

# QR Code Detector

## Overview

In the world where every other person carries an android phone or iPhone, QR Codes have gained a lot of popularity. The amount of information that these small and strange looking codes can carry is huge. Whether it's a URL, or a coupon code, or a person's PayTM account details, or a package's information, these codes can store it all.

QR Code (Quick Response Code) were first designed in **1994** for automotive industry in Japan and since then they have left their mark in about every sector.

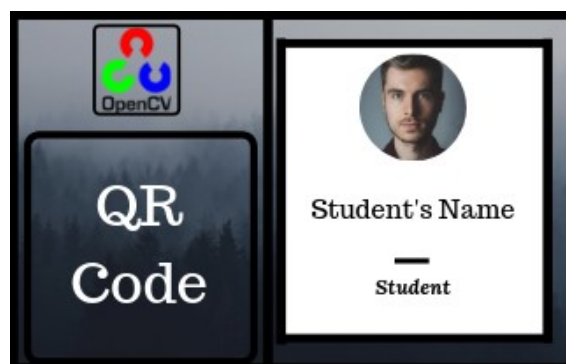
## Aim

In this project, we will implement a **QR Code Detector and Decoder** using **OpenCV** from scratch. We will then run our QR Code Detector on a sample image attached below.



## Additional Exercise

1. Create your own **OpenCV ID Card** (unofficial). You can use the following template as reference.



The QR Code should contain the following details:

Name: Your name will come here, ID: a random numeric string

Use the QR Code Detector to detect the QR Code on your ID card. Also try out different image processing techniques you have learnt so far in the course, for example, image blurring to blur the ID Card photo and see how it affects the performance of your QR Code Detector. Share your findings with other students on Discussion Forum.

1. Collect 5 photos of QR Codes that you have come across in your day-to-day life. Use our QR Code Detector to detect and decode the QR Code in the images. Don't forget to discuss your results and findings with other

## Step 0: Include Libraries

```
In [1]: #include "../resource/lib/public/includeLibraries.h"
```

```
In [2]: #include <opencv2/opencv.hpp>
```

```
In [3]: #include <opencv2/objdetect.hpp>
```

```
In [4]: #include <opencv2/imgcodecs.hpp>
```

```
In [5]: #include <opencv2/highgui/highgui.hpp>
```

```
In [6]: #include <opencv2/imgproc/imgproc.hpp>
```

```
In [7]: #include <iostream>
```

```
In [8]: #include <string>
```

```
In [9]: using namespace std;
```

```
In [10]: using namespace cv;
```

## Step 1: Read Image

```
In [11]: // Image Path
string imgPath = DATA_PATH + "/images/IDCard-Satya.png";

// Read image and store it in variable img
///
Mat img = cv::imread(imgPath);
///
```

```
In [12]: cout << img.size().height << " " << img.size().width;

204 324
```

```
In [13]: ///
/// AUTOGRADER TEST - DO NOT REMOVE
///
```

## Step 2: Detect QR Code in the Image

```
In [14]: Mat bbox, rectifiedImage;

// Create a QRCodeDetector Object
// Variable name should be qrDecoder
///
QRCodeDetector qrDecoder = cv::QRCodeDetector();
///

// Detect QR Code in the Image
// Output should be stored in opencvData
///
std::string opencvData = qrDecoder.detectAndDecode(img, bbox, rectifiedImage);
///

// Check if a QR Code has been detected
if(opencvData.length() > 0)
    cout << "QR Code Detected" << endl;
else
    cout << "QR Code NOT Detected" << endl;

QR Code Detected
```

```
In [15]: ///
/// AUTOGRADER TEST - DO NOT REMOVE
///
```

```
In [16]: ///
/// AUTOGRADER TEST - DO NOT REMOVE
///
```

## Step 3: Draw bounding box around the detected QR Code

```
In [17]: int n = bbox.rows;

// Draw the bounding box
///
for(int i = 0 ; i < n ; i++)
{
    cv::line(img, cv::Point2i(bbox.at<float>(i,0),bbox.at<float>(i,1)),
              cv::Point2i(bbox.at<float>((i+1) % n,0), bbox.at<float>((i+1) % n,1)),
              Scalar(255,0,0), 3);
}
///
```

## Step 4: Print the Decoded Text

```
In [18]: // Since we have already detected and decoded the QR Code
// using qrDecoder.detectAndDecode, we will directly
// use the decoded text we obtained at that step (opencvData)

cout << "QR Code Detected!" << endl;
///
cout << opencvData << endl;
///
```

```
QR Code Detected!
Name: Satya, ID: 1234
```

```
Out[18]: @0x490bfa0
```

```
In [19]: ///
/// AUTOGRADER TEST - DO NOT REMOVE
///
```

## Step 5: Save and display the result image

```
In [20]: // Write the result image
string resultImagePath = "./QRCode-Output.png";

///
cv::imwrite(resultImagePath, img);
///
```

```
Out[20]: true
```

```
In [21]: ///
/// AUTOGRADER TEST - DO NOT REMOVE
///
```

We will display the image with the help of markdown.

**Display the final image below using markdown**



```
In [ ]:
```