

Міністерство науки і освіти України
Національний університет “Львівська політехніка”



Курсова робота
з дисципліни «Об’єктно-орієнтоване програмування C++ » на тему:
«Керування IP камерами спостереження за допомогою бібліотеки Open CV на мові
програмування C++»

Виконав:
ст. гр. ТР – 45
Пилип С.С.
Прийняв:
Кайдан М.В.

Львів 2021

Зміст

Вступ	3
Розділ 1. Open CV	4
Розділ 2. Встановлення Та Налаштування Бібліотеки.....	7
Розділ 3. Схема Обладнання Та Налаштування Камер	11
Розділ 4. Візуальна Складова	13
Розділ 5. Функціональна складова.....	14
Розділ 6. Виконання програми	21
Висновок.....	23
Список Використаної Літератури	24

Вступ

Метою курсової роботи є підключення IP камер спостереження та керування зображенням за допомогою бібліотеки C++ Open CV. Я використовуватиму Open CV версії 4.5.4. Камери використовую фірми Hikvision DS-2CD1021-I. Також використовую мобільний смартфон на базі OS Android та додаток IP Webcam, даний додаток дає змогу використовувати мобільний пристрій як IP камеру. В самій роботі я задав зміну відтворюваного зображення за допомогою клавіш на клавіатурі ПК.

Розділ 1. Open CV

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим кодом) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях.

Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Історія:

Офіційно проект OpenCV був запущений у 1999 році за ініціативою Intel Research з ціллю розвивати CPU-ресурсомісткі додатки. Основними вкладниками у проект була Intel's Performance Library Team та певна кількість експертів з чисельної оптимізації у Inter Russia. На перших етапах розвитку OpenCV основними задачами бібліотеки були:

- Розвивати дослідження у напрямку комп'ютерного зору, забезпечуючи добре оптимізований та відкритий код бібліотеки.
- Поширювати знання у сфері комп'ютерного зору, забезпечуючи загальну інфраструктуру, яку б могли розвивати розробники, таким чином код ставатиме більш легким для сприйняття та обміну.
- Розвивати засновані на роботі з комп'ютерним зором комерційні додатки, створюючи не залежну від платформи, оптимізовану та безкоштовну

бібліотеку. Для цього використовувалася ліцензія, яка не вимагала від таких комерційних додатків бути відкритими.

Перша альфа-версія OpenCV була оприлюднена на IEEE конференції з комп'ютерного зору й розпізнавання образів у 2000 році, і п'ять бета-версій було випущено у період між 2001 і 2005 роками. Перша версія 1.0 була випущена у 2006 році. У середині 2008 року, OpenCV отримала корпоративну підтримку від Willow Garage і знову перейшла у стадію активної розробки. «Пре-релізна» версія 1.1 була випущена у жовтні 2008 року.

Другий великий випуск OpenCV відбувся у жовтні 2009 року. OpenCV 2 включала у себе серйозні зміни у інтерфейсі C++. Ці зміни спрямовані на більш прості, тип-безпечні моделі, додавання нових функцій, і кращу реалізацію існуючих моделей в плані швидкодії (особливо на багатоядерних системах). Офіційні релізи надалі відбуваються кожні 6 місяців[3] і розробкою займається незалежна команда з Росії, яка підтримується комерційними корпораціями.

У серпні 2012 року, підтримку OpenCV було передано некомерційній організації, OpenCV.org.

Застосування:

Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- Аналіз та обробка зображень
- Системи з розпізнавання обличчя

- Ідентифікації об'єктів
- Розпізнавання жестів на відео
- Відстежування переміщення камери
- Побудова 3D моделей об'єктів
- Створення 3D хмар точок зі стерео камер
- Склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю
- Система взаємодії людини з комп'ютером
- Пошуку схожих зображень із бази даних
- Усування ефекту червоних очей при фотозйомці зі спалахом
- Стеження за рухом очей
- Аналіз руху
- Ідентифікація об'єктів
- Сегментація зображення
- Трекінг відео
- Розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності.

Мова програмування:

OpenCV написана на C++ і її основний інтерфейс також реалізовано на C++, але бібліотека і досі представляє старіший C інтерфейс. Наразі реалізовано інтерфейс мовами Python, Java і MATLAB / OCTAVE (починаючи з версії 2.5). API для цих інтерфейсів можна знайти в онлайн документації Оболонки для інших мов, таких як C#, C#, Ruby були розроблені з метою охоплення ширшої аудиторії.

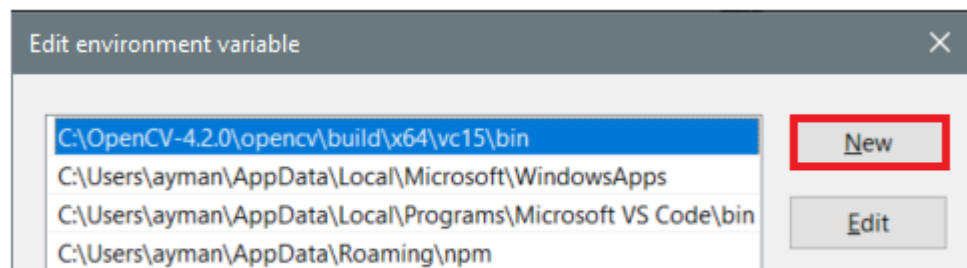
Всі нові розробки та алгоритми OpenCV наразі розробляються у C++ інтерфейсі.

Розділ 2. Встановлення Та Налаштування Бібліотеки

Для коректної роботи бібліотеки Open CV нам потрібно її встановити на наш пристрій. Попередньо на пристрої вже встановлена Visual Studio 2019. Бібліотеку скачуємо в офіційного сайту(<https://opencv.org/releases/>). Я вирішив використовувати останню версію даної бібліотеки – 4.5.4. На сайті завантажуюмо встановлювач для OS Windows.

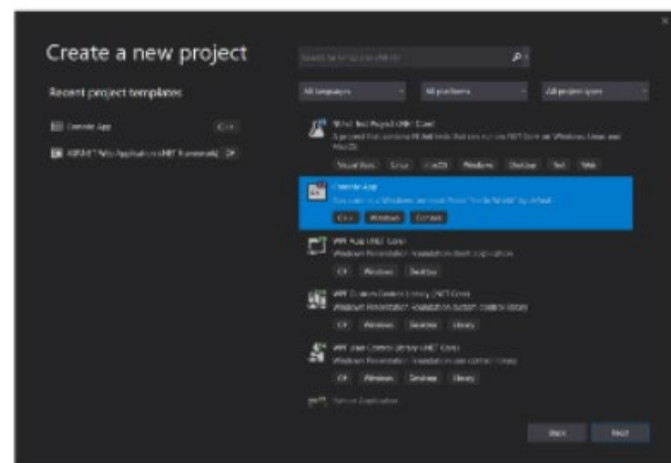
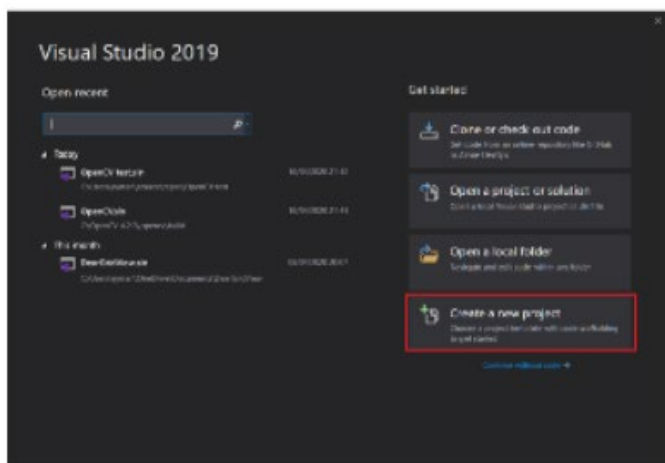
Далі потрібно пройти процес встановлення і раджу встановлювати саме в корінь основного диску операційної системи.

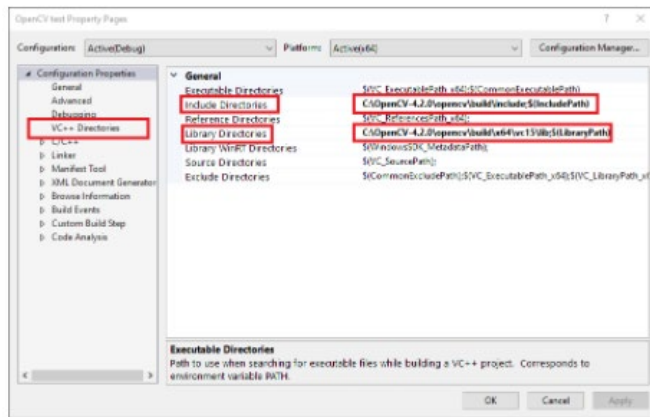
Наступним кроком є встановлення шляху для цієї бібліотеки.



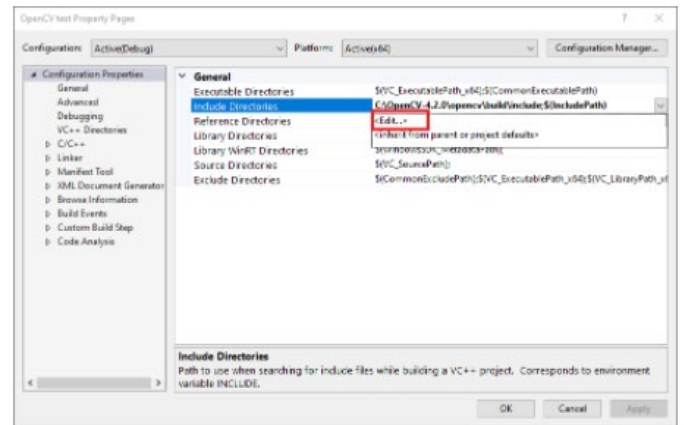
Editing the system path

Далі створюємо новий проект типу консолі. Та виконуємо наступні дії:

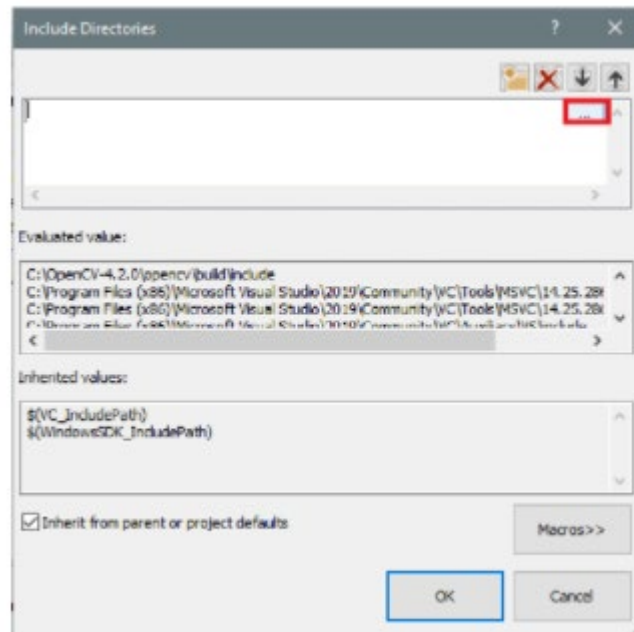




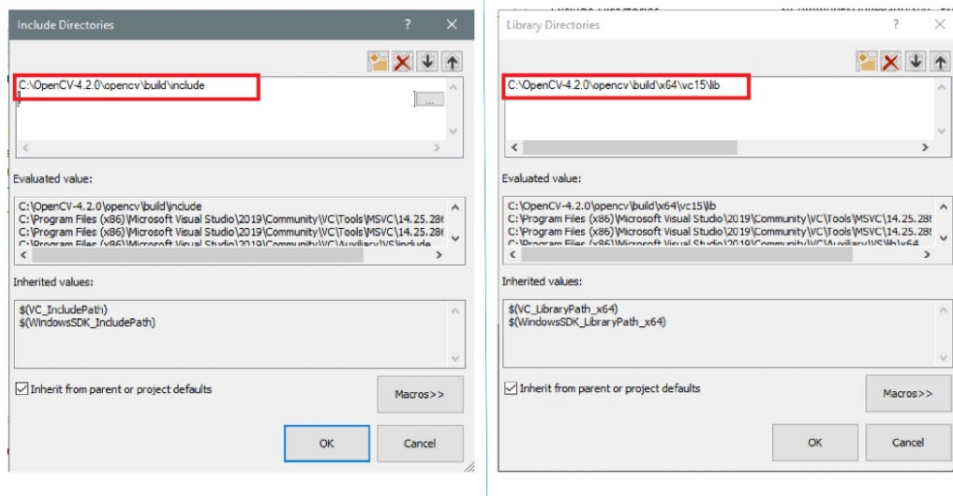
Edit both Include and Library Directories



Click on the right arrow next to blue entry

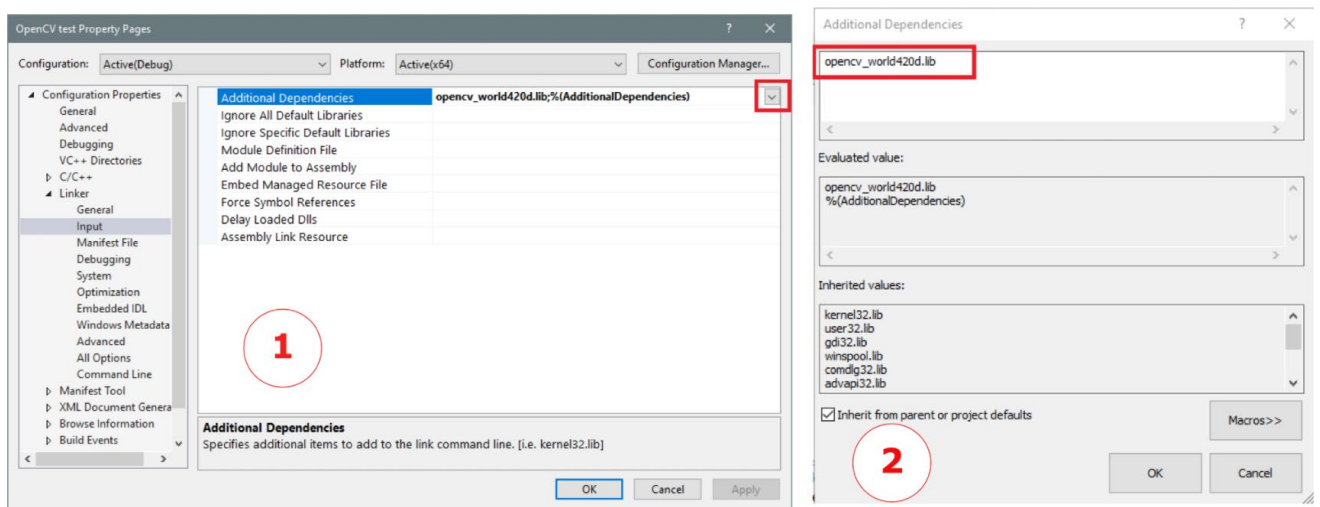


Для каталогу Include ви повинні додати такий шлях: C:\OpenCV-4.2.0\opencv\build\include. Зробіть те ж саме для каталогів бібліотеки, додавши цей внутрішній шлях: C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib.



Include and Lib directories both added to the project

Відредагуйте компоувальник проекту VC++ за допомогою `opencv_world420d.lib` динамічної бібліотеки OpenCV. Тут ви знайдете DLL (D ynamic L ink L ibrary):
`C:\OpenCV-4.2.0\opencv\build\x64\vc15\lib` скопіюйте ім'я файлу `opencv_world420d.lib` та вставте його в поле залежностей.



В моєму випадку ще потрібно було додати шлях в самому середовищі.

Конфигурация: Активная (Debug) Платформа: Активная (x64) Диспетчер конфигураций...

Свойства конфигурации

Общие

Дополнительно

Отладка

Каталоги VC++

С/С++

Компоновщик

Общие

Ввод

Файл манифеста

Отладка

Система

Оптимизация

Внедренный IDL

Метаданные Wind

Дополнительно

Все параметры

Командная строка

Инструмент манифес

Генератор XML-докум

Информация об исхо

События сборки

Общие

Каталоги исполняемых файлов

Включаемые каталоги

Внешние каталоги включения

Каталоги ссылок

Каталоги библиотек

Каталоги библиотек WinRT

Каталоги исходного кода

Исключаемые каталоги

Содержимое общедоступного проекта

Общедоступные каталоги включаемь

Все файлы заголовков являются общ

Каталоги общедоступных модулей С+

Все модули являются общедоступны

Каталоги исполняемых файлов	<code>\$(VC_ExecutablePath_x64);\$(CommonExecutablePath)</code>
Включаемые каталоги	<code>c:\opencv\build\include;\$(IncludePath)</code>
Внешние каталоги включения	<code>\$(VC_IncludePath);\$(WindowsSDK_IncludePath);</code>
Каталоги ссылок	<code>\$(VC_ReferencesPath_x64);</code>
Каталоги библиотек	<code>c:\opencv\build\x64\vc15\lib;\$(LibraryPath)</code>
Каталоги библиотек WinRT	<code>\$(WindowsSDK_MetadataPath);</code>
Каталоги исходного кода	<code>\$(VC_SourcePath);</code>
Исключаемые каталоги	<code>\$(CommonExcludePath);\$(VC_ExecutablePath_x64);\$(VC_Libr</code>

Содержимое общедоступного проекта

Общедоступные каталоги включаемь

Все файлы заголовков являются общ

Каталоги общедоступных модулей С+

Все модули являются общедоступны

Каталоги исполняемых файлов

Путь, который нужно использовать при поиске исполняемых файлов при сборке проекта VC++.

Соответствует переменной окружения PATH.

OK

Отмена

Применить

Розділ 3. Схема Обладнання Та Налаштування Камер

Для функціонування програми нам потрібно скласти схему підключення камер. Для цього нам потрібний WI-FI роутер, хаб з підтримкою додаткового живлення PoE через Ethernet кабель, камери(підключені по кабелю) та смартфон(підключений по WI-FI).

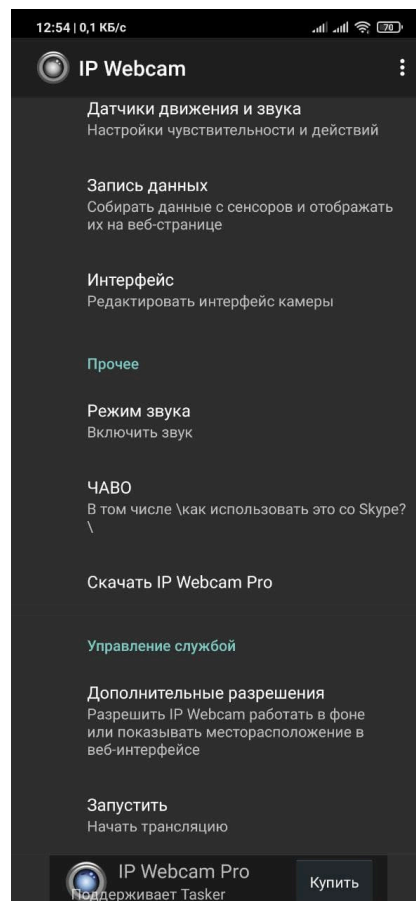


Також нам потрібні деякі додатки:

- SADP Tools Hikvision - ПЗ здійснює пошук активних мережевих пристроїв Hikvision у вашій підмережі та відображає інформацію про пристрої. Дозволяє також змінити базові установки мережі пристроїв, які підтримують це програмне забезпечення. Тут ми знайдемо IP адрес для під'єднання до наших камер.

SADP												
Total number of online devices: 2				Unbind		Export		Refresh		Filter		
ID	Device Type	Status	IPv4 Address	Port	Enhanced SDK Service Port	Software Version	IPv4 Gateway	HTTP Port	Device Serial No.	Subnet Mask	MAC	
001	DS-2CD1021-I	Active	192.168.1.101	8000	N/A	V5.5.83build 190...	192.168.1.1	80	DS-2CD1021-I20190801AAWRD444...	255.255.255.0	bc-b	
002	DS-2CD1021-I	Active	192.168.1.104	8000	N/A	V5.5.83build 190...	192.168.1.1	80	DS-2CD1021-I20190801AAWRD444...	255.255.255.0	bc-b	

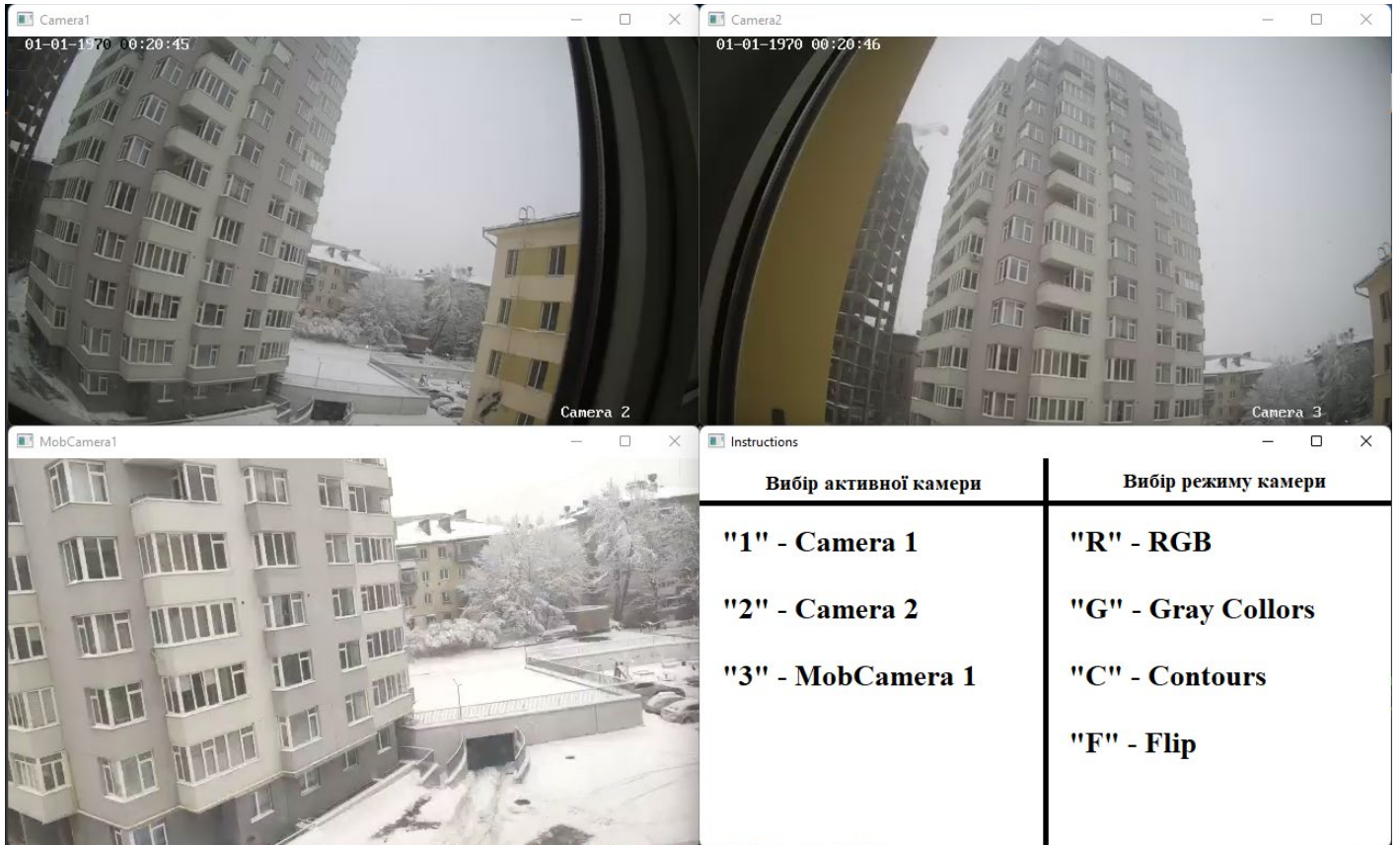
- IP Webcam – ПЗ для пристроїв на базі OS Android, яке дає змогу перетворити будь пристрій в IP камеру. Адреса для під'єднання вказується при натиску на кнопку Запустити.



Розділ 4. Візуальна Складова

Візуальна складова, з якою буде взаємодіяти користувач, це 4 програмних вікна. В 3 вікна будуть виводитись зображення з двох камер та камери смартфона.

При запуску програми ми бачимо звичайне зображення з камер:



Розділ 5. Функціональна складова

Реалізація програми написана в одному файлі мовою C++:

```
/*Підключення необхідних бібліотек*/
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui_c.h>
#include <iostream>

using namespace cv;
using namespace std;

int KomKey[4] = { 1, 1, 1 }, CameraN = 1; //Змінні для роботи з
камерами(вибір та змінна режиму камери)
int c;

VideoCapture cap, cap1, mob; //змінна в які захоплюються зображення з
камер

Mat cam1, cam2, mobcam1; //Області які відображаються в вікнах

Mat frame(Mat Camkey1, int Komkey1) // Спосіб відображення
{
    Mat CamExport;
```

```

switch (Komkey1)
{
case 1://звичайне зображення
    return Camkey1;
    break;
case 2:
    cvtColor(Camkey1, CamExport, 6);// чорнобіле зображення
    return CamExport;
    break;
case 3:
    Canny(Camkey1, CamExport, 500, 1000, 5, true);//зображення
    контурів
    return CamExport;
    break;
case 4:
    flip(Camkey1, CamExport, 1);// відхзеркалене зображення
    return CamExport;
default:
    break;
}

}

int main(int argc, char* argv[])
{

```



```
cap.open("rtsp://admin:qwerty123456@192.168.1.101:554/Streaming/Channels/102");// підключення до першої камери
```

```
cap1.open("rtsp://admin:qwerty123456@192.168.1.104:554/Streaming/Channels/102");// підключення до другої камери
```

```
mob.open("rtsp://192.168.1.100:8080/h264_pcm.sdp");//  
підключення до мобільної камери
```

```
/*Перевірка на підключення*/
```

```
if (!cap.isOpened()) // if not success, exit program
```

```
{
```

```
    cout << "Cannot open the video cam-1" << endl;
```

```
    return -1;
```

```
}
```

```
if (!cap1.isOpened()) // if not success, exit program
```

```
{
```

```
    cout << "Cannot open the video cam-2" << endl;
```

```
    return -1;
```

```
}
```

```
if (!mob.isOpened()) // if not success, exit program
```

```
{
```

```
    cout << "Cannot open the video mob" << endl;
```

```
    return -1;
```

```
}
```

```
/*Створення Вікон з певними розмірами*/
```

```
namedWindow("Camera1", CV_WINDOW_AUTOSIZE);
```

```
moveWindow("Camera1", 0, 0);//create a window called "Camera1"
```

```
namedWindow("Camera2", CV_WINDOW_AUTOSIZE);
```



```

moveWindow("Camera2", 640, 0); //create a window called "Camera2"
namedWindow("MobCamera1", CV_WINDOW_AUTOSIZE);
moveWindow("MobCamera1", 0, 390); //create a window called
"MobCamera1"
namedWindow("Instructions", CV_WINDOW_AUTOSIZE);
moveWindow("Instructions", 640, 390);
Mat img = imread("C:/Users/Serge/Desktop/4 курс 1 семестр/ООП
C++/cursova/ConsoleApplication1/Instructions.png",
IMREAD_COLOR); //зчитування зображення налаштувань
imshow("Instructions", img);
while (1)
{

    int b = waitKey(5);
    switch (b) //Вибір камери по натиску на клавішу
    {
        case 49: //1
            CameraN = 1;
            break;
        case 50: //2
            CameraN = 2;
            break;
        case 51: //3
            CameraN = 3;
            break;
        default:
            break;
    }
    switch (b) //вибір режиму роботи для вибраної камери

```

```

{
case 114://R
    KomKey[CameraN-1] = 1;
    break;
case 103://G
    KomKey[CameraN-1] = 2;
    break;
case 99://C
    KomKey[CameraN-1] = 3;
    break;
case 102://F
    KomKey[CameraN - 1] = 4;
    break;
default:
    break;
}

```

```

cap.read(cam1); //позиціонування камери в даній області

```

```

cap1.read(cam2);

```

```

mob.read(mobcam1);

```

```

/*Перевірка на стан підключення*/

```

```

bool bSuccess = cap.read(cam1); // read a new frame from
video
if (!bSuccess) //if not success, break loop
{

```

```

        cout << "Cannot read a frame from video stream" << endl;
        break;
    }
    bSuccess = cap1.read(cam2); // read a new frame from video
    if (!bSuccess) //if not success, break loop
    {
        cout << "Cannot read a frame from video stream" << endl;
        break;
    }
    bSuccess = mob.read(mobcam1); // read a new frame from video
    if (!bSuccess) //if not success, break loop
    {
        cout << "Cannot read a frame from video stream" << endl;
        break;
    }

    cam1 = frame(cam1, KomKey[0]);
    cam2 = frame(cam2, KomKey[1]);
    mobcam1 = frame(mobcam1, KomKey[2]);

    imshow("Camera1", cam1); //show the frame in "MyVideo"
window
    imshow("Camera2", cam2);
    imshow("MobCamera1", mobcam1);

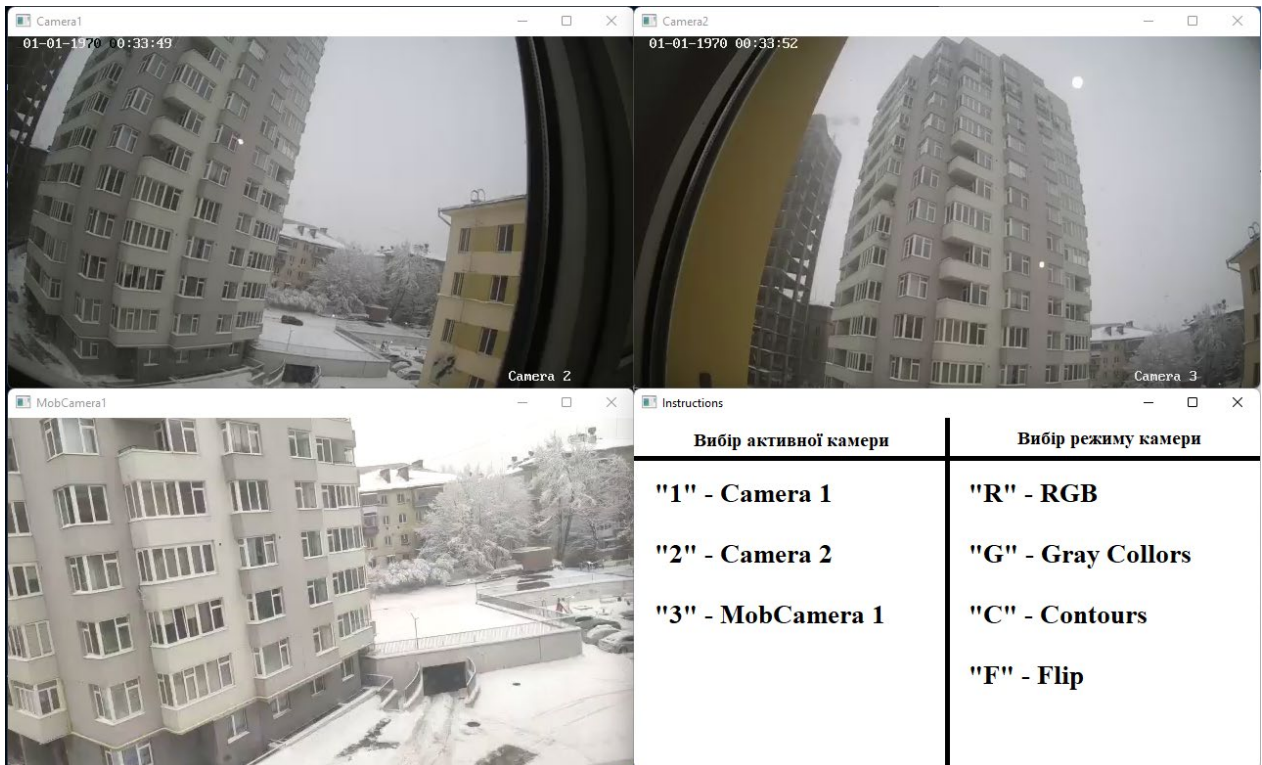
    /*Вихід з програми*/
    if (waitKey(30) == 27) //wait for 'esc' key press for 30ms.
    If 'esc' key is pressed, break loop
    {

```

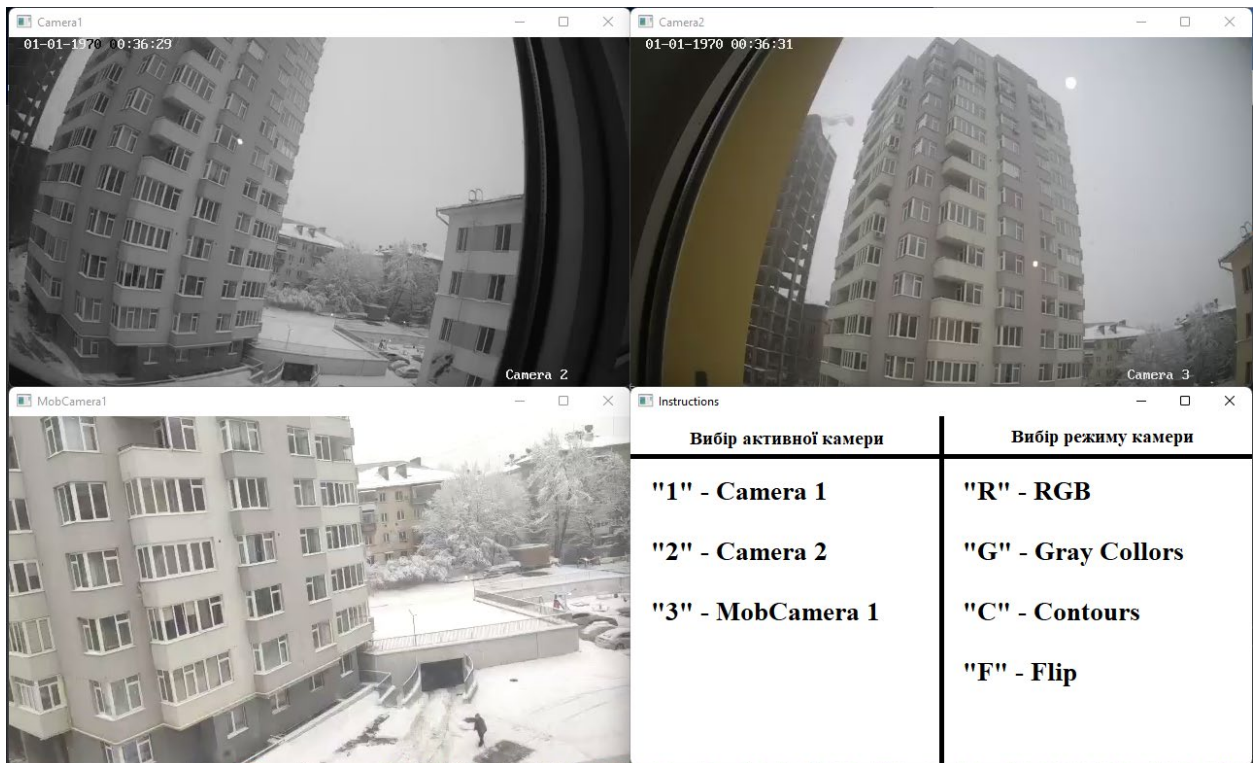
```
        cout << "esc key is pressed by user" << endl;  
        break;  
    }  
}  
return 0;  
}
```

Розділ 6. Виконання програми

Звичайний вигляд:



Змінимо для першої камери режим відображення на чорно-біле:



Далі на 2 камері встановимо відображення Контурів:



А на зображенні з смартфона розвернемо зображення:



Висновок

В даній курсовій роботі було виконано під'єднання камер за допомогою бібліотеки Open CV. Також було додано функцію зміни зображення з камер на чорно-біле, контурне, обернене та звичайне. Це реалізовано за допомогою вибору камери та перевірки яку клавішу було натиснено.

Алгоритми OpenCV застосовують для аналізу та обробки зображень, систем з розпізнавання обличчя, ідентифікації об'єктів, розпізнавання жестів на відео, відстежування переміщення камери, побудова 3D моделей об'єктів, створення 3D хмар точок зі стерео камер, склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю, система взаємодії людини з комп'ютером, пошуку схожих зображень із бази даних, усування ефекту червоних очей при фотозйомці зі спалахом, стеження за рухом очей, аналіз руху, ідентифікація об'єктів, сегментація зображення, трекінг відео, розпізнавання елементів сцени і додавання маркерів для створення доповненої реальності.

Список Використаної Літератури

- 1) <https://nadzor.ua/product/sadp-tools-dla-windows-i-macos>
- 2) <https://coderoad.ru/42214012/c-opencv-%D0%BE%D1%82%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5-%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B2-%D0%BE%D0%BA%D0%BD%D0%B5>
- 3) <https://stackoverflow.com/questions/35675676/access-mjpeg-streamer-in-opencv-3-0>
- 4) https://coderoad.ru/59600937/cv2-cvtColor-img-cv2-COLOR_BGR2RGB-%D0%BD%D0%B5-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82
- 5) <https://ravesli.com/urok-99-peredacha-argumentov-po-adresu/>
- 6) <https://ravesli.com/urok-65-operator-switch/>
- 7) <https://robocraft.ru/blog/computervision/365.html>
- 8) <https://answers.opencv.org/question/3664/ip-network-camera-access/>
- 9) <https://blog.actorsfit.com/a?ID=00350-1a1cde2d-e2a2-450a-8f77-bcf99147b411>
- 10) <https://overcoder.net/q/44017/%D0%BA%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D1%82%D1%8C-rgb-%D0%B2-%D1%87%D0%B5%D1%80%D0%BD%D0%BE-%D0%B1%D0%B5%D0%BB%D0%BE%D0%B5-%D0%B2-opencv>
- 11) <https://answers.opencv.org/question/182681/how-to-set-fps-width-frame-and-height-frame-of-a-ipcamera-in-c/>

- 12) <https://www.codegrepper.com/code-examples/python/opencv+set+frame+size>
- 13) <https://coderoad.ru/14287/%D0%A3%D0%B2%D0%B5%D0%BB%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-%D1%80%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B7%D0%B0%D1%85%D0%B2%D0%B0%D1%82%D0%B0-%D0%BA%D0%B0%D0%BC%D0%B5%D1%80%D1%8B-%D0%B2-OpenCV>
- 14) <https://question-it.com/questions/1186520/opencv-izmenit-razreshenie-videocapture-vo-vremja-zahvata>
- 15) <http://rusproject.narod.ru/winapi/r/rect.html>
- 16) <https://stackoverflow.com/questions/14028193/size-of-matrix-opencv>
- 17) https://www.google.com/search?q=canvas+c%2B%2Bbuilder&rlz=1C1CHZN_ruUA964UA964&oq=canvas+c%2B%2B&aqs=chrome.1.69i57j0i22i30j0i512j0i22i30l7.14076j0j7&sourceid=chrome&ie=UTF-8
- 18) [https://www.google.com/search?q=Vec3b\(0%2C+255%2C+0\)&rlz=1C1CHZN_ruUA964UA964&oq=Vec3b\(0%2C+255%2C+0\)&aqs=chrome..69i57j33i160.454j0j7&sourceid=chrome&ie=UTF-8](https://www.google.com/search?q=Vec3b(0%2C+255%2C+0)&rlz=1C1CHZN_ruUA964UA964&oq=Vec3b(0%2C+255%2C+0)&aqs=chrome..69i57j33i160.454j0j7&sourceid=chrome&ie=UTF-8)
- 19) <https://www.py4u.net/discuss/84016>
- 20) <https://ip-nablyudenie.ru/kak-uznat-rtsp-potok-s-ip-kamery>
- 21) <https://hikvision.org.ua/ru/chto-takoe-rtsp-ssylka-i-kak-sformirovat-ee-dlya-ustroystv-hikvision/>
- 22) <https://opencv.org/releases/>
- 23) <https://stackoverflow.com/questions/18682201/opencv-where-can-i-find-cv-window-autosize-constants>