Другий етап виконання курсової роботи

**Визначення функціональних та нефункціональних вимог до програми, яка буде розроблятися. Вивчення засобів стандартної бібліотеки Python для роботи з даними різних типів. Опис даних з якими буде працювати програма.**

**2.1 Визначення функціональних та нефункціональних вимог до програми, яка буде розроблятися.**

Функціональні вимоги:

Програма повинна містити декілька функцій, котрі при правильній взаємодії мають давати потрібний результат користувачу. Деякими із них є наприклад: визначення лиця за допомогою каскадів на відео, визначення рук за допомогою каскадів на відео, визначення координат положення рук та лиця, подальше відслідковування лиця та рук на відео.

Нефункціональні вимоги:

Програма повинна працювати швидко, щоб надати користувачу максимальну зручність у використанні. Бажана частота кадрів це 60.

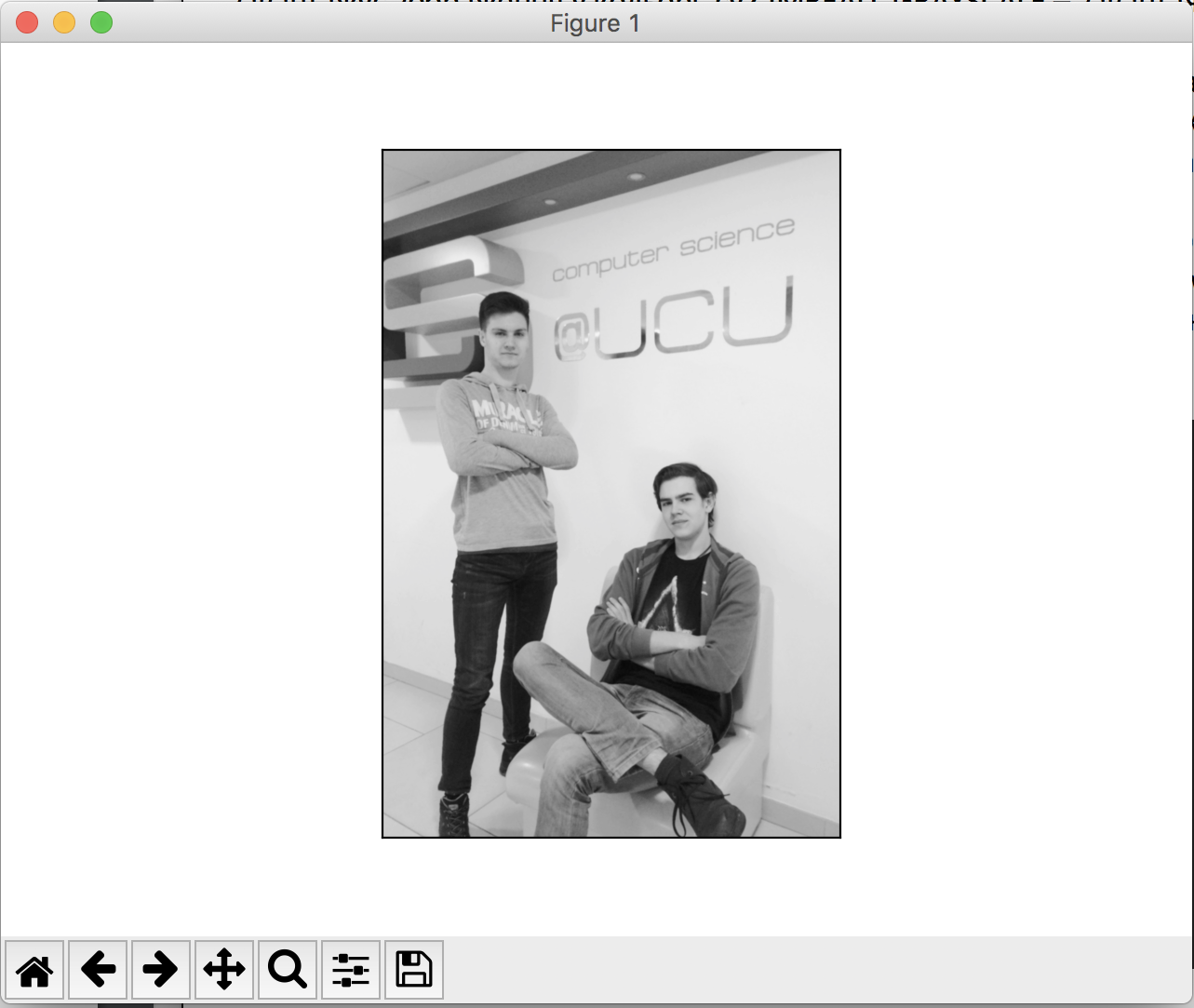
**2.3 Опис можливостей бібліотеки OpenCV.**

OpenCV (Open Source Computer Vision Library) – це відкрита бібліотека для комп’ютерного бачення та машинного навчання. Бібліотека OpenCV була створена для того, щоб забезпечити загальну інфраструктуру для програм комп’ютерного бачення та щоб прискорити використання машинного сприйняття в комерційних продуктах. OpenCV є продуктом ліцензії BSD, що дозволяє розробникам легко використовувати та змінювати код. Дана бібліотека надає дуже багато можливостей, і про деякі із них я розкажу.

**Читання та відображення зображень**

Використовуючи функцію cv2.imread() можна зчитувати зображення. Функція приймає два параметри: шлях до зображення, та параметр відкриття зображення(їх може може бути три: cv2.IMREAD\_COLOR – завантажує зображення у кольорі, cv2.IMREAD\_GRAYSCALE – завантажує чорно-біле зображення та cv2.IMREAD\_UNCHANGED – завантажує зображення, включаючи його альфа канали). Функція cv2.imshow() дозволяє виводити зображення на екран(у даному випадку розмір вікна буде відповідати розміру зображення). Функція приймає два параметри: назва вікна, котра передається у вигляді стрічки, сама картинка. Функція cv2.waitKey() – це функція для пррив’язки клавіш, як параметр вона приймає значення у мілісекундах і очікує на якусь подію з клавіатури. Якщо передано значення 0, тоді функція безкінечно очікує на подію. cv2.destroyAllWindows() закриває усі створені вікна. Для того щоб закрити тільки одне вікно, необхідно використати функцію cv2.destroyWindow(), де потрібно передати назву вікна, як параметр. Функція cv2.imwrite() дозволяє зберегти зображення(перший параметр – назва зображення, другий параметр - зображення).

import numpy as np  
import cv2  
  
img = cv2.imread('image.jpg', 0)  
cv2.imshow('image', img)  
k = cv2.waitKey(0)  
if k == 27: # wait for ESC key to exit  
 cv2.destroyAllWindows()  
elif k == ord('s'): # wait for 's' key to save and exit  
 cv2.imwrite('image\_gray.png', img)  
 cv2.destroyAllWindows()



**2.4 Опис даних які буде отримувати та з якими буде працювати. Формат, та відповідні до формату всі елементи.**

Із інтернету ми беремо датасети зображень, на основі котрих створюємо ккскажи для розпізнавання обличчя та рук. Сама програма працює із відео, котре знімається в реальному часі із вебкамери комп’ютера. Саме відео розбивається на зображення. Ці зображення аналізує програма і шукає каскади, схожі до натренованих.

**2.5 Приклад використання бібліотеки OpenCV.**

Тут наведено приклад використання бібліотеки OpenCV. Даний модуль дає змогу захоплювати відео із веб-камери конвертувати його у чорно-біле та виводити результат на екран у реальному часі.

import numpy as np  
import cv2  
  
cap = cv2.VideoCapture(0)  
  
while(True):  
 # Capture frame-by-frame  
 ret, frame = cap.read()  
  
 # Our operations on the frame come here  
 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
  
 # Display the resulting frame  
 cv2.imshow('frame',gray)  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
  
# When everything done, release the capture  
cap.release()  
cv2.destroyAllWindows()