Detecting and Decoding QRCODE using OpenCV Python

INDEX

FIGURE INDEX

1. ABSTRACT: -

Barcode technology is the main pillar of the automatic identification which can be used in some real time applications with many types of codes. Multiple product’s barcode to be simultaneously detected is the paper’s proposal. The various types of code and their real time applications sometimes faces different problems, thus the improvement of the effectiveness of the code should be done persistently. The propose code which will have implementation shop billing systems and various products. The code will identify the barcodes using the image processing technique. Already available images or images taken through mobile camera will be taken to recognize the barcodes. It will not only detect 1D barcodes such as EAN-13 barcodes but also 2D barcodes such as QR codes.Mostly, it will be angle invariant, requires less user interaction and executable on available computers. Such model will help the customers to reduce the time for the billing system while buying various products in stores. The code is implemented in Python IDE using OpenCV library.

2. PROBLEM DEFINITION: -

The very first problem is the captured image by a camera may contains severallabels. These labels which contain QRCode may appear different by dependingupon on types of illumination, angle and orientation. Due to the detectionwhich is based on feature characteristics ofQRCode, distance, angles, resolution andbrightness in the original images is quite a big challenge. The recentresearch in this field is basically based on visual sensor cameras with differentresolutions.

In this work we focused on detection of different kinds of labelsand this is to realize an image for QRCode detection in the real environmentwhich is flexible to face the real life problems. In this process, sinceOpenCV does not have any dedicated modules that can be used to read and decode barcodes and QR codes .However, what OpenCV can do is facilitate the process of reading barcodes and QR codes, including loading an image from disk, grabbing a new frame from a video stream, and processing it. Once we have the image or frame we can then pass it to a dedicated Python barcode decoding library such as a Zbar. The ZBar library will then decode the barcode or QR code. OpenCV can come back in to perform any further processing and display the result.

Project Idea:

The project is based on scanning the image containing the QR code and recognizing the code and printing the information stored in the QR code.

3. LITERATURE SURVEY: -

The author Raghav Puri et al.[1]presented that almost all the products and items that exist have a unique code or an ID associated with them. This special Id is what we call a barcode. Barcode detection is very common and necessary today as it makes our job easy. Barcode has information (data bits) encoded into it and while scanning it decodes the information. The concept of image processing is used to detect the barcode extract information out of it and then use it practically.

The team of University of Technical Malayalam Melaka and University of Fukui where they are trying to recognize the Barcode using a web camera. They developed a program on MATLAB for this purpose.

Another team of the University of Szeged used morphology to have efficient detection on 1D and 2D algorithms. They have improved the detection technique from their previous alogs.

Vidya pratishtha's college of engineering Baramati ,Pune University made an android application which easily managed barcode detection from android camera of different sizes and solving the problems of uneven problems of illumination.

The author Kopila Pariyar [2] developed the process of Detecting Barcodes Using Image Processing where she gave us complete information about the barcodes like the thickness of each line represents particular value. Barcodes represents the information which is present on almost every packaging of products and it is an easy way to give quick input. There are different types of barcodes which are very popular named as, EAN, EAN-13, EAN-8, Code 39, Code 93, Code 128 and UPC (Universal Product Code) and they are called as 1D barcodes. Implementation of this project in real time basis was done by two methods which include reading barcodes using digital camera and the other one was reading barcodes using mobile phones camera. The first method was directly done from the camera images using edge detection method whereas the other approach is usually done for the 2D barcodes having four corners. The barcode recognition system using image processing is used by big organization to read their large amount of diverse data. This paper basically gives us idea about various methods of barcode detection perform within their limit individually.

A method in order to lower the consumed threshold was purpsed by Weibing Chen[3]. A practical image pre-processing method was proposed for Quick Response (QR) barcode recognition., The encoding characteristic of QR had been used instead of using the traditional methods such as edge detection and line detection.

N. M. Z. Hashim, et al [2], 'Barcode Recognition System', in a journal proposed a project to develop a barcode recognition system by using image processing. The system will be able to read barcode through an image and the system capable to capture the image by using a webcam. This project used 'MaTlab' software program to develop the system and it will integrate with webcam or digital camera. System will analyse the image and then display on the Graphical User Interface (GUI) the barcode type, data and size of the image. System is designed to recognize different types of barcode and display the data once the barcode image is captured.

4. ARCHITECTURE & BLOCK DIAGRAM: -

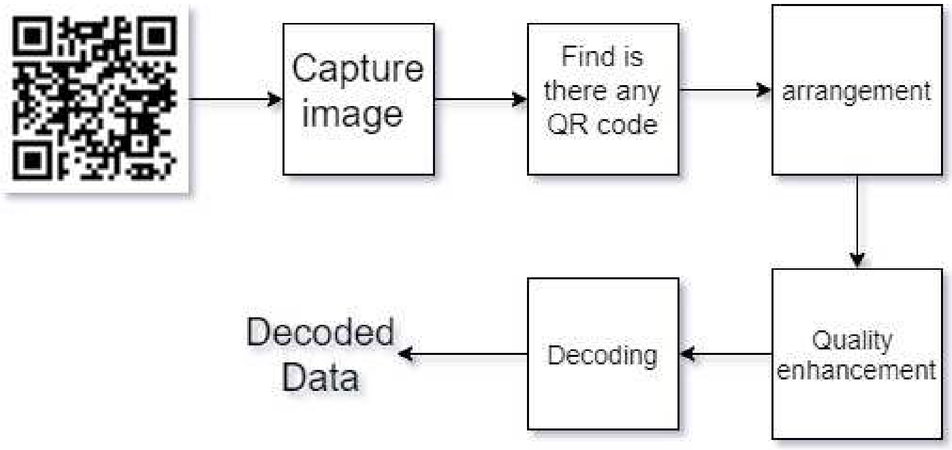


Fig 1. ARCHITECTURE AND BLOCK DIAGRAM

Basically QR code detection is very simple process in first we get a QR code from any devices such as your video for video capturing devices and your phone's camera.

Then algorithm is detecting weather in that image or video you have that QR code if it is then it will be decoding data. If not, then it will be applied algorithm to find the rectangle and check the weather for 3 points of the QR code is present on that picture if it is present that it will be crop that image and decode the data from that QR code. After that you can use your decoded data to use wherever you like.

3. FLOWCHART: -

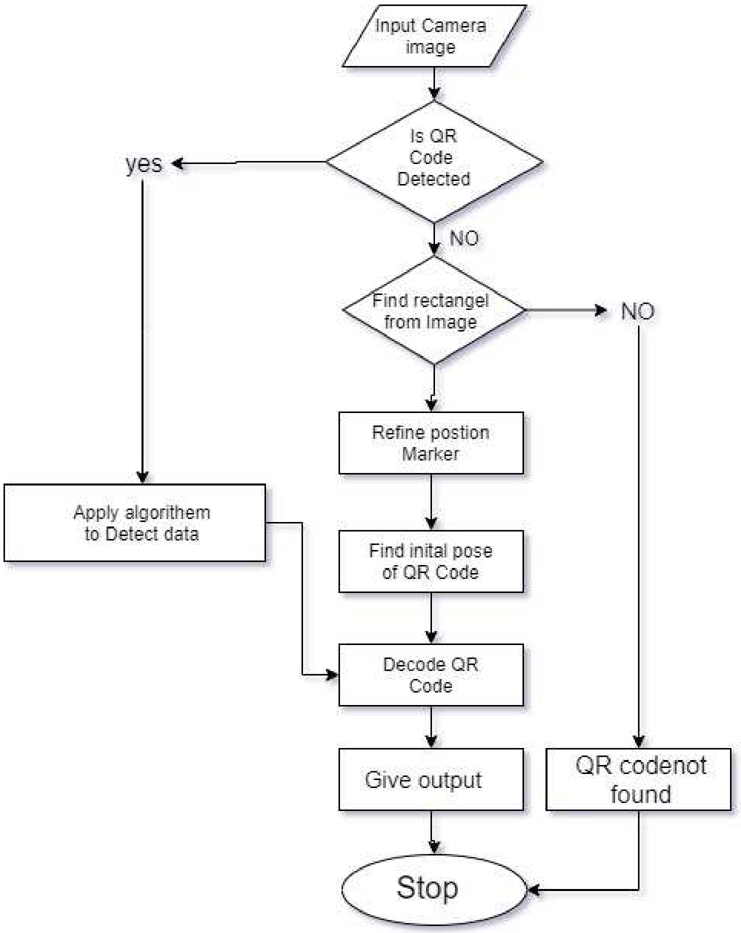


Fig 2. FLOWCHART

We detect and decode the data of QRCode using 3 different inputs which includes preloaded image, preloaded video and the live video using webcam. So in this flowchart firstly the input is given which can be any one among the above mentioned input methods and then the function detected whether there is QRCode. If the QRCode was found it apply the decode function and if not, then it finds the barcode using rectangle method and further the decode function gives the decoded output. Finally, the final output is displayed.

4. IMPLEMENTATION: -

STEP 1: ALGORITHM

We will be using OpenCV to scan and to decode the barcodes we need extra library i.e. Zbar Library. In our project we have divided our whole project into 4 parts

i. For scanning the QR Code in image

ii. For scanning QR Code in Recorded Video

iii. For Scanning QR Code through webcam(live)

iv. For checking Authorized or Unauthorized entry

The steps to create the relationship is:

The common step in all the 4 types is to import required libraries like pyzbar (use to decode the qr codes), cv2 (use for the manipulation of the image,videos), matplotlib.pyplot (use to plot images).

- Existing virtual environment that has OpenCV ready to go.
- Installation of pyzbar
- Load the input image / video by using **cv2.imread**(“Path of the file”**)** into a variable1
- By using **pyzbar.decode(variable1)** decoding the barcodes present the upload pictures and for videos **cap.read()** was used.
- Draw the barcode data and barcode type on the image/video.
- Print the barcode type and data to the terminal and show the output (for images).
- For video, release the video capture object using **cap.release()** and print the data.

For live video

- Read the frame using **cap.read()** and decode the barcode present by using **pyzbar.decode().**
- Extract the bounding box location of the barcode using **.rect** and draw the bounding box surrounding the barcode on the image by using **cv2.rectangle()**
- The barcode data is a bytes object so if we want to draw it on our output image we need to convert it to a string first.
- Draw the barcode data and barcode type on the image.
- Stop if “**q**” key is pressed.
- Release the VideoCapture object.

To check Authorised and Unauthorised

- Load dataset. In our case dataset is a text file which contains information which are authorised. To open our data file **with open(“filename.txt”) as file:**
- To read the text in the data file we provided we used **file.read().splitlines().**
- Read the frame using **cap.read()** and decode the barcode present by using **pyzbar.decode().**

- Extract the bounding box location of the barcode using **.rect** and draw the bounding box surrounding the barcode on the image by using **cv2.rectangle()**
- The barcode data is a bytes object so if we want to draw it on our output image we need to convert it to a string first.
- Using **if-else** check whether the data obtained from the live scanning of barcode is already present in the list i.e. datafile, if present print Authorised else Unauthorised.
- Draw the barcode data and barcode type on the image.
- Stop if "**q**" key is pressed.
- Release the VideoCapture object.

STEP 2: SOURCE CODE

```
                              """Libraries Requried"""

from pyzbar import pyzbar          #use to decode the qr codes
import cv2                         #use for the manipulation of the image,videos
import matplotlib.pyplot as plt    #use to plot images


                         """For scanning the QR Code in image"""

# load the input image
image = cv2.imread(r"C:\Users\ASUS\Desktop\All Related to  Project'\Project\barcode.jpeg")
# find the barcodes in the image and decode each of the barcodes
plt.imshow(image)
plt.show()
barcodes = pyzbar.decode(image)

# loop over the detected barcodes
for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the bounding box surrounding the
barcode on the image
    #barcodeType="notfound"
    (x, y, w, h) = barcode.rect
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on our output image we need to convert it
to a string first
    barcodeData = barcode.data.decode("utf-8")
    barcodeType = barcode.type

    # draw the barcode data and barcode type on the image
    text = "{} ({})".format(barcodeData, barcodeType)
    cv2.putText(image, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,0.5, (0, 0, 255), 2)

    # print the barcode type and data to the terminal
    print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))
    print(x,y,x+w,y+h)

# show the output image
plt.imshow(image)
plt.show()
```

```
""""for scanning QR Code in Recorded Video""""

cap = cv2.VideoCapture('video2.mp4')
while (True):
# Read the frame
    check,img = cap.read()

    barcodes = pyzbar.decode(img)
    for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the bounding box surrounding the barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

    # the barcode data is a bytes object so if we want to draw it on our output image we need to convert it to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

    # draw the barcode data and barcode type on the image
        text = "{} ({})".format(barcodeData, barcodeType)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)

    cv2.imshow('img', img)
    crop_img = img[y:y+h, x:x+w]

    # Stop if 'q' key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break

# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
print("[INFO] Found {} barcode: {}".format(barcodeType, barcodeData))


""""For Scanning QR Code during live videorecording""""

cap = cv2.VideoCapture(0)
while (True):
# Read the frame
    check,img = cap.read()
    barcodes = pyzbar.decode(img)

    for barcode in barcodes:
    # extract the bounding box location of the barcode and draw the bounding box surrounding the barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
    # the barcode data is a bytes object so if we want to draw it on our output image we need to convert it
to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

    # draw the barcode data and barcode type on the image
        text = "{} ({})".format(barcodeData, barcodeType)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)
    cv2.imshow('img', img)

    # Stop if escape key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break

# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
```

""""for checking Authorized or Unauthorised entry""""

```
#Files containing QR Codes
with open('datafile.txt') as file:
    qrlist=file.read().splitlines()
print(qrlist)

cap = cv2.VideoCapture(0)
while (True):

# Read the frame
    check,img = cap.read()
    barcodes = pyzbar.decode(img)
    for barcode in barcodes:

    # extract the bounding box location of the barcode and draw the bounding box surrounding the
barcode on the image
        (x, y, w, h) = barcode.rect
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

    # the barcode data is a bytes object so if we want to draw it on ur output image we need to convert it
to a string first
        barcodeData = barcode.data.decode("utf-8")
        barcodeType = barcode.type

        if barcodeData in qrlist:
            myOutput = 'Authorized'
            myColor = (0,255,0)
        else:
            myOutput = 'Un-Authorized'
            myColor = (0, 0, 255)
    # draw the barcode data and barcode type on the image
        text = "{} ({}) {}".format(myOutput,barcodeType,barcodeData)
        cv2.putText(img, text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX,
            0.5, (0, 0, 255), 2)
    cv2.imshow('img', img)
    # Stop if escape key is pressed
    k = cv2.waitKey(1) #& 0xff
    if k==ord('q'):
        break

# Release the VideoCapture object
cap.release()
# out.release()
cv2.destroyAllWindows()
```

5. RESULT & DISCUSSION: -

We divided our project into 4 parts and obtained the desired result. Thereby I will be discussing results of all the parts.

i. For scanning the QR Code in image
In this section we required preloaded images which contain one or many or no barcodes with different annotations and size. Later our code detected barcodes i.e.., QR code and decoded the data stored in that barcodes respectively while highlighting them.

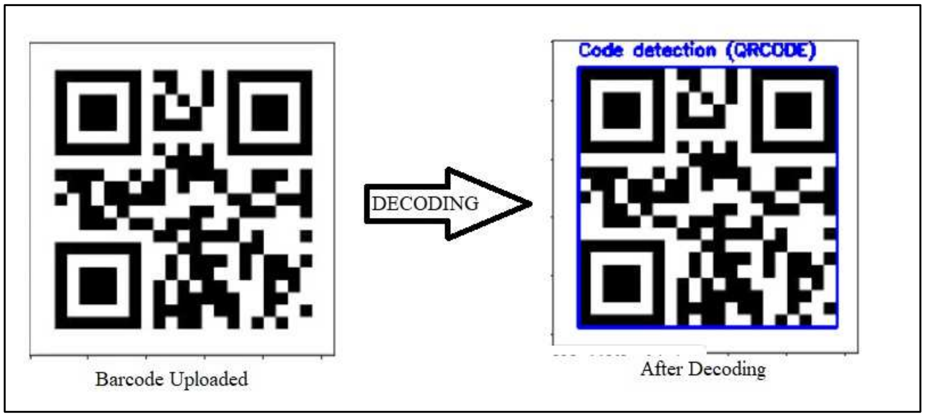
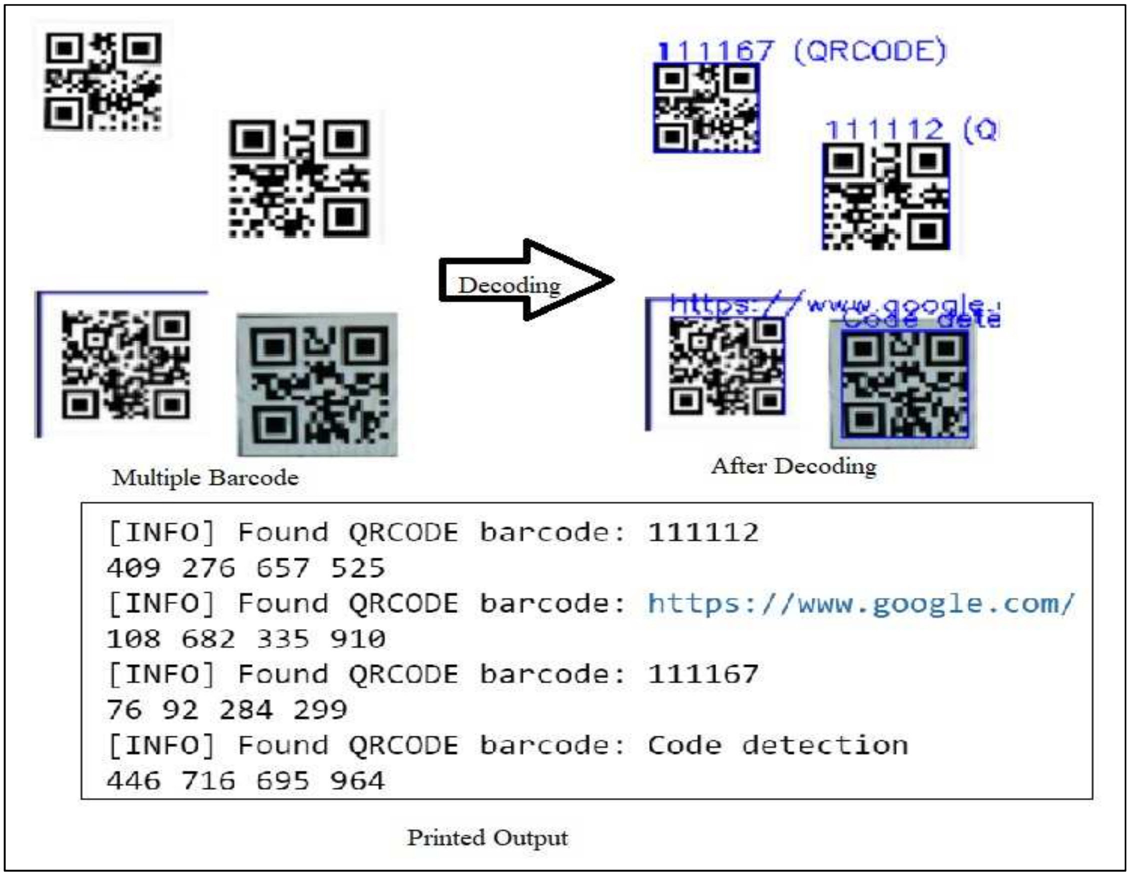


Fig 3. Using One Barcode

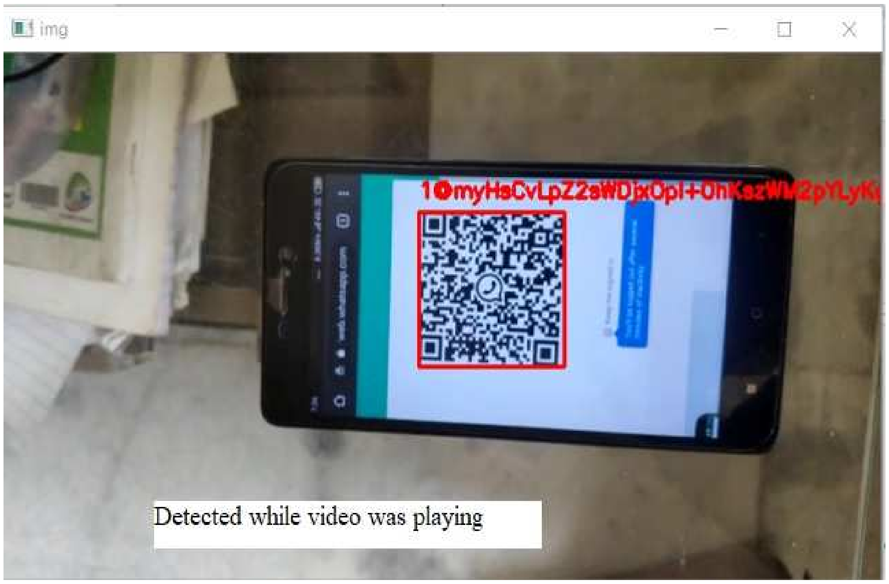


```
[INFO] Found QRCODE barcode: 111112
409 276 657 525
[INFO] Found QRCODE barcode: https://www.google.com/
108 682 335 910
[INFO] Found QRCODE barcode: 111167
76 92 284 299
[INFO] Found QRCODE barcode: Code detection
446 716 695 964
```

Fig 4. Using Multiple Barcode

ii. For scanning the QR code in pre-recorded video

In this section we required preloaded videos which contain one or many or no barcodes with different annotations and size. Later our code detected barcodes i.e.., QR code and decoded the data stored in that barcodes respectively while highlighting them.



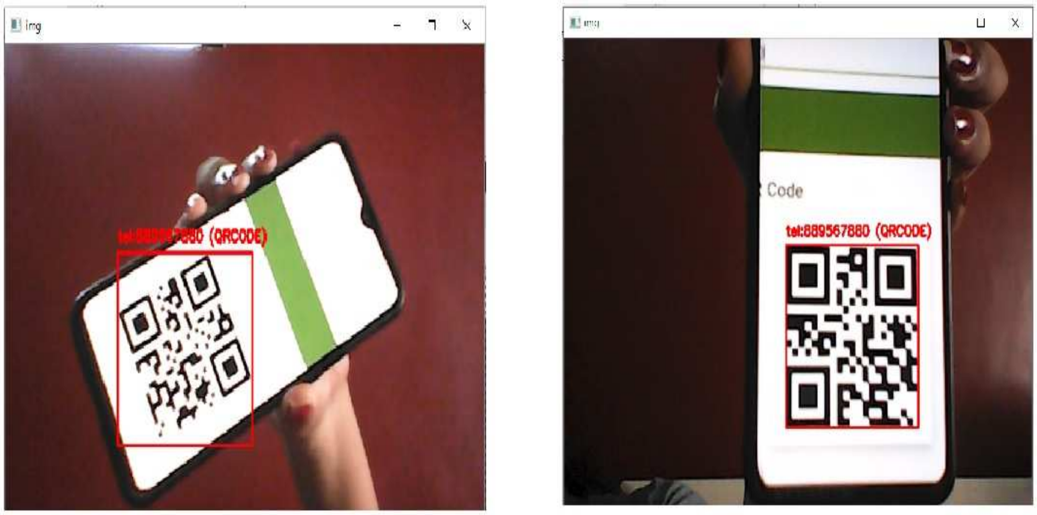
1@myHsCvLpZ2aWDjxOpl+0nKszWM2pYLyK

Detected while video was playing

Fig 5. Scanning of QR code in a pre-recorded video

iii. For scanning the QR code in live video

In this section we just our webcam to scan the required QR code by changing the distance between the barcode and webcam, changing its annotations.



tel:889567880 (QRCODE)

tel:889567880 (QRCODE)

Fig 6. Detection and displaying of data decoded of single barcode

Due to different size and quality of barcode and unsteady movement the barcodes were detected at different distance from the camera.

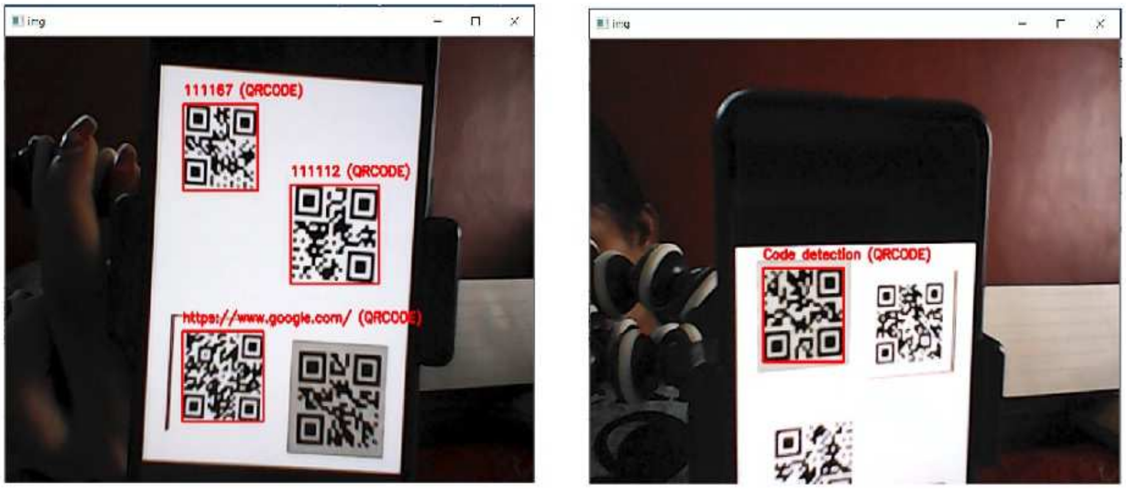
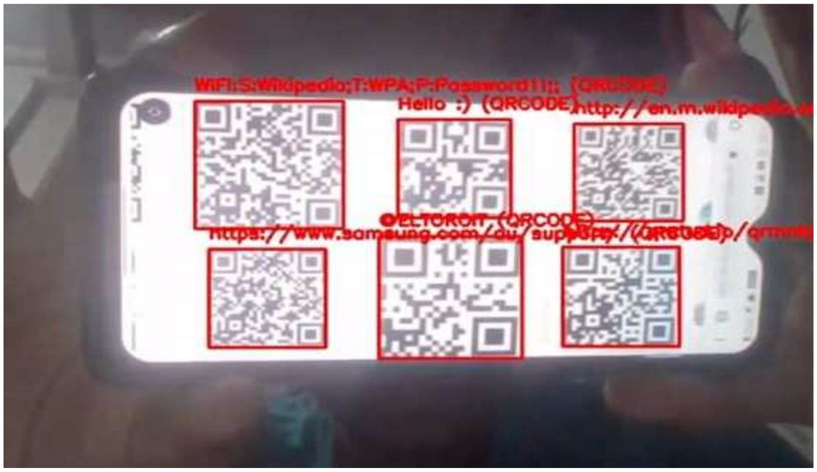
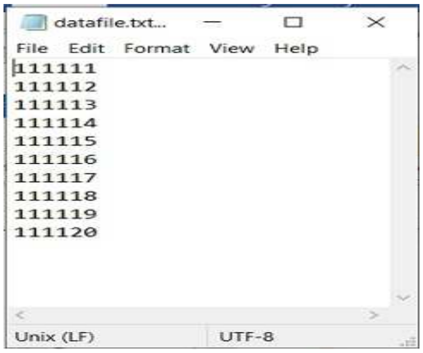


Fig 7. Detection and displaying of data decoded of multiple barcode


WIFI:S:Wikipedia;T:WPA;P:Password11;; (QRCODE)
Hello :) (QRCODE)
http://en.m.wikipedia.w
QELTOROIT (QRCODE)
https://www.samsung.com/au/support/
(QRCODE)

iv. To check whether the QR Code is Authorised or Unauthorised

In this section we uploaded a dataset (i.e. a text file with data particular numbers which can be presented as authorised employee number). Later we scanned QR code with different data and compared with the dataset to give the result for the same.



```
111111
111112
111113
111114
111115
111116
111117
111118
111119
111120
```

Fig 8. Dataset (.txt file)

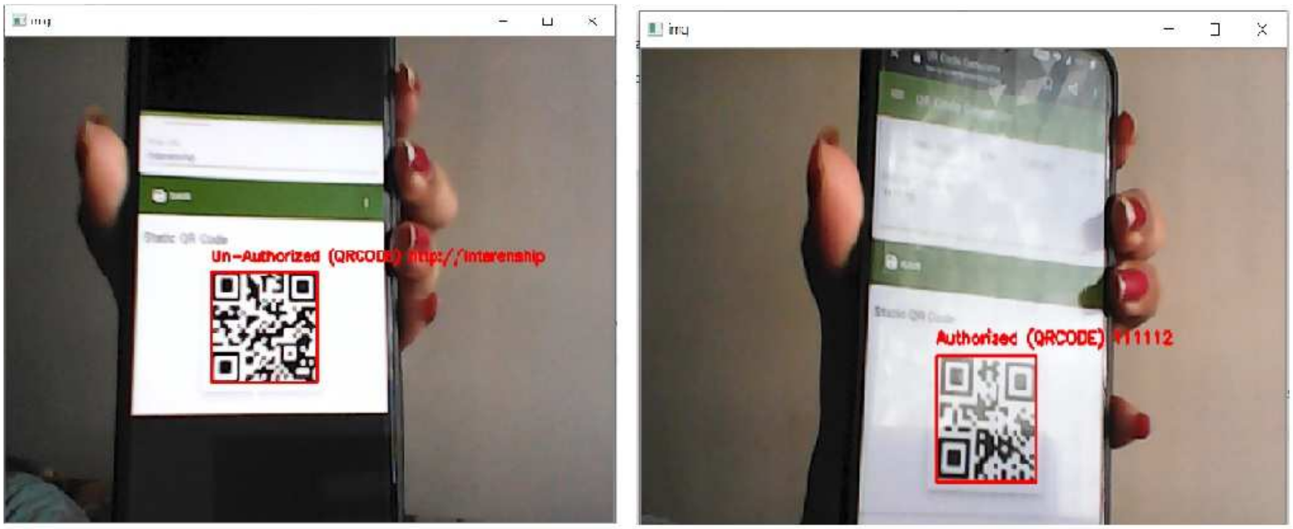

img
Static QR Code
Un-Authorized (QRCODE) http://Interenship
img
Static QR Code
Authorised (QRCODE) 11112

Fig 9. Detection of QR code and displaying whether the QR code is authorised or not

6. CONCLUSION: -

QR-Code reader can be implemented in many ways and one of the way is what we have done using pybzar library. It is also an easy concept as it requires only open source and numpy library along with pybzar, this can be used within a particular organization to check the authorised I'd cards by making proper data base which is an essential part of taking attendance and many other things. Here we have not only learnt about different kinds of barcode used but also find many applications on real time basis. The different techniques to extract the data from the barcodes using image, preloaded videos and even for live-videos gives the scope for preserving the necessary details.

7. FUTURE SCOPE: -

Image Processing has taught us to see the upcoming future technologies. Our project was implemented using basics of image processing with open cv python. Later as we grow in our skills this project can be taken to greater level by making it into an application. From our project we can make two applications.

One of the application can be a QR code scanner for both desktop as well as mobile. We can perform three operations.

- Ask user to upload or open the barcode image to scan and display result.
- Ask user to upload or open the barcode videos to scan and display result.
- Ask user to bring the image in front of the camera, scan and immediately display result.

Another application of opposite can be to scan whether the is the employee of a company or not. As we know that anyone can get a hold on company ID card and misuse it. Hence we can provide individual employee an ID card with their unique QR code which will be containing data of that particular employee. Letter this application can be connected with a safety or an entry gate barrier. When a person its entry in the company has to scan his or her ID if the display shows that the persons authorised the entry barrier gate will open or else it will remain in the lock condition and the security system will be informed immediately.

8. REFERENCES: -

- https://www.irjaes.com/pdf/V4N1Y18-IRJAES/IRJAES-V4N1P110Y19.pdf
- https://docs.opencv.org/3.4/de/dc3/classcv_1_1QRCodeDetector.html
- https://stackoverflow.com/questions/28210912/how-to-detect-a-single-qr-code-using-opencv-python-and-zbar
- https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/
- https://www.learnopencv.com/barcode-and-qr-code-scanner-using-zbar-and-opencv/
- https://techtutorialsx.com/2020/01/01/python-pyzbar-detecting-and-decoding-barcode/

GROUP MEMBERS:

1. **Neha Jaiswar (SE E&TC A 23)** ______________________________
2. **Riya Gupta (SE E&TC A 15)** ______________________________
3. **Rohit Pal (SE E&TC A 46)** ______________________________
4. **Aniket Vishwakarma (FE CMPN 50)** ______________________________
5. **Amrita Gupta (FE CMPN 51)** ______________________________
6. **Vrushti Gangwal (FE CMPN 46)** ______________________________

SUPERVISOR:

1. **Ms. Anupriya Babbar** ______________________________
2. **Mr. Shailesh Sangle** ______________________________