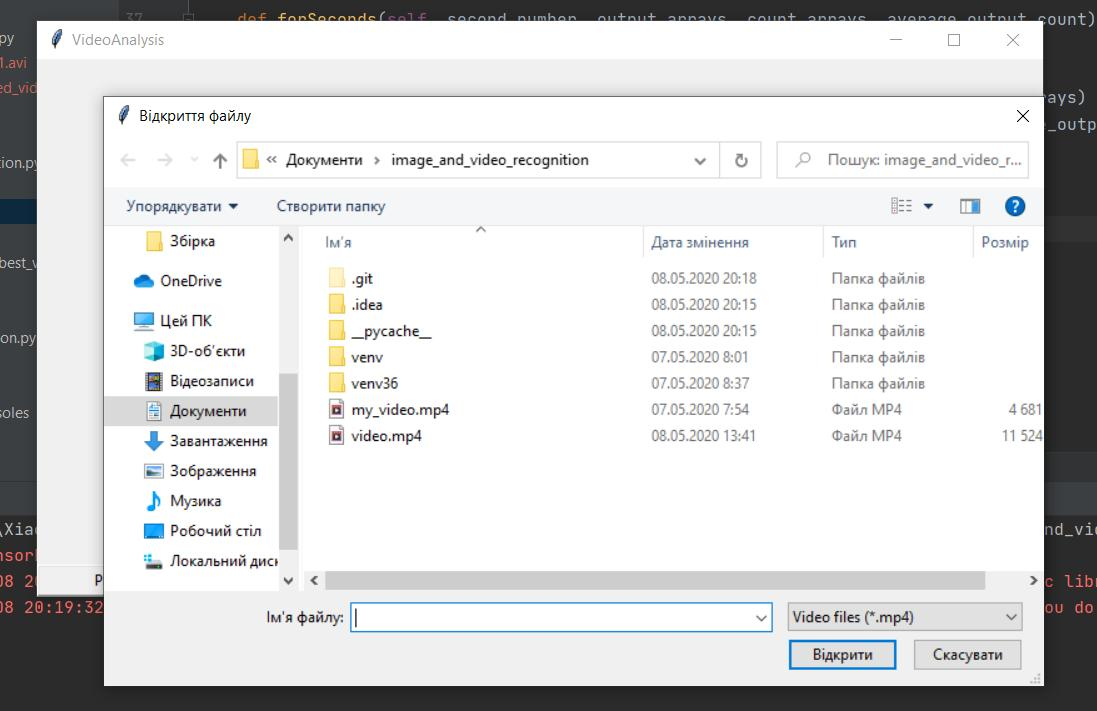
Зокрема, для користувача системи наявні наступні можливості:

1. Функція “Play” дозволяє відтворювати обраний відеопотік. Якщо користувач не вибрав жодний файл для аналізу, то автоматично будуть відображатися дані з відеореєстратора, що поставлені на пристрій, з якого запускається програмний продукт (рис. 2.)

**

*Рис. 2. Відображення відео-потоку з камери користувача*

1. Функція “Pause” дозволяє призупинити процес відеопотоку, що вибраний користувачем.
2. Функція “Choose video-file” дозволяє обрати користувачу файл, який він хоче у результаті проаналізувати (рис. 3.)

**

*Рис. 3. Вибір файлу для аналізу*

1. Функція “Choose my webcam” дозволяє користувачу переключити відеопотік з обраного файлу на відеокамеру, що знаходиться на використовуваному пристрої.

**Висновок**: у ході даної роботи було поставлено середовище розробки та ознайомлено з Python.