

```

//James Rogers Jan 2021 (c) Plymouth University
#include<iostream>
#include<opencv2/opencv.hpp>

using namespace std;
using namespace cv;

int main()
{
    //Path to image file
    string Path = "../Task3/PCB Images/";
    // check both images
    for(int i=0; i<2;i++){
        cout << "LOADING IMAGE - ";
        if( i == 0){
            cout << "PCB.png" <<endl;
        }
        else {
            cout << "PCB_COMPLETE.png" << endl;
        }
        int count = 10;
    }
    //loop through component images
    Mat PCB;
    for(int n=0; n<10; ++n){

        //read PCB and component images
        if(i == 0){
            PCB = imread(Path+"PCB.png");
        }
        else{
            PCB = imread(Path+"PCB_COMPLETE.png");
        }
        Mat Component = imread(Path+"Component"+to_string(n)+".png");

        //=====Your code goes here=====
        Mat matchImage;
        matchTemplate( PCB, Component, matchImage, TM_SQDIFF_NORMED ); //Normed Square Difference
Matching
        double minval, maxval; //Minimum and maximum values
        Point minloc, maxloc, matchloc; //coordinates for min error,
max error, and the match location
        minMaxLoc(matchImage, &minval, &maxval, &minloc, &maxloc);
        matchloc = minloc; //SQDIFF returns 0 as a
perfect match
        rectangle(matchImage, matchloc, Point( matchloc.x + Component.cols, matchloc.y +
Component.rows), Scalar(0,0,255), 2, 8, 0);
        rectangle(PCB, matchloc, Point( matchloc.x + Component.cols, matchloc.y + Component.rows),
Scalar(0,0,255), 2, 8, 0);

        cout << matchloc << endl;
        cout << minval << endl <<endl;
        if(minval >0.01){
            cout << "Component number " << n<<" not found" << endl;
            count -= 1;
        }
        //display the results untill x is pressed
        while(waitKey(10)!='x'){
            imshow("Target", Component);
            imshow("PCB", PCB);
            imshow("Result",matchImage);
        }
    }
    cout << "Components present: " << count << "/10" << endl;
}
}

```