

# SECURE GATE PASS SYSTEM THROUGH DETECT, VISION AND SPEECH SYSTEM (DVS SYSTEM)

H.Jayasree<sup>1</sup>, Shreejaa Talla<sup>2</sup>, Riya Punia<sup>3</sup>

<sup>1</sup>Assoc.Prof., <sup>2,3</sup>IV/IV-IIsem B.E., Dept. of CSE,  
MVSR Engineering College, Nadergul, Telangana, India

**Abstract:** Security check of vehicles, material that get into the organization and that are to be transmitted in various sections usually takes place through manual data entry and checking may sometimes leads to errors, is time consuming and requires lot of man-labour. Data entry is manually handled which does not provide proper security to data and data stored on papers is subject to loss due to physical damage. DVS system (Detect, Vision & Speech) is an automated version of existing Material Gate Pass system. The purpose of this system is to detect a vehicle, generate and scan QR-code on vehicle and produce relevant voice synthesized instructions on each operation which is achieved by using a Python Library named PYTTSSX (Pyttssx is a cross-platform speech (Mac OSX, Windows, and Linux) library. Voice metadata can be set to age, gender, id, language or name. The speech engine comes with a large amount of voices. A text-to-speech (TTS) system converts normal language text into speech. As security is a major concern now a day in every organization, this System facilitate for a smooth entry and exit in the organization without doing any kind of manual checking. DVS System prevents the theft or unauthorized carry.

**Index Terms:** Tensorflow, QRcode, RCNN, ROI

## I. INTRODUCTION

The vehicle loaded with material is detected by Object detection - the process of finding real-world object instances like car, bike, TV, flowers, and humans in still images or Videos shown in Figure1. It allows for the recognition, localization, and detection of multiple objects within an image which provides us with a much better understanding of an image as a whole. It is commonly used in applications such as image retrieval, security, surveillance, and advanced driver assistance systems that can be implemented using Deep Neural network with supervised learning via Tensorflow library<sup>11</sup>: Tensorflow is Google's Open Source Machine Learning Framework for dataflow programming across a range of tasks. Nodes in the graph represent mathematical operations, while the graph edges represent the multi-dimensional data arrays (tensors) communicated between them. Tensors are just multidimensional arrays, an extension of 2-dimensional tables to data with a higher dimension. There are many features of Tensorflow which makes it appropriate for Deep Learning.

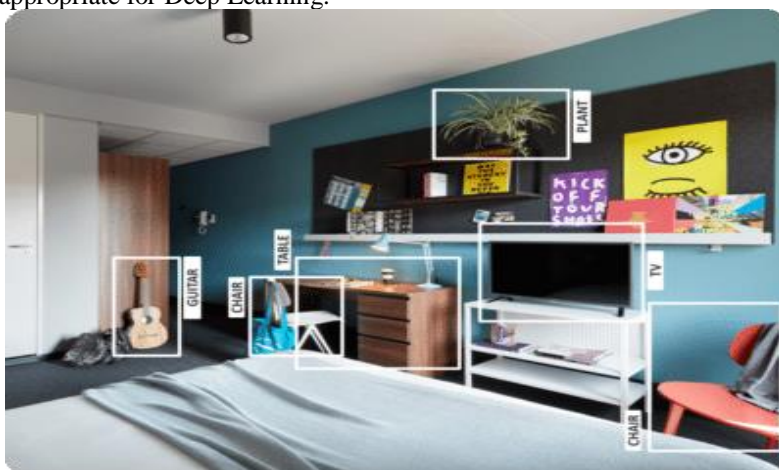


Figure 1 : Detected objects using Tensorflow

Furthermore it consists of a pertinent QR-code (containing the details or information about the vehicle that transport materials) that is generated and further scanning is done by using Zbar [1] library of python for scanning the QR-code. This system consists of two different phases: one is Admin phase and other is Security phase. Admin generates a particular format of QR-code inclusive of materials present in a vehicle. Furthermore, it checks for the message or details provided by security and maintain the database by clearing unnecessary data or invalid vehicle's data. Security detects the vehicle, scans the QR-code and sends appropriate message to admin. Scanning QR-code is only accessed under authorized conditions.

## II. BACK GROUND WORK

### 2.1 History of Material Gate Pass Systems

In many large organizations, a lot of movements take place every day. It may be employees moving in or out, movement of raw materials for production etc. All these movements of the organization's premises, factories and companies need to be monitored and controlled. This is where the concept of a gate pass comes into play. Gate pass can be used to authorize the movements of humans, materials and machines to or from the premises of the organization. It will help to monitor and track all the movements happening in an organization.

**What is gate pass?** Gate pass is simply a printed document that needs to be filled out by the person who wants to move out of the premises of the organization. The person is supposed to fill out the necessary information including the purpose of travel to get approval. Usually the approval is done by authorized persons that just sign off the filled out gate will pass document if he/she feels that the reason for moving out or in the organization is acceptable. The authorized gate pass document is verified by the security persons in charge of the gate and they will allow the movement only if they see the authorized gate pass.

**Types of gate passes:** There are generally two types of gate passes. Employee pass and material pass. Since most of the movement are either humans going in or out or materials being shipped in or out, two different gate passes can be made for these two purposes.

**Employee gate pass:** It is for employees who want to move out. The reason may be personal or professional. Either way, he/she would have to fill out the employee gate pass and submit for the approval of the higher official. The higher official will sign off the document after noting down the movement. Finally, the employee can submit the document at gate to move out. This is an effective way to restrict unauthorized movement.

**Materials gate pass:** For moving materials, three copies of the gate pass are made. The passes are signed by the authorized person. Two copies of the pass are given to the person responsible for the movement (usually the driver) and one of the copies will be held by the security team at the gate. The other copy will be kept by the driver until the delivery reaches its destination.

**Then:** In many organizations and companies Gate entry is controlled by two main gates. Guest access by car requires gate pass form signed and faxed to Organization Security office by the inviting person. This is not an efficient way and reasons are pretty simple. In some organizations, even in this modern era Gateman / Guard prefers a manual entry in registers about person or vehicle. This means, no authentication is done by Guard and which result in many miss happenings.

**The problem rose with this system:** Any fax owner worldwide can issue a gate pass. This results in undesired people or vehicle entry in companies' premises. There is no control over the guest arrival. Anyone willing to enter can enter in the company. It may be possible some unauthorized may enter and park their vehicle in company parking area and police start asking a question to guard and they are helpless. There is no way of knowing who and when and by who was invited. An entry in register does not mean that information is true. There is no authentication process. The proposed system will replace the current tax policy and register entry by sophisticated Intranet system to overcome the above problems. Noting data in the register is very laborious works as well as time-consuming.

**Benefits of using gate pass:** There are a lot of advantages if gate pass system is implemented in an organization. The first and foremost being the restriction of unauthorized movement in the organization's premises, and apart from this, the gate pass allows the organization to get a record of the time of movement; and to track the person responsible for the movement. In the case of materials, gate pass prevents the theft or unauthorized carry. It also improves the discipline inside the organization.

The basic operation of DVS system is to detect via camera and send the flag to security that the required object is detected. Furthermore, voice instructions are to be generated to provide movement and flow, then QR code scanner scans the details and cross verifies for the validity. The validity of scanned details is obtained from the database stored within admin consisting of two phases: returnable details and non-returnable details. Admin clears the data of non-returnable object and saves only the return date of returnable objects. Additionally once the verification is done process clears up all previous details except few exceptions. The proposed system model is shown in Figure 2.

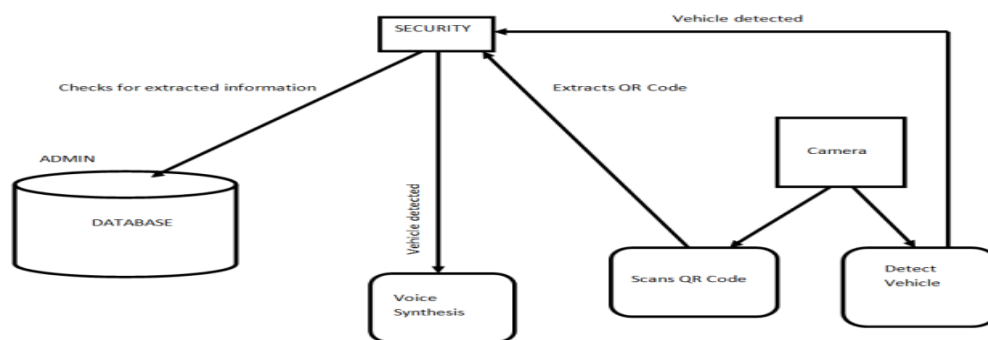


Figure 2: System model

### III. DVS SYSTEM

#### 3.1 Object Detection

##### How TensorFlow works?

TensorFlow allows developers to create dataflow graphs—structures that describe how data moves through a graph, or a series of processing nodes. Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor. TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications. The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together. TensorFlow applications can be run on most any target that's convenient: a local machine, a cluster in the cloud, iOS and Android devices, CPUs or GPUs. If you use Google's own cloud, you can run TensorFlow on Google's custom TensorFlow Processing Unit (TPU) silicon for further acceleration. The resulting models created by TensorFlow, though, can be deployed on most any device where they will be used to serve predictions.

**Algorithm - Faster RCNN:**

1. We take an image as input and pass it to the ConvNet which returns the feature map for that image.
2. Region proposal network is applied on these feature maps. This returns the object proposals along with their objectness score.
3. A ROI pooling layer is applied on these proposals to bring down all the proposals to the same size.
4. Finally, the proposals are passed to a fully connected layer which has a softmax layer and a linear regression layer at its top, to classify and output the bounding boxes for objects.

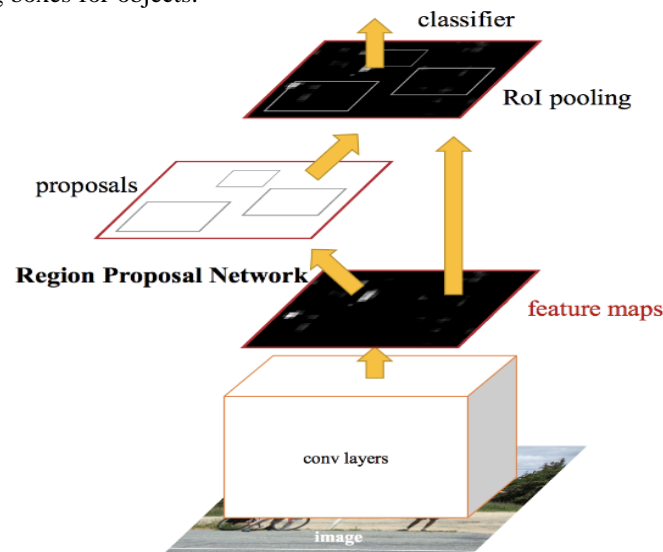


Figure 3: Regional Proposal Network

**RPN Working:** To begin with, Faster RCNN takes the feature maps from CNN and passes them on to the Region Proposal Network. RPN uses a sliding window over these feature maps, and at each window, it generates  $k$  Anchor boxes of different shapes and sizes:

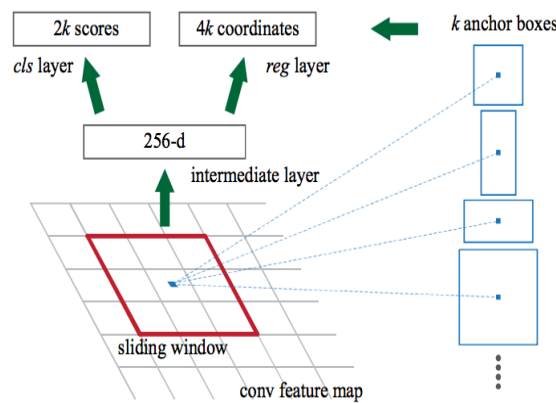


Figure 4: Inner layers of RPN

Anchor boxes are fixed sized boundary boxes that are placed throughout the image and have different shapes and sizes. For each anchor, RPN predicts two things: i) The first is the probability that an anchor is an object (it does not consider which class the object belongs to) ; ii) Second is the bounding box regressor for adjusting the anchors to better fit the object.

We now have bounding boxes of different shapes and sizes which are passed on to the ROI pooling layer. Now it might be possible that after the RPN step, there are proposals with no classes assigned to them. We can take each proposal and crop it so that each proposal contains an object. This is what the ROI pooling layer does. It extracts fixed sized feature maps for each anchor.

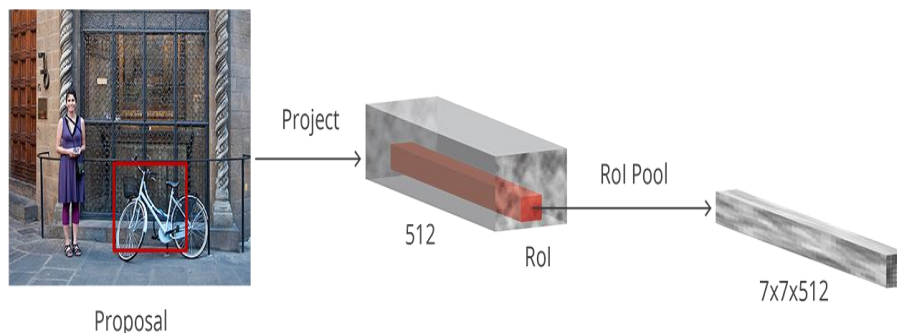


Figure 5: Working of ROI pool

Then these feature maps are passed to a fully connected layer which has a softmax and a linear regression layer. It finally classifies the object and predicts the bounding boxes for the identified objects.

### Problems with Faster RCNN

All of the object detection algorithms we have discussed so far use regions to identify the objects. The network does not look at the complete image in one go, but focuses on parts of the image sequentially. This creates two complications: i) The algorithm requires many passes through a single image to extract all the objects; ii) As there are different systems working one after the other, the performance of the systems further ahead depends on how the previous systems performed.

### 3.2 QR-Code Generation

A QR Code is a two-dimensional square barcode which can store encoded data; most of the time the data is a link to a website (URL).

#### Algorithm:

Step 1: Creating QR codes – the QR Code object is designed to be smart about how it constructs QR codes. It can automatically figure out what mode and version to use to construct a QR code based on the data and the amount of error correction. The error correction level defaults to the highest possible level of error correction.

Step 2: Encoding Data – the standard calls the data's encoding its mode. The QR code standard defines how to encode any given piece of data. There are four possible modes. This supports three of them: numeric, alphanumeric and binary. How much data a code version can hold is dependent on what mode is used and the error correction level. For example, the binary encoding always requires more code words than the numeric encoding. Because of this, it is generally better to allow the QRCode object to auto-select the most efficient mode for the code's contents.

Step 3: Rendering QR codes – there are five possible formats for rendering the QR Code [3]. The first is to render it as a string of 1's and 0's. Next, the code can be displayed directly in compatible terminals. They also take a scaling factor, that way each module is not rendered as 1 pixel, here image rendering is used.

### 3.3 QR-code Scanner

OpenCV can facilitate the process of reading barcodes and QR codes[1], 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.

#### Algorithm:

1. Extracts the bounding box (x, y) - coordinates from the real - time object video, that enables to localize and determine where the current barcode is.
2. Draw a bounding box rectangle in the video frame around the detected barcode.
3. Decode the barcode into an "utf-8" string and extract the type of barcode.
4. Format and draw the barcodeData and barcodeType on the image.
5. And finally, output the same data and type information to the terminal for debugging purposes

### 3.4 PYTTTSX

Pytttsx[8] is a cross-platform speech (Mac OSX, Windows, and Linux) library. You can set voice metadata such as age, gender, id, language and name. The speech engine comes with a large amount of voices. A text-to-speech (TTS) [2] system converts normal language text into speech.

## IV. ENVIRONMENT SETUP

The following Python libraries are used for implementation: QRTOOLS – to read QR codes and generate them ; PILLOW - to open, manipulate, and save different image file formats ; MATPLOTLIB – a plotting library that provides an object-oriented API for embedding plots into ; PYZBAR -to detect and decode barcodes and QR codes of different types ; ARGPARSE - to write user-friendly command-line interfaces. argparse will figure out how to parse arguments out of sys.argv ; IMUTILS - to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images ; PYTHON-OPENCV - a highly optimized library for numerical operations with a MATLAB-style syntax to solve computer vision problems. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib; TENSORFLOW – TensorFlow<sup>7</sup> is an open source software library released in 2015 by Google to make it easier for developers to design, build, and train deep learning models. CYTHON - a superset of the Python programming language, designed to give C-like performance with code that is written mostly in Python with optional additional C-inspired syntax. Cython is a compiled language that is typically used to generate CPython extension modules; PYPNG - to read and write all PNG formats. PNG supports a generous variety of image formats ; LXML - aims to provide a Pythonic API by following as much as possible the ElementTree API ; PROTOBUF - protocol buffers are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data ; COCO API - COCO is a large image dataset designed for object detection, segmentation, person keypoints detection, stuff segmentation, and caption generation. This package provides Matlab, Python, and Lua APIs that assists in loading, parsing, and visualizing the annotations in COCO; CONTEXTLIB - This function is a decorator that can be used to define a factory function for async with statement asynchronous context managers, without needing to create a class or separate \_\_aenter\_\_() and \_\_aexit\_\_() methods. It must be applied to an asynchronous generator function; PYTTTSX – Pytttsx[5] is a cross-platform (Mac OSX, Windows, and Linux) library. You can set voice metadata such as age, gender, id, language and name. The speech engine comes with a large amount of voices. A text-to-speech (TTS) system converts normal language text into speech.

Creating Environment for object detection and qr-code scanning :

In the command terminal that pops up, create a new virtual environment called “tensorflow1” by issuing the following command:

```
C:\> conda create -n tensorflow1 pip python=3.5
```

Then, activate the environment by issuing:

```
C:\> activate tensorflow1
```

Install tensorflow in this environment by issuing:

```
(tensorflow1) C:\> pip install --ignore-installed --upgrade tensorflow
```

Install the other necessary packages by issuing the following commands:

```
(tensorflow1) C:\> conda install -c anaconda protobuf
```

```
(tensorflow1) C:\> pip install pillow
(tensorflow1) C:\> pip install lxml
(tensorflow1) C:\> pip install Cython
(tensorflow1) C:\> pip install jupyter
(tensorflow
1) C:\> pip install matplotlib
(tensorflow1) C:\> pip install pandas
(tensorflow1) C:\> pip install opencv-python
(tensorflow1) C:\> pip install imutils
(tensorflow1) C:\> pip install pandas
(tensorflow1) C:\> pip install pyzbar
(tensorflow1) C:\> pip install argparse
(tensorflow1) C:\> pip install pywin32 pypiwin32pytsx3
```

A PYTHONPATH variable must be created those points to the \models, \models\research, and \models\research\slim directories. Do this by issuing the following commands (from any directory):

```
(tensorflow1)C:\>setPYTHONPATH=C:\tensorflow1\models;C:\tensorflow1\models\research;C:\tensorflow1\models\research\slim
```

Compile the Protobuf files, which are used by TensorFlow to configure model and training parameters. Unfortunately, the short protoc compilation command posted on TensorFlow's Object Detection API installation page does not work on Windows. Every .proto file in the \object\_detection\protos directory must be called out individually by the command.

In the Anaconda Command Prompt, change directories to the \models\research directory and copy and paste the following command into the command line and press Enter:

```
protoc
--python_out=. .\object_detection\protos\anchor_generator.proto .\object_detection\protos\argmax_matcher.proto .\object_detection\protos\bipartite_matcher.proto .\object_detection\protos\box_coder.proto .\object_detection\protos\box_predictor.proto .\object_detection\protos\eval.proto .\object_detection\protos\faster_rcnn.proto .\object_detection\protos\faster_rcnn_box_coder.proto .\object_detection\protos\grid_anchor_generator.proto .\object_detection\protos\hyperparams.proto .\object_detection\protos\image_resizer.proto .\object_detection\protos\input_reader.proto .\object_detection\protos\losses.proto .\object_detection\protos\matcher.proto .\object_detection\protos\mean_stddev_box_coder.proto .\object_detection\protos\model.proto .\object_detection\protos\optimizer.proto .\object_detection\protos\pipeline.proto .\object_detection\protos\post_processing.proto .\object_detection\protos\preprocessor.proto .\object_detection\protos\region_similarity_calculator.proto .\object_detection\protos\square_box_coder.proto .\object_detection\protos\ssd.proto .\object_detection\protos\ssd_anchor_generator.proto .\object_detection\protos\string_int_label_map.proto .\object_detection\protos\train.proto .\object_detection\protos\keypoint_box_coder.proto .\object_detection\protos\multiscale_anchor_generator.proto .\object_detection\protos\graph_rewriter.proto
```

run the following commands from the C:\tensorflow1\models\research directory:

```
(tensorflow1) C:\tensorflow1\models\research> python setup.py build
(tensorflow1) C:\tensorflow1\models\research> python setup.py install
```

## V. RESULTS

### 5.1 Output Screens

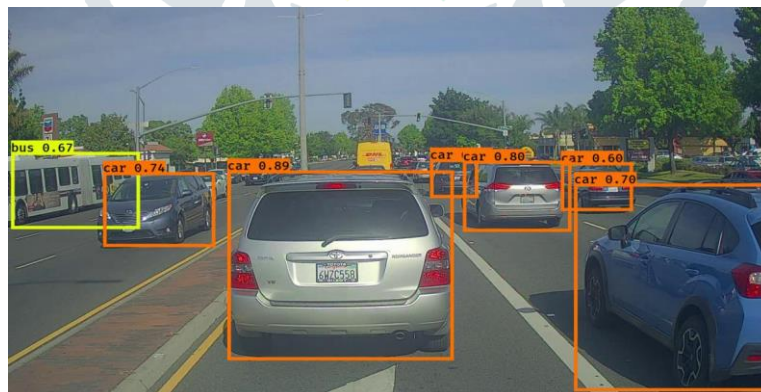


Figure 6: Vehicles detected on webcam footage

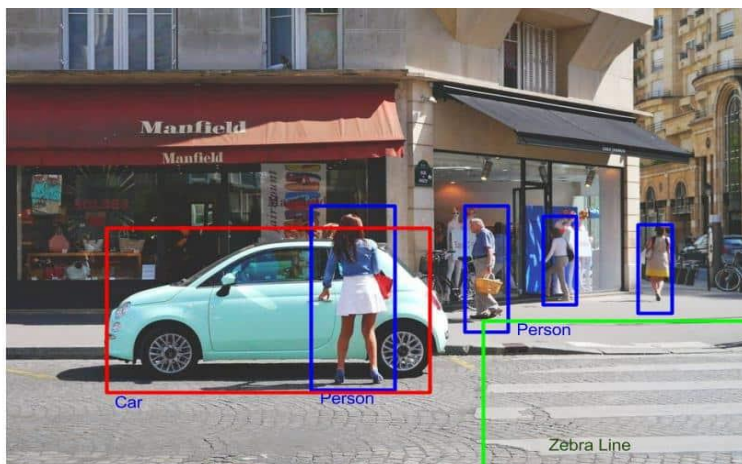


Figure 7: all objects gets detected

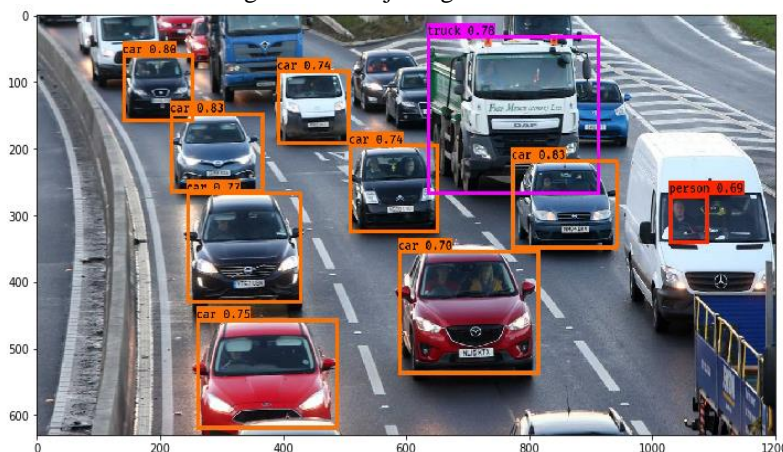


Figure 8: Different type of Vehicles detected on webcam footage

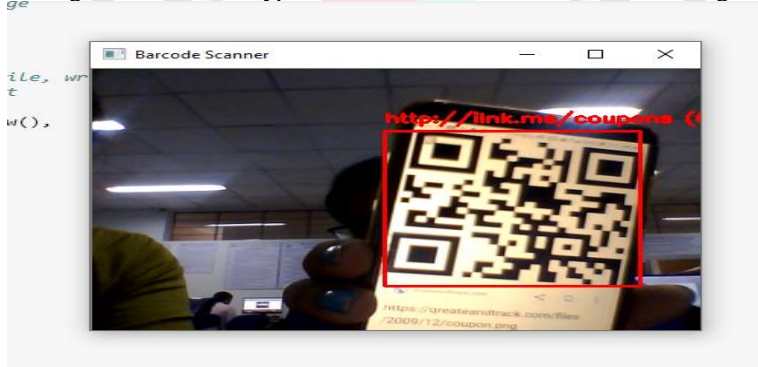


Figure 9: QR code scanned

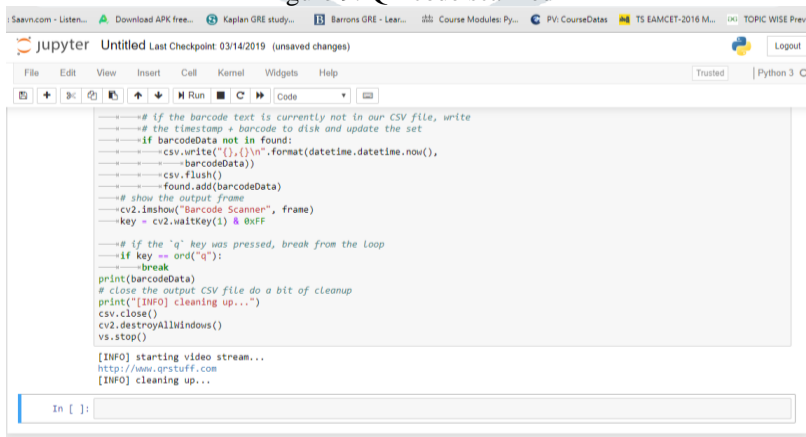


Figure 10: Website link retrieved from QR code

## VII. CONCLUSION

Noting data in the register is a very laborious work as well as time-consuming. The Material Gate Pass allows you to manage all Incoming and Outgoing items from your office or company. Material Gate Pass is available for use and it is easy to use and manage. This system is defined as a gate pass security system. The system ensure securing company from outside visitors, contractors and the departments, company vehicle security, material, visitor scrap information, contractor. Material Gate Pass helps the organization as well as a visitor to manage the Gate Passes. Frequently Visitors have data entry in top level to make them easily entry and exit. The proposed system reduces the manual effort by doing the whole work automatically using

technologies like audacity, tensor flow and QR codes. It is an automated version for the existing gate pass system that simplifies the verification and at the same time improves security.

## REFERENCES

1. Adrian Rosebrock , An OpenCv barcode and QR code scanner with Zbar , on May 21 2018, in Image Processing, Raspberry Pi , Tutorials, available from URL: <https://www.pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/>
2. Akhil Goel, Convert text to speech in Python, in geeksforgeeks , available from URL: <https://www.geeksforgeeks.org/convert-text-speech-python/>
3. Deepak Srivatsav, Reading and generating QR codes in Python using QRtools, in geeksforgeeks, available from URL: <https://www.geeksforgeeks.org/reading-generating-qr-codes-python-using-qrtools/>
4. Pulkit Sharma, A step-by-step introduction to the Basic Object Detection algorithms(part-1), Oct 11 2018, available from URL: <https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the-basic-object-detection-algorithms-part-1/>
5. Dan, Marian Macchi, Karen Wallace, Expert name pronunciation, Innovative speech technology, text-to-speech, available from URL: <https://www.espeech.com/>
6. Hao Gao, Faster R-CNN Explained, Sep 17 2017, available from URL: <https://medium.com/@smallfishbigsea/faster-r-cnn-explained-864d4fb7e3f8>
- Serdar Yegulalp, What is Tensor Flow? The machine learning library explained, June 18 2019 available from URL: <https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html>
7. NanoDano, Text-to-speech in Python with pyttsx3 , Oct 27 2018, in Dev-Dungeon Virtual Hacker space website, available from URL: <https://www.devdungeon.com/content/text-speech-python-pyttsx3>
8. Natesh M Bhat, pyttsx3-Text-to-speech x-platform, May 10 2019, available from URL: <https://pyttsx3.readthedocs.io/en/latest/>
9. David Jones, Pure Python Library for PNG image encoding/decoding, in github, available from URL: <https://github.com/drj11/pypng>

