Programming language design and program analysis

Last project

Railway sleepers shift inspection using Image processing.

Name: Abdulbary Ahmed Othman Naji.

Student ID: SL20215017.

University of science and technology of china (USTC).

I. Abstract:

In this Report I will talk about the problem of Railway sleepers shift and how we can solve this problem using Image Processing. Railway contains series of sleepers that connected together and these sleepers with time they may shift a little from their original position this problem will cause with time dangerous effects for the railway which leads to distortion in railway.

II. Problem definition

The problem in Railway the sleepers creep with time so this creeping process we can't notice because it's slow and the creep extension is too small in mm this slight shift may extend with time and cause a harm distortion in the Railway so we must avoid and monitor this slight difference in some way.

III. Problem Solution using Image Processing

Install devices every 500 meter and each device is embedded with Camera Module, 4G module, ARM processor and power management Module for low power consumption. The device is placed 10 meters far away from the Railway in tightly fixed location to ensure no slight motion in the device otherwise the results may not be accurate. The device should be calibrated and aligned directly toward the target region of interest. Device abstract block diagram shown in fig 1.

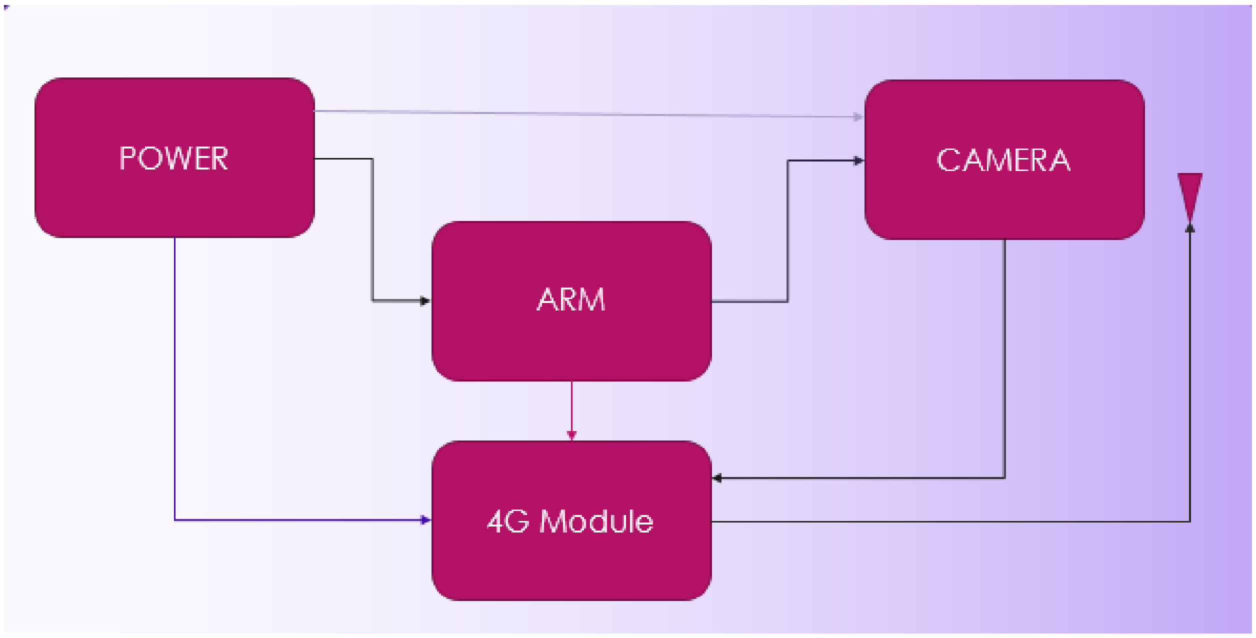
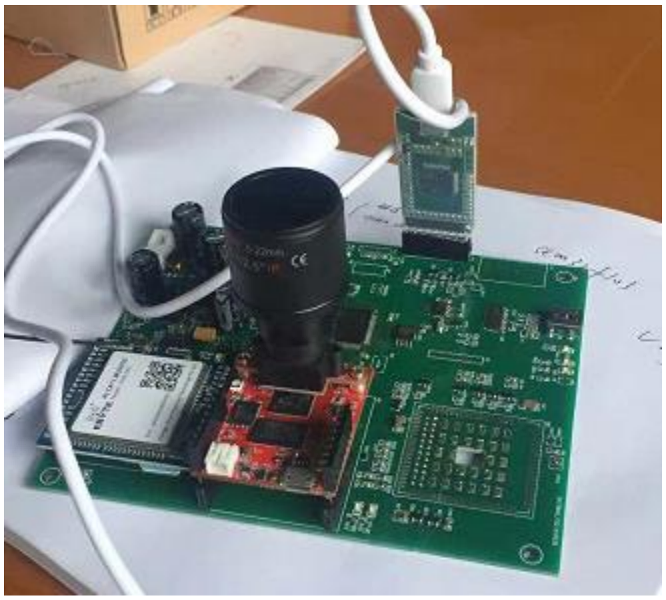


Fig 1. Device block diagram.



- **Device components description:**

  The device main function is to send a captured image every day at a specific time to the server for further processing after the device send the image it enters the sleep mode and configure the time and RTC (Real-time clock) alarm to wake up next day. Time synchronization is provided using the server to synchronize all the devices installed along the Railway.

1. Camera Module

   The Camera captures high resolution images with WQXGA2 high resolution 2592x1944. The Captured images then sent to the Server using 4G module. The Camera used is shown in fig 2.

2. 4G module

   Send the captured images to the server, and responsible for direct communication between the Server and Device.

3. ARM Processor

   For power management and RTC alarm to ensure low power consumption because the device use battery for power supply to the whole components of the System. The device should have an alarm to wake from deep sleep mode.

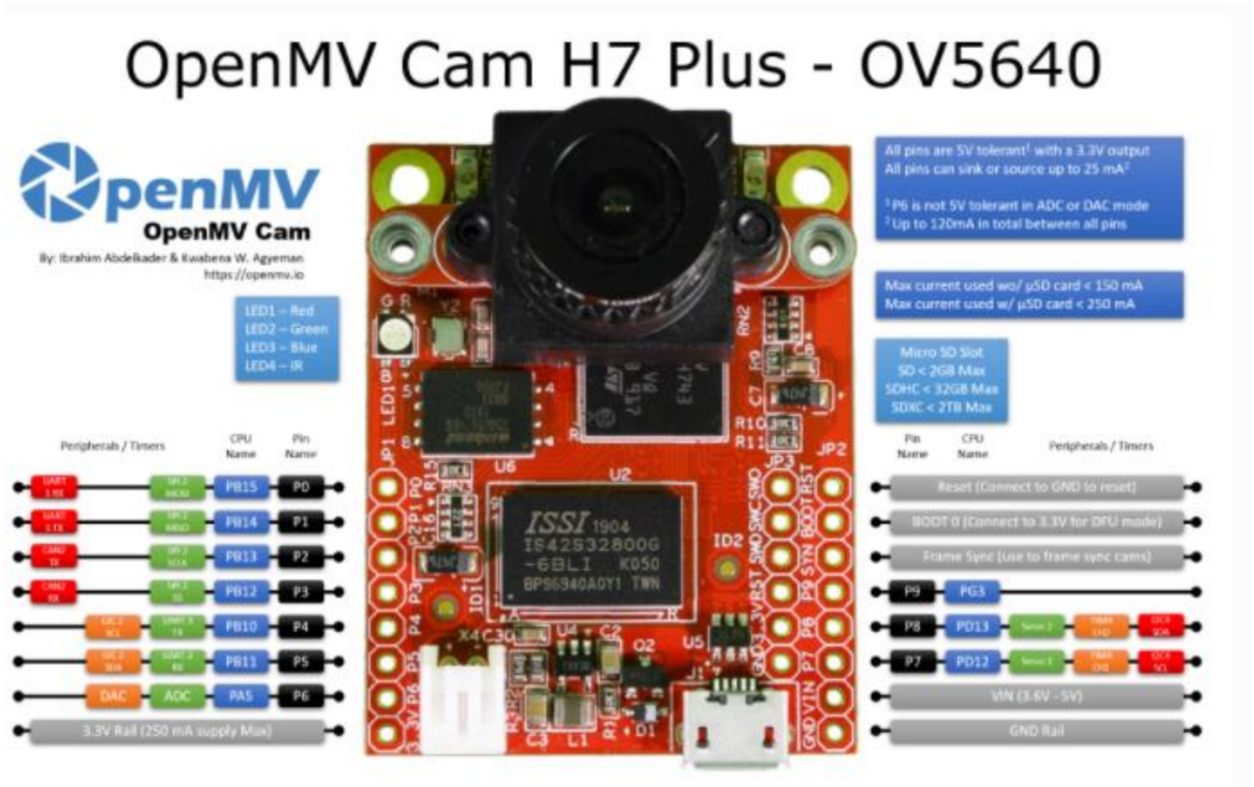
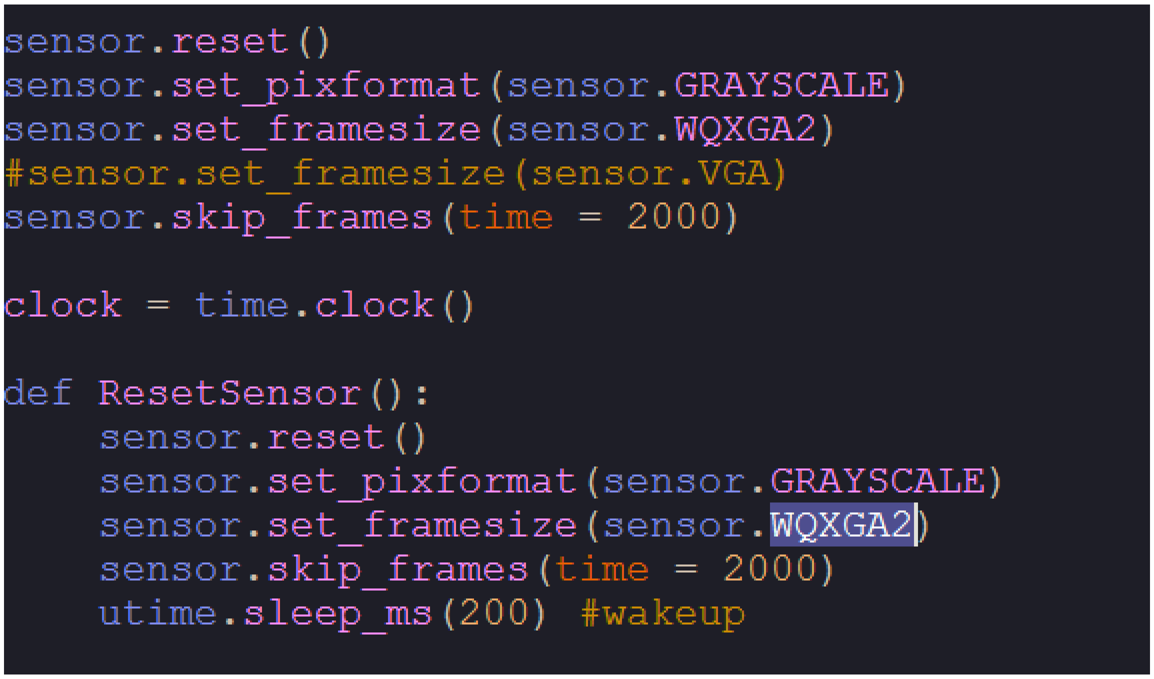

OpenMV Cam H7 Plus - OV5640
OpenMV
OpenMV Cam
By: Ibrahim Abdelkader & Kwabena W. Agyeman
https://openmv.io
LED1 – Red
LED2 – Green
LED3 – Blue
LED4 – IR
All pins are 5V tolerant with a 3.3V output
All pins can sink or source up to 25 mA
P6 is not 5V tolerant in ADC or DAC mode
Up to 120mA in total between all pins
Max current used wo/ µSD card < 150 mA
Max current used w/ µSD card < 250 mA
Micro SD Slot
SD < 2GB Max
SDHC < 32GB Max
SDXC < 2TB Max
Peripherals / Timers
CPU Name
Pin Name
PB15
PB14
PB13
PB12
PB10
PB11
PA5
P0
P1
P2
P3
P4
P5
P6
DAC
ADC
3.3V Rail (250 mA supply Max)
Reset (Connect to GND to reset)
BOOT 0 (Connect to 3.3V for DFU mode)
Frame Sync (use to frame sync cams)
P9
P8
P7
PG3
PD13
PD12
VIN (3.6V - 5V)
GND Rail

Fig.2 Camera Device for capturing images.



```
sensor.reset()
sensor.set_pixformat(sensor.GRAYSCALE)
sensor.set_framesize(sensor.WQXGA2)
#sensor.set_framesize(sensor.VGA)
sensor.skip_frames(time = 2000)

clock = time.clock()

def ResetSensor():
    sensor.reset()
    sensor.set_pixformat(sensor.GRAYSCALE)
    sensor.set_framesize(sensor.WQXGA2)
    sensor.skip_frames(time = 2000)
    utime.sleep_ms(200)  #wakeup
```

- Server Side processing
  The Server main function is receiving all images sent by terminal nodes installed along the railway and process these images, store

the results in database based on the processing results it reports distortion.

- Image processing Part.

  Here the received images are processed using openCV. The reference images and template cross image and received image all of them are the main inputs for the Image Processing algorithm,

  The algorithm is based on Cross Correlation Method to match template in both reference image and test image the find shift between both Images reference image and received image. Based on the following equation.

1. **method=TM_SQDIFF**

$$R(x,y)=\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2$$

2. **method=TM_SQDIFF_NORMED**

$$R(x,y)=\frac{\sum_{x',y'}(T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$$

3. **method=TM_CCORR**

$$R(x,y)=\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))$$

4. **method=TM_CCORR_NORMED**

$$R(x,y)=\frac{\sum_{x',y'}(T(x',y')\cdot I(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2\cdot\sum_{x',y'}I(x+x',y+y')^2}}$$

5. **method=TM_CCOEFF**

$$R(x,y)=\sum_{x',y'}(T'(x',y')\cdot I'(x+x',y+y'))$$

where

$$T'(x',y') = T(x',y') - 1/(w \cdot h) \cdot \sum_{x'',y''} T(x'',y'')$$
$$I'(x+x',y+y') = I(x+x',y+y') - 1/(w \cdot h) \cdot \sum_{x'',y''} I(x+x'',y+y'')$$

6. **method=TM_CCOEFF_NORMED**

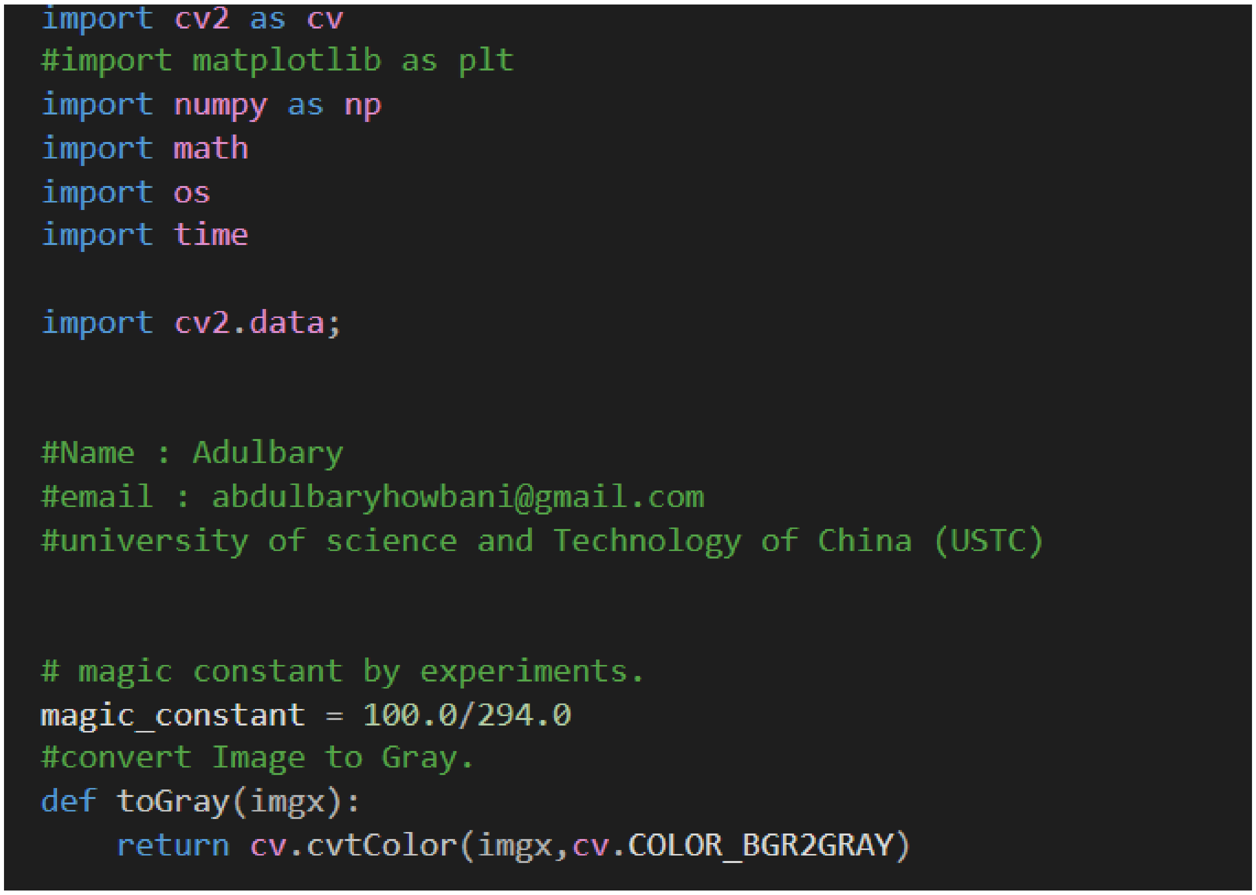
$$R(x,y) = \frac{\sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}}$$

$M1 = (x0, y0)\ for\ reference\ image.$

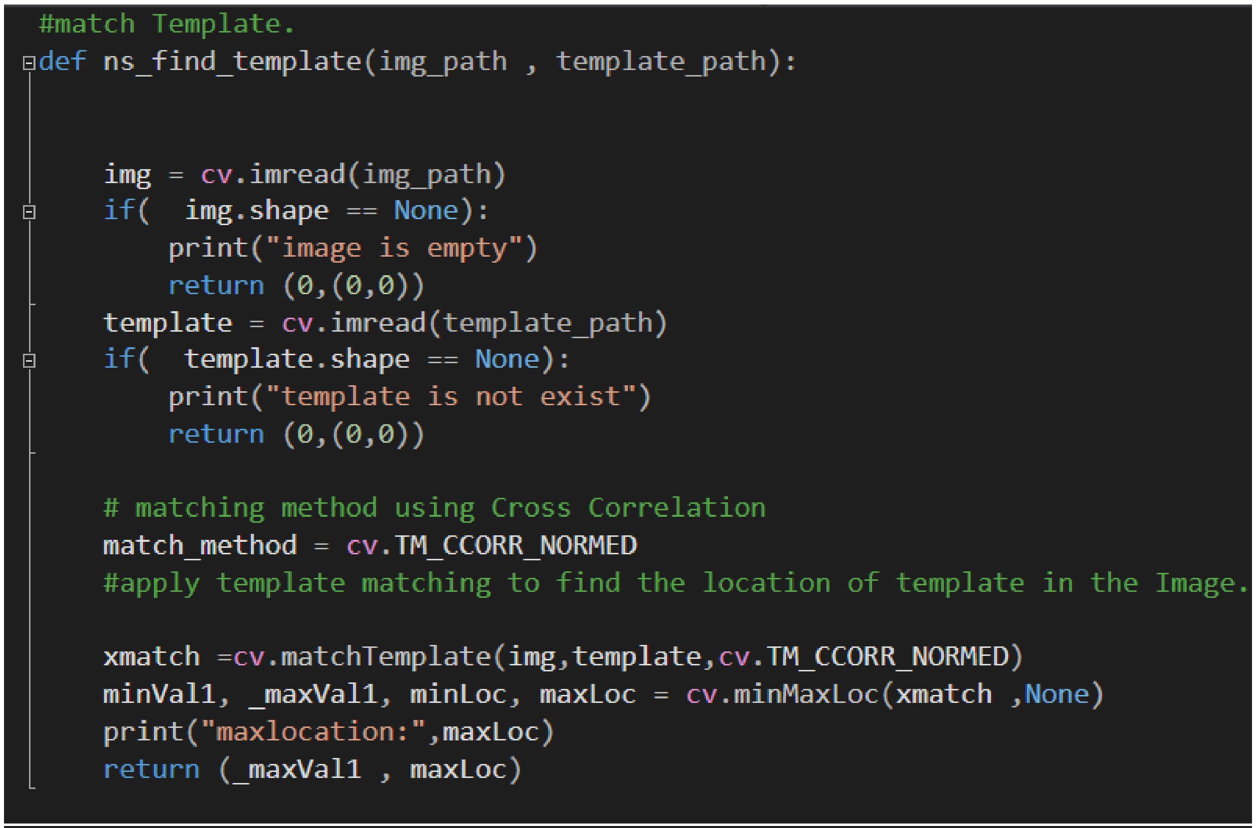
$M2 = (x1, y1)\quad for\ test\ image$

M1, M2 are the match result for both reference image and received image respectively. The match process is based on template matching in other words to find a template inside an image. In each railway sleeper we can make a cross sign and save it as template and capture a reference image when the Railway is constructed or any time and capture an image for the first time as reference image.

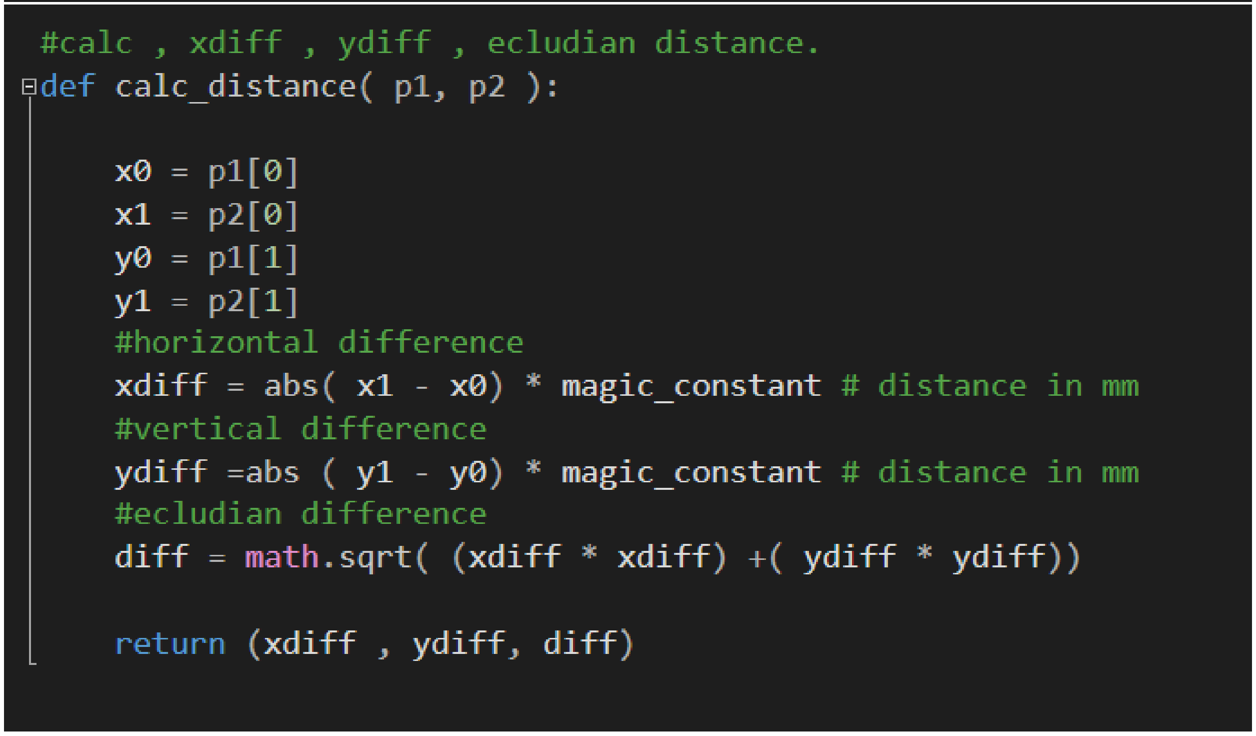
- Python Code snapshots.



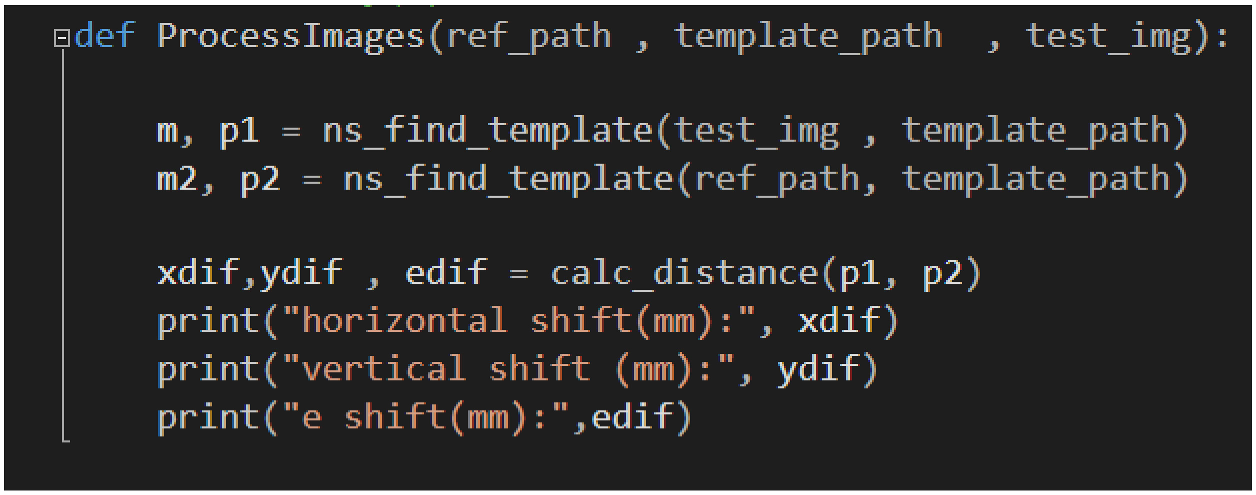
```
import cv2 as cv
#import matplotlib as plt
import numpy as np
import math
import os
import time

import cv2.data;


#Name : Adulbary
#email : abdulbaryhowbani@gmail.com
#university of science and Technology of China (USTC)


# magic constant by experiments.
magic_constant = 100.0/294.0
#convert Image to Gray.
def toGray(imgx):
    return cv.cvtColor(imgx,cv.COLOR_BGR2GRAY)
```



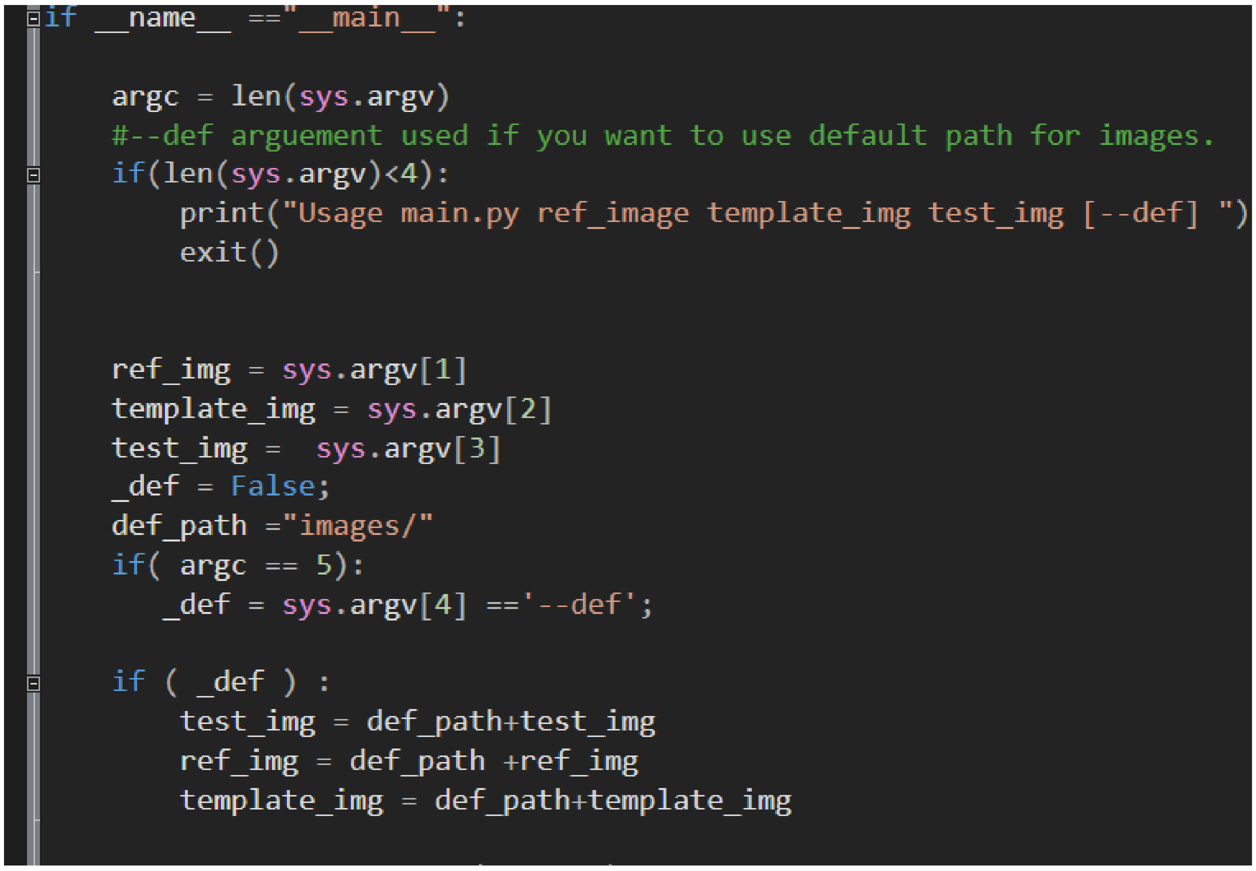
```
#match Template.
def ns_find_template(img_path , template_path):


    img = cv.imread(img_path)
    if(  img.shape == None):
        print("image is empty")
        return (0,(0,0))
    template = cv.imread(template_path)
    if(  template.shape == None):
        print("template is not exist")
        return (0,(0,0))

    # matching method using Cross Correlation
    match_method = cv.TM_CCORR_NORMED
    #apply template matching to find the location of template in the Image.

    xmatch =cv.matchTemplate(img,template,cv.TM_CCORR_NORMED)
    minVal1, _maxVal1, minLoc, maxLoc = cv.minMaxLoc(xmatch ,None)
    print("maxlocation:",maxLoc)
    return (_maxVal1 , maxLoc)
```



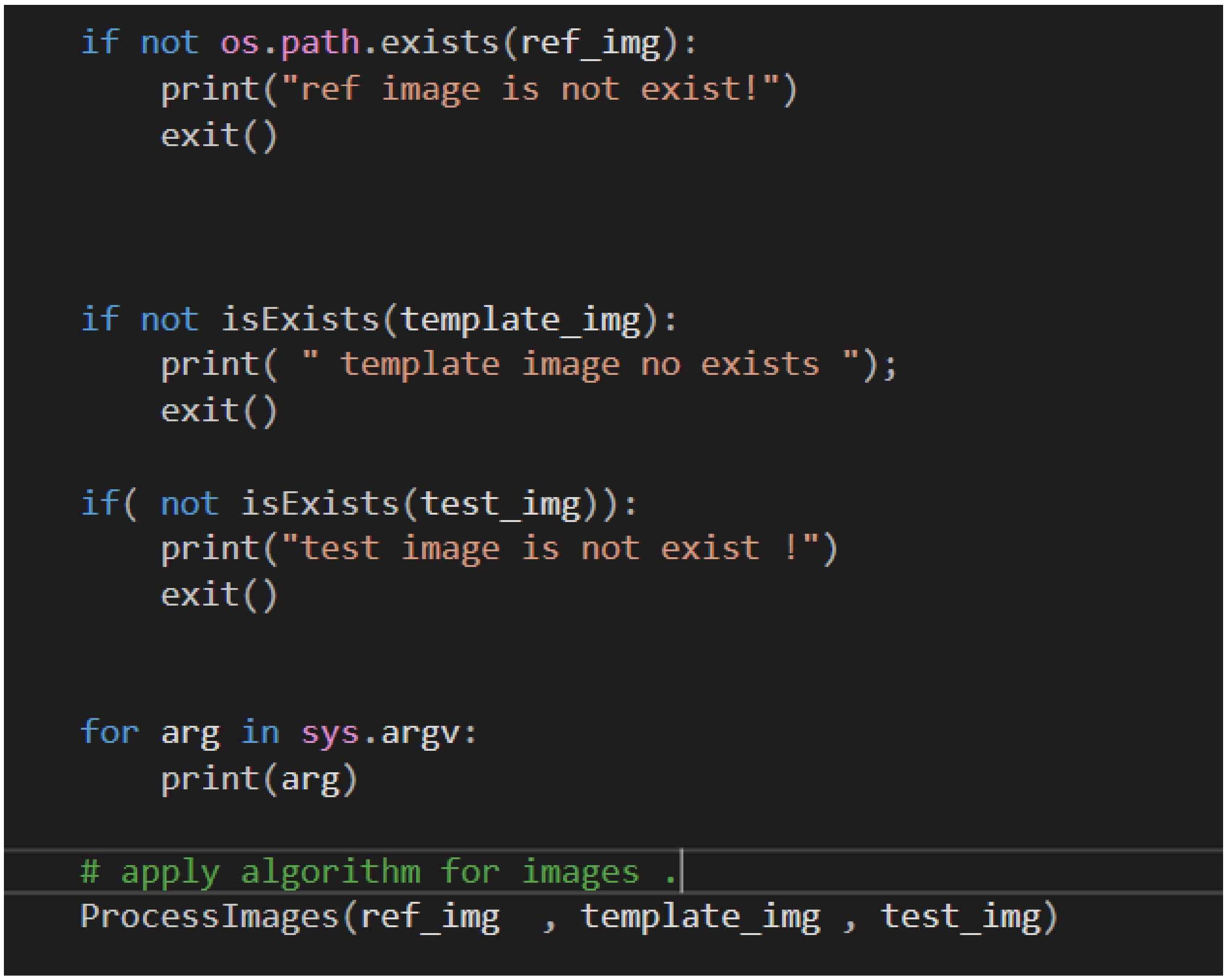
```
#calc , xdiff , ydiff , ecludian distance.
def calc_distance( p1, p2 ):

    x0 = p1[0]
    x1 = p2[0]
    y0 = p1[1]
    y1 = p2[1]
    #horizontal difference
    xdiff = abs( x1 - x0) * magic_constant # distance in mm
    #vertical difference
    ydiff =abs ( y1 - y0) * magic_constant # distance in mm
    #ecludian difference
    diff = math.sqrt( (xdiff * xdiff) +( ydiff * ydiff))

    return (xdiff , ydiff, diff)
```



```
def ProcessImages(ref_path , template_path  , test_img):

    m, p1 = ns_find_template(test_img , template_path)
    m2, p2 = ns_find_template(ref_path, template_path)

    xdif,ydif , edif = calc_distance(p1, p2)
    print("horizontal shift(mm):", xdif)
    print("vertical shift (mm):", ydif)
    print("e shift(mm):",edif)
```



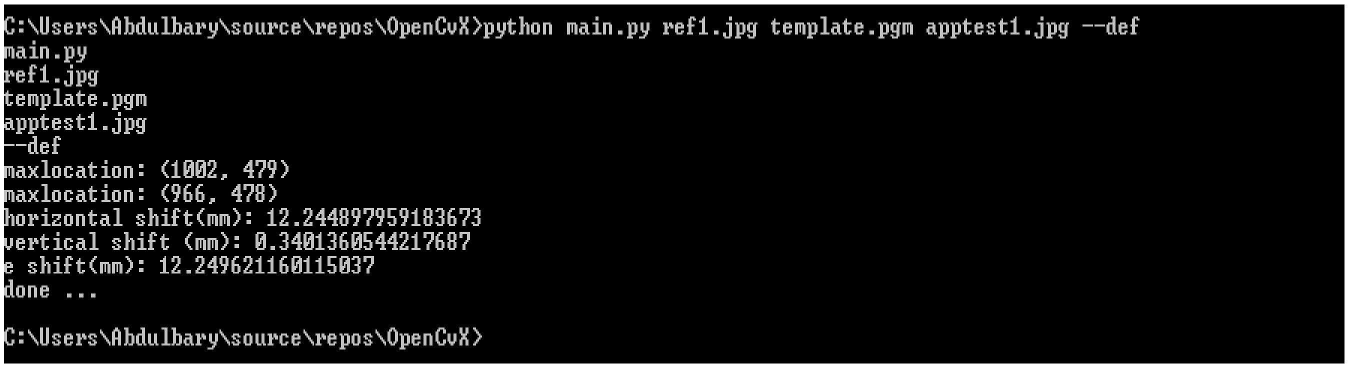
```
if __name__ =="__main__":

    argc = len(sys.argv)
    #--def arguement used if you want to use default path for images.
    if(len(sys.argv)<4):
        print("Usage main.py ref_image template_img test_img [--def] ")
        exit()


    ref_img = sys.argv[1]
    template_img = sys.argv[2]
    test_img =  sys.argv[3]
    _def = False;
    def_path ="images/"
    if( argc == 5):
       _def = sys.argv[4] =='--def';

    if ( _def ) :
        test_img = def_path+test_img
        ref_img = def_path +ref_img
        template_img = def_path+template_img
```



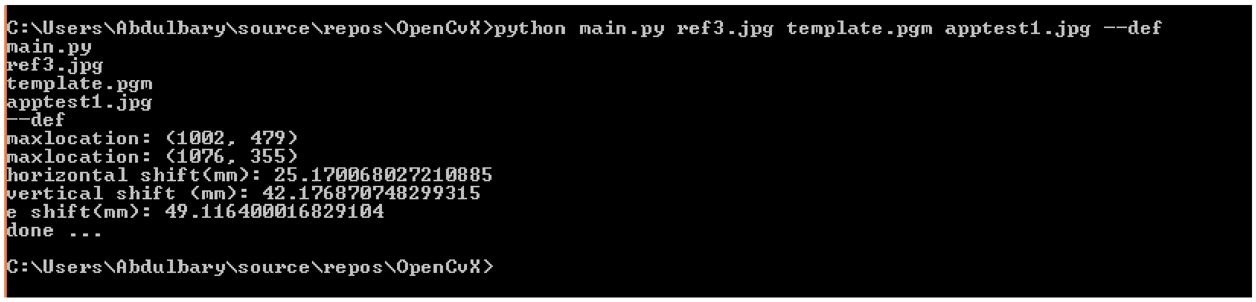
```python
    if not os.path.exists(ref_img):
        print("ref image is not exist!")
        exit()



    if not isExists(template_img):
        print( " template image no exists ");
        exit()

    if( not isExists(test_img)):
        print("test image is not exist !")
        exit()


    for arg in sys.argv:
        print(arg)

    # apply algorithm for images .
    ProcessImages(ref_img  , template_img , test_img)
```

How to run python code.

From command prompt.



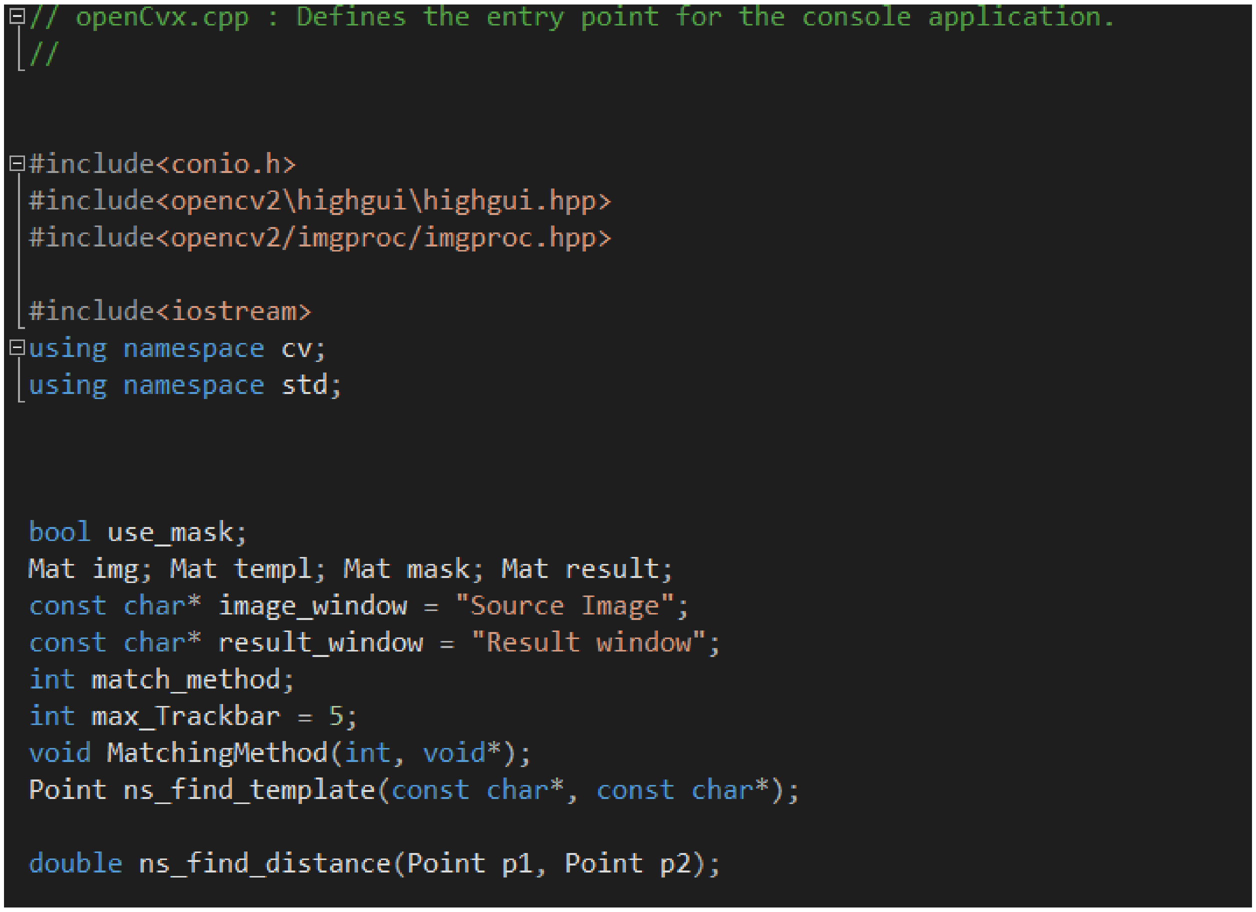
```
C:\Users\Abdulbary\source\repos\OpenCvX>python main.py ref1.jpg template.pgm apptest1.jpg --def
main.py
ref1.jpg
template.pgm
apptest1.jpg
--def
maxlocation: (1002, 479)
maxlocation: (966, 478)
horizontal shift(mm): 12.244897959183673
vertical shift (mm): 0.3401360544217687
e shift(mm): 12.249621160115037
done ...

C:\Users\Abdulbary\source\repos\OpenCvX>
```



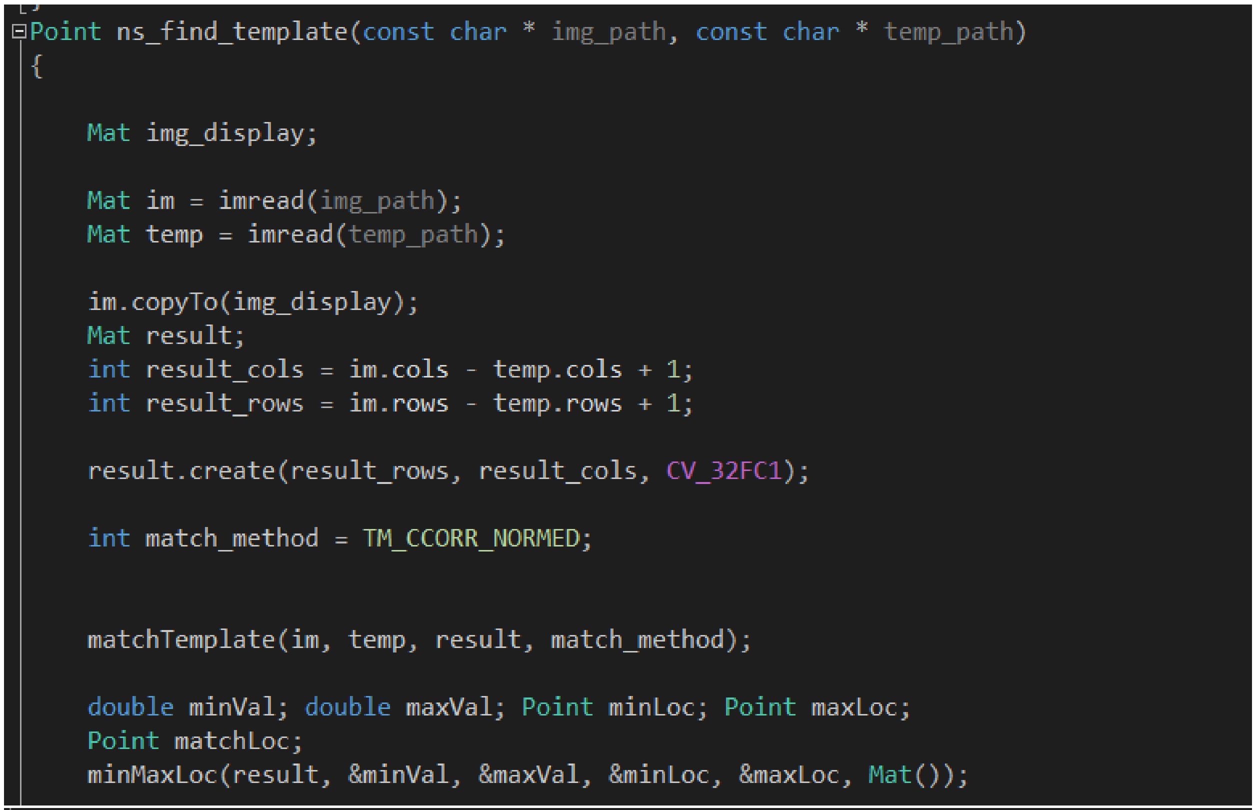
```
C:\Users\Abdulbary\source\repos\OpenCvX>python main.py ref3.jpg template.pgm apptest1.jpg --def
main.py
ref3.jpg
template.pgm
apptest1.jpg
--def
maxlocation: (1002, 479)
maxlocation: (1076, 355)
horizontal shift(mm): 25.170068027210885
vertical shift (mm): 42.176870748299315
e shift(mm): 49.116400016829104
done ...

C:\Users\Abdulbary\source\repos\OpenCvX>
```

- C++ OpenCv Code.

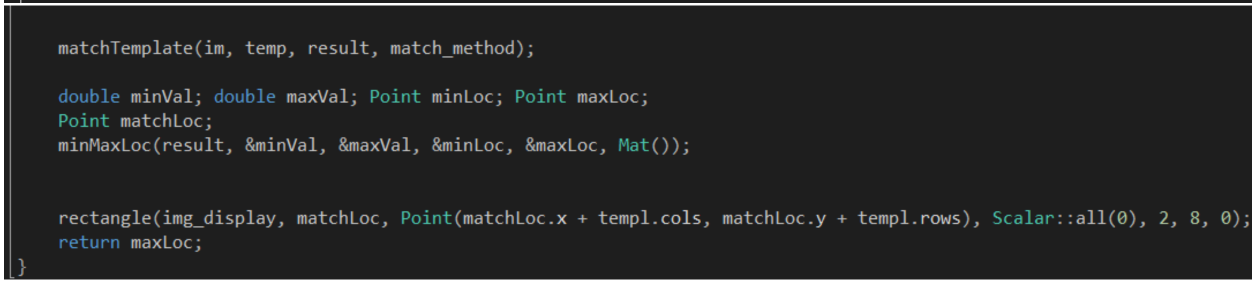
  To run C++ code you should have OpenCv Version 2.4 or higher.



```cpp
// openCvx.cpp : Defines the entry point for the console application.
//


#include<conio.h>
#include<opencv2\highgui\highgui.hpp>
#include<opencv2/imgproc/imgproc.hpp>

#include<iostream>
using namespace cv;
using namespace std;



bool use_mask;
Mat img; Mat templ; Mat mask; Mat result;
const char* image_window = "Source Image";
const char* result_window = "Result window";
int match_method;
int max_Trackbar = 5;
void MatchingMethod(int, void*);
Point ns_find_template(const char*, const char*);

double ns_find_distance(Point p1, Point p2);
```

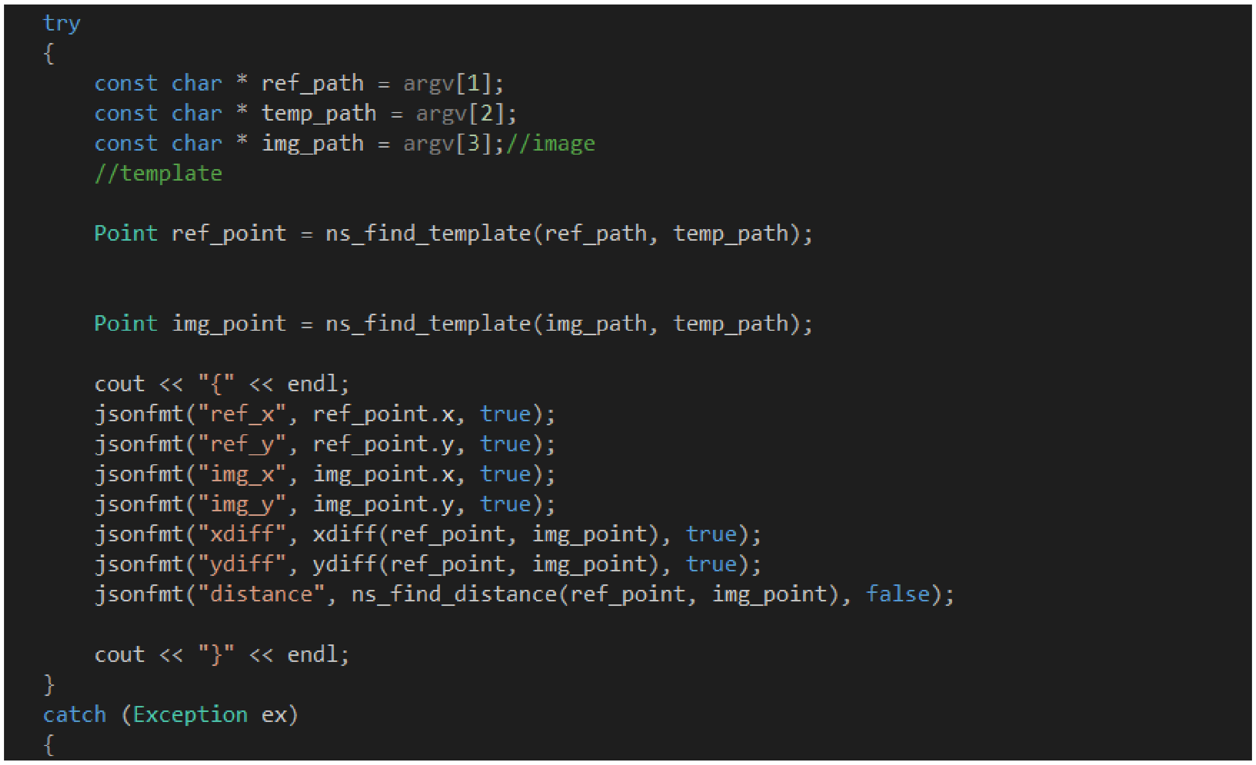


```
Point ns_find_template(const char * img_path, const char * temp_path)
{

    Mat img_display;

    Mat im = imread(img_path);
    Mat temp = imread(temp_path);

    im.copyTo(img_display);
    Mat result;
    int result_cols = im.cols - temp.cols + 1;
    int result_rows = im.rows - temp.rows + 1;

    result.create(result_rows, result_cols, CV_32FC1);

    int match_method = TM_CCORR_NORMED;


    matchTemplate(im, temp, result, match_method);

    double minVal; double maxVal; Point minLoc; Point maxLoc;
    Point matchLoc;
    minMaxLoc(result, &minVal, &maxVal, &minLoc, &maxLoc, Mat());
```



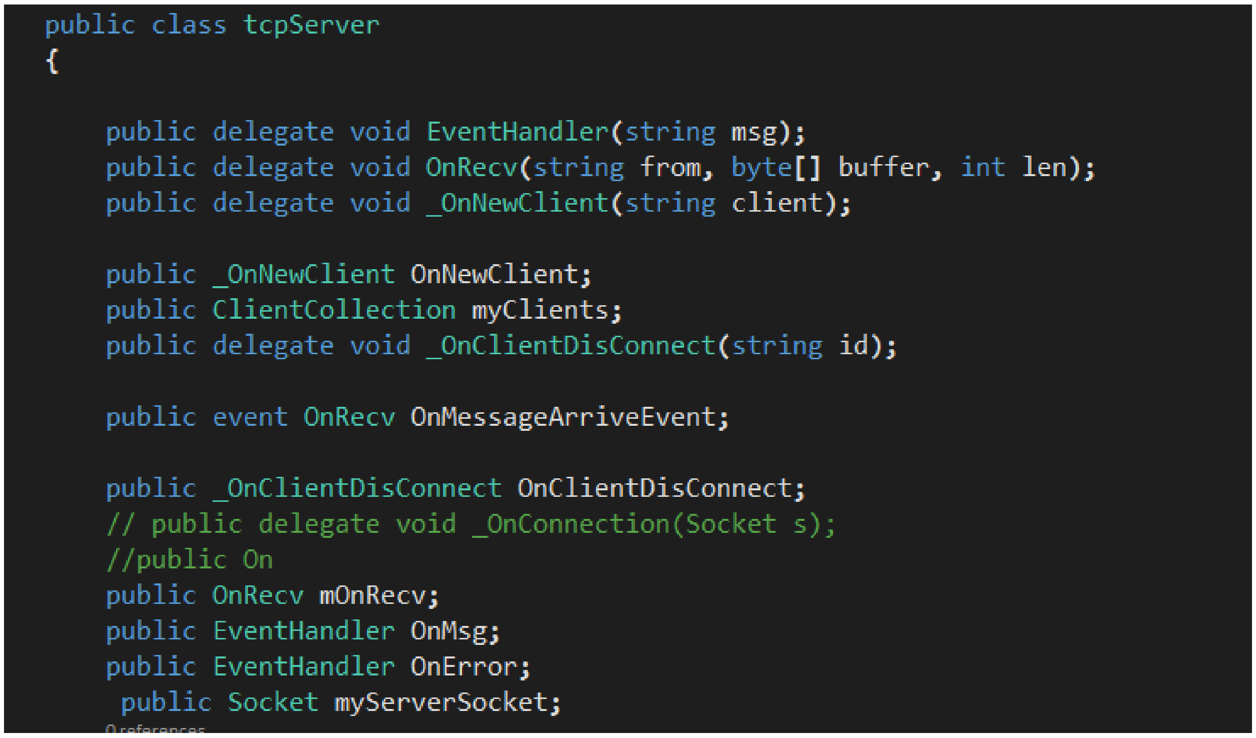
```
    matchTemplate(im, temp, result, match_method);

    double minVal; double maxVal; Point minLoc; Point maxLoc;
    Point matchLoc;
    minMaxLoc(result, &minVal, &maxVal, &minLoc, &maxLoc, Mat());
```

```

    rectangle(img_display, matchLoc, Point(matchLoc.x + templ.cols, matchLoc.y + templ.rows), Scalar::all(0), 2, 8, 0);
    return maxLoc;
}
```

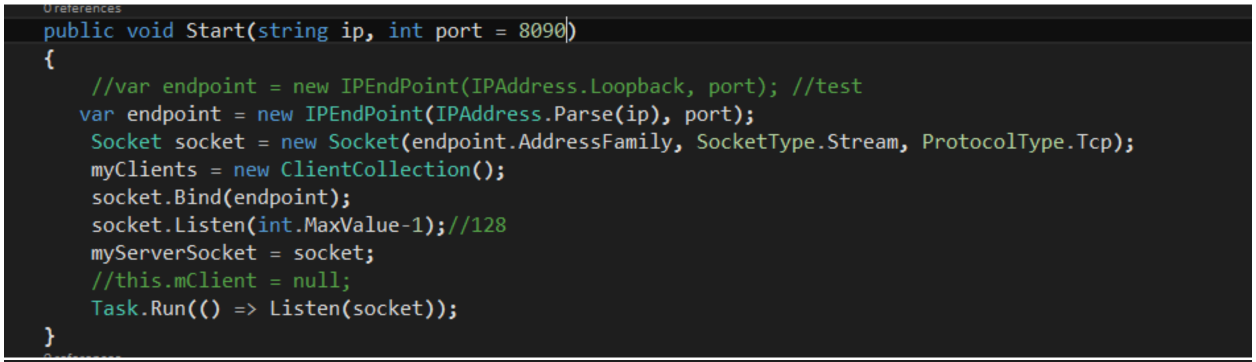


```
try
{
    const char * ref_path = argv[1];
    const char * temp_path = argv[2];
    const char * img_path = argv[3];//image
    //template

    Point ref_point = ns_find_template(ref_path, temp_path);


    Point img_point = ns_find_template(img_path, temp_path);

    cout << "{" << endl;
    jsonfmt("ref_x", ref_point.x, true);
    jsonfmt("ref_y", ref_point.y, true);
    jsonfmt("img_x", img_point.x, true);
    jsonfmt("img_y", img_point.y, true);
    jsonfmt("xdiff", xdiff(ref_point, img_point), true);
    jsonfmt("ydiff", ydiff(ref_point, img_point), true);
    jsonfmt("distance", ns_find_distance(ref_point, img_point), false);

    cout << "}" << endl;
}
catch (Exception ex)
{
```

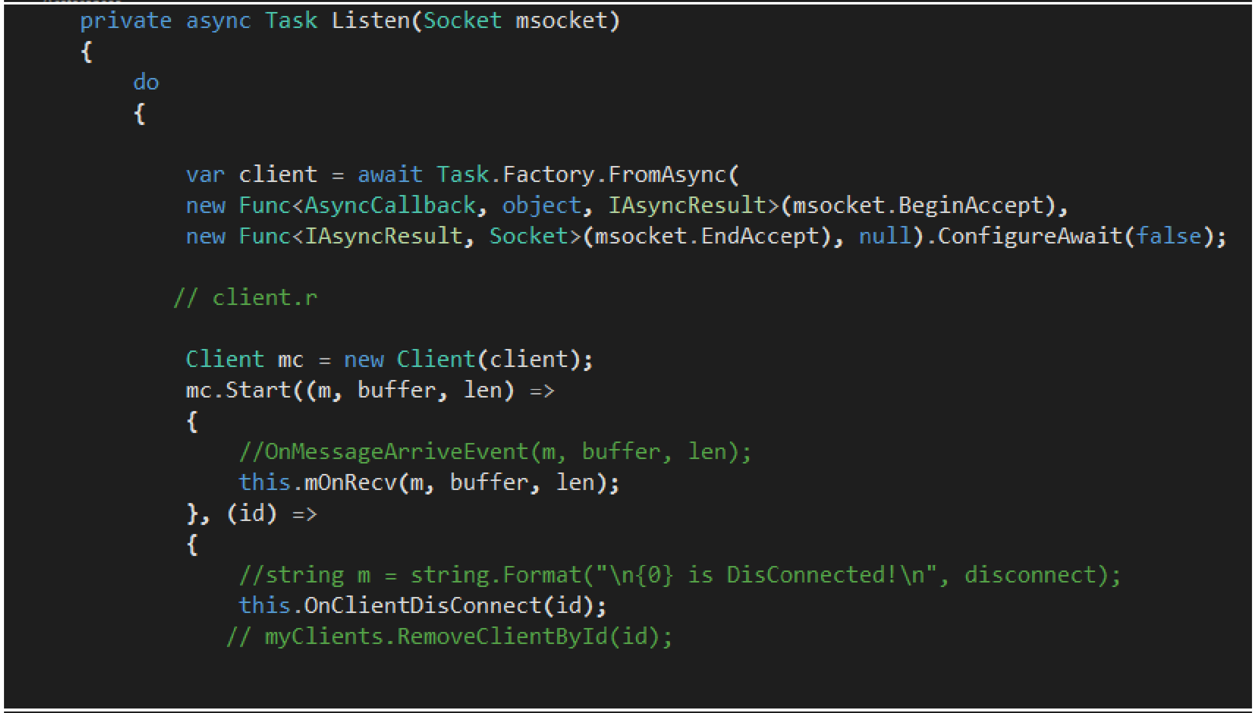
- Server Application code.
  The Server application is built using C# programming language. which is TCP Server.



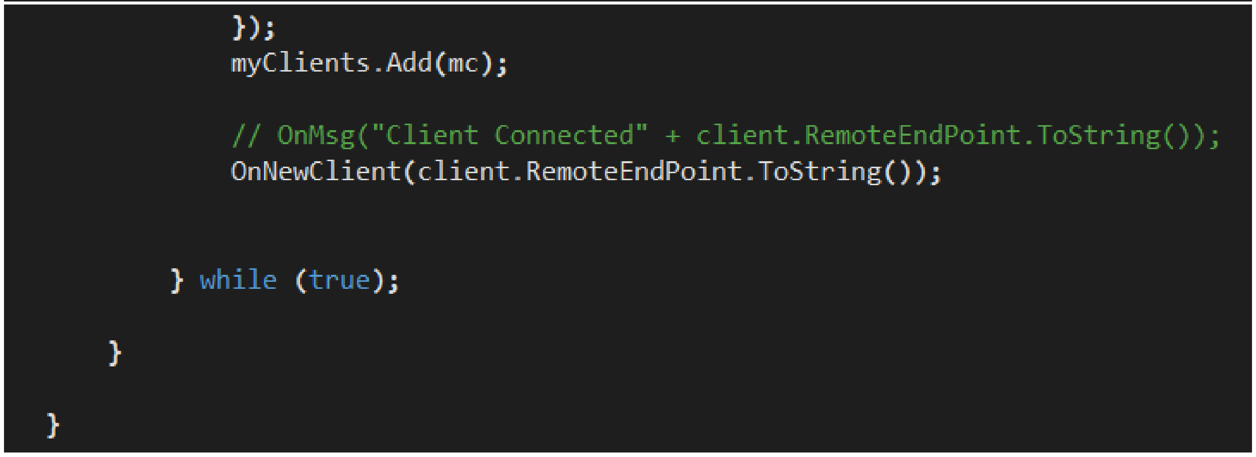
```
public class tcpServer
{

    public delegate void EventHandler(string msg);
    public delegate void OnRecv(string from, byte[] buffer, int len);
    public delegate void _OnNewClient(string client);

    public _OnNewClient OnNewClient;
    public ClientCollection myClients;
    public delegate void _OnClientDisConnect(string id);

    public event OnRecv OnMessageArriveEvent;

    public _OnClientDisConnect OnClientDisConnect;
    // public delegate void _OnConnection(Socket s);
    //public On
    public OnRecv mOnRecv;
    public EventHandler OnMsg;
    public EventHandler OnError;
     public Socket myServerSocket;
```



```
public void Start(string ip, int port = 8090)
{
    //var endpoint = new IPEndPoint(IPAddress.Loopback, port); //test
   var endpoint = new IPEndPoint(IPAddress.Parse(ip), port);
    Socket socket = new Socket(endpoint.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
    myClients = new ClientCollection();
    socket.Bind(endpoint);
    socket.Listen(int.MaxValue-1);//128
    myServerSocket = socket;
    //this.mClient = null;
    Task.Run(() => Listen(socket));
}
```



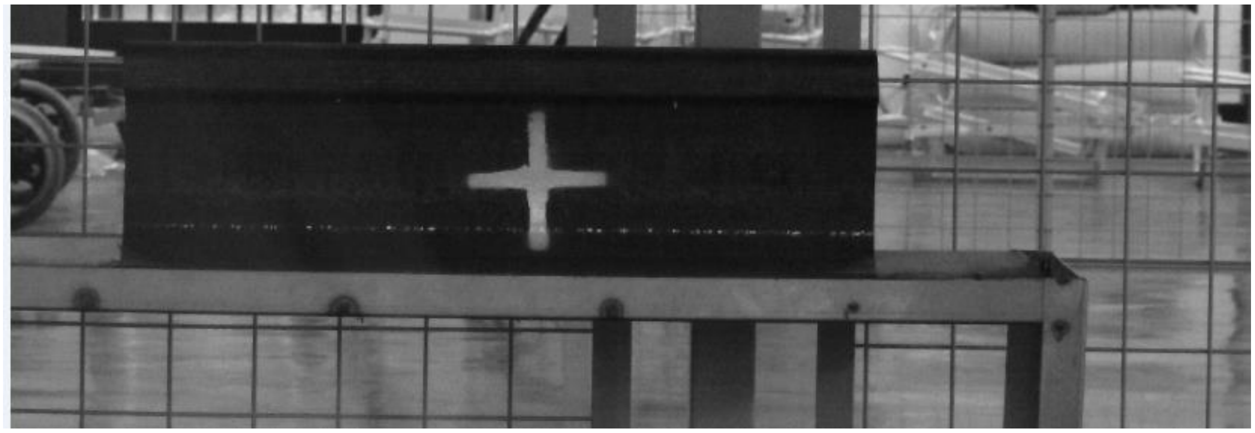
```
    private async Task Listen(Socket msocket)
    {
        do
        {

            var client = await Task.Factory.FromAsync(
            new Func<AsyncCallback, object, IAsyncResult>(msocket.BeginAccept),
            new Func<IAsyncResult, Socket>(msocket.EndAccept), null).ConfigureAwait(false);

           // client.r

            Client mc = new Client(client);
            mc.Start((m, buffer, len) =>
            {
                //OnMessageArriveEvent(m, buffer, len);
                this.mOnRecv(m, buffer, len);
            }, (id) =>
            {
                //string m = string.Format("\n{0} is DisConnected!\n", disconnect);
                this.OnClientDisConnect(id);
               // myClients.RemoveClientById(id);
```

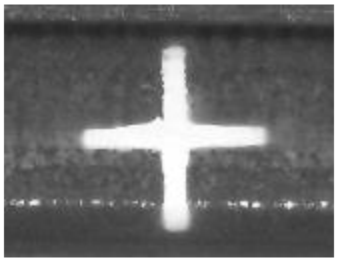


```
            });
            myClients.Add(mc);

            // OnMsg("Client Connected" + client.RemoteEndPoint.ToString());
            OnNewClient(client.RemoteEndPoint.ToString());


        } while (true);

    }

}
```

All codes are in github in my account.
All images tests , templates and references in images folder.

Ref image 0. Snapshot





Template image, you can choose any template.