Automatic Toll System Using DIP

Aiswarya R Krishnan[1], Emily Joy[2], Anjana MR[3], and Livya George[4]
[1,2,3,4] *Sahrdaya College of engineering and technology, Department of Computer Science and Engineering*

***Abstract*- Automatic toll collection aims at eliminating the delay at toll plazas by collecting toll without cash. In many developed countries the toll collection is implemented by using Radio Frequency ID (RFID).But using RFID is very costly. So image processing can be used as an alternative. In this paper, a camera will be used to take picture of the vehicle's license plate to sort the toll charge according to vehicles category. Using the captured picture, with the help of which a image processing technique will be implemented to extract the registration number of the car from the license plate, with the help of this extraction the details of the vehicles owner will be extracted from the database and the respective amount will be deducted, because that person has already paid the toll amount in advance for a respective duration In addition with multiple automatic toll taking booth there can be a manual booth with operator who will handle those vehicles which experience issues like not registered user or balance is empty. The system is more flexible and can be used at different toll plazas within a country by providing unique id for the toll plazas.**

Index Terms- RFID, Image processing, Pixels, Resolution, Automatic license plate recognition, License plate

I. INTRODUCTION

Due to the increasing number of vehicles, the traffic is increasing at the toll plazas. There are long queues on busy highways. Nowadays people prefer to use their personal vehicles instead of public transports which results in in-crease of traffic. Increasing number of vehicles on the roads, result into many problems such as congestion, air pollution and fuel wastage etc. Most of the toll plazas are operated manually, where there is an operator on each lane for collecting the toll amount i.e. for every lane there are two opera-tors one for operating the system and the other for interacting with the driver. So huge manpower is required. One more thing which results in congestion is the drivers sometimes start chitchatting with the toll operator, hence the vehicles waiting in the queue gets irritated and may result in chaos. At the present time transport or other vehicles are essential element of Traffic control structure at different toll centers.

The purpose of the project is collecting the toll according to vehicles and builds the real time application which recognizes vehicles licenses number plate at entry gate. Automatic toll collection is considered as one of the intelligent transport systems. It is aimed at making toll taxation more efficient, reliable, and safe and environment friendly. In the past, customer would have to wait at the toll booth to pay the collector, creating traffic congestion, pollution and of course of a lot of frustration. Today Automatic toll collection successfully re-moves unnecessary traffic delays; keep an eye on any car that might not be correctly registered. Automated toll collection is fast becoming a globally accepted of toll collection.

The vehicles number plate will be detected using a camera that will be placed at the toll checkpoint, with the help of which a image processing technique will be implemented to extract the registration number of the car from the number plate, with the help of this extraction the details of the vehicles owner will be extracted from the database and the respective amount will be deducted, even that information will be extracted from the database because that person has already paid the toll amount in advance for a respective duration.

Modules and Implementation. The block diagram of the proposed system consists of six different blocks.

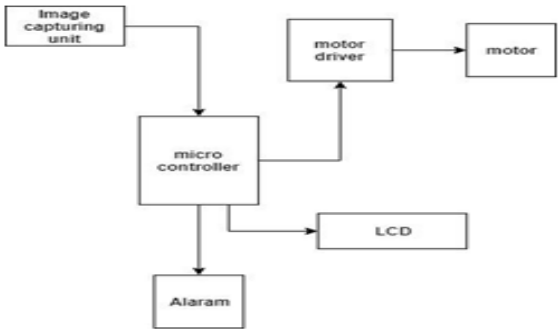

Fig. 1. Block diagram

- Image capturing unit: This unit captures the image of the license plate. The camera will be place at the toll gate for caspturing the license plate image. This image will be used for image processing and the characters in the license plate will be the output of the same.
- Microcontroller: The microcontroller used will be Arduino Mega 2560.e Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital in-put/output pins. The purpose of the microcontroller in the system is that the microcontrollers will checks the characters output from the image capturing unit with the database and validates the user.
- LCD: The unit is used to display the vehicle number and the balance in the user's account. The LCD will be a 16*4 LCD. This translates a display 16 characters per line in 4 lines. In this LCD each character is displayed in a 5×7 pixel matrix.
- Motor: The motor will be used to open and close the toll gate automatically. The DC motor will be con-trolled by the motor driver.
- Alarm: The alarm will be blown when an unregistered user enter the toll plaza .That is when captured image characters are compared with the database in the microcontroller and if the characters are invalid, then the alarm will be blown.

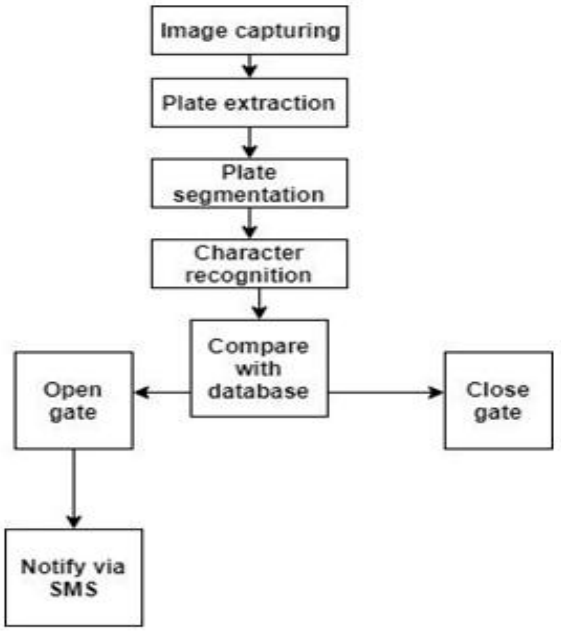

Fig. 2. Architecture diagram

ARCHITECTURE DIAGRAM

- Capture image: The first step is the capturing of an image using electronic devices such as optical (digital/video) camera; webcam etc can be used to capture the acquired images.
- Plate region extraction: Thee extraction of the number plate in an image is to find the row and column value of that image, then modify the image by using r/3:r. and save the image in another variable, assign the location and display the image, find the row and column for modified image.
- Plate segmentation: The output after extraction will be labeled and separate each character split the characters and finds the length of number plate for finding the correlation with the database.
- Character recognition: The OCR is now used to com-pare the each individual character against the complete alphanumeric database. The OCR actually uses correlation method to match individual character and finally the number is identified and stored in string format in a variable.

The characters output from the image processing unit will be compared with the database and if the values exist in the database, it means that it is a registered user and the gate will be opened then a notification will be send via SMS to indicate the balance. Otherwise, alarm will be blown.

ALGORITHM. Algorithm for Hardware section is given below. Initially set all the port directions for motor driver, LCD and sensors. Then initialize the serial communication. serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels. Only one channel is used in serial communication so it is cheap compared to parallel. Starting the gate will be closed, when a vehicle entered the image is captured and check whether it is valid. If yes gate is opened and again wait to get the sensor data to close the gate. If the data is not vali a alarm s generate. In the second algorithm initially we

```
Algorithm 1 Hardware section
0: Set port Direction
0: Initialize the serial communication
0: Close the gate
0: Wait for data
0: if Data = valid then
0:    open the gate
0:    Wait for sensor data
0:    close gate
0:    Goto step 3
0: elsegenerate aralam
0:    Stop
```

wait for the sensor data then we capture the image. Adjust the contrast and brightness of image. Then filter the image using Median filter. The median filter is a nonlinear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing. Detect the number plate. Now compare that number with data stored in the data base .Whether it is valid access the details and send data to the micro controller.

Image Acquisition: This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is al-ready in digital form. Generally, the image acquisition stage involves pre-processing, such as scaling etc.

Image Enhancement: Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness contrast etc.

```
Algorithm 2 Toll section
0: Start
0: Wait for sensor data
0: Capture the image
0: Adjust the contrast and brightness of the image
0: Filter noise
0: Detect the number plate
0: Segment the numbers and alphabets
0: if Datapresentindatabase then Access the details
0:    if Details = valid then Send to Micro controller
0:    elsegenerate aralam
0:       Stop
```

Image Restoration: Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Color Image Processing: Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Inter-net. This may include color modeling and processing in a digital domain etc. Wavelets and Multi-Resolution Processing: Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

Compression: Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data.

Proposed System. The proposed system provides a base for implementing automatic number plate detection using image processing for toll collection at toll checkpoints. This system will help in saving time as well as help in reducing congestion at toll checkpoints. This system will also help in monitoring any fraudulent behaviour that takes place at the toll checkpoints. The proposed system will capture an image placed at the toll checkpoint and will perform certain processes to detect the number plate of a vehicle.

Following are the steps that need to followed to detect a number plate:

- Image capturing
- Plate extraction
- Plate segmentation
- Character recognition

By using the characters we will check whether the vehicle is from kerala or other states and also checks whether it is valid vehicle or not by using the database which contain the data we are already entered during the process of data entry. If the vehicle is valid then the fee will taken from the account. If their is no cash in the account, it will make as fine. Then the gate will automatically open and also form a SMS in mobile to inform the balance amount .if the vehicle is invalid then form an alarm so that the police can know that it is a fraud driver. In this we also use LCD screen for forming expiry date of the account.

In this we use MATLAB as attached to SQL for making the database which contain the data of vehicles owners and also we enter the cash of each vehicles like car, bike etc separately.so that it can be used in the process of toll system. MATLAB (matrix laboratory) is a multiparadigm numerical computing

environment and proprietary programming language developed by Math Works. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C, Java, Fortran and Python. Although MAT-LAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional pack-age, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

By using the Arduino we do the process of micro controller. Arduino is an open source hardware and software company, project and user community that designs and manufactures single board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public Li-cense (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or bread-boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading pro-grams from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project. With the help of arduino we making the gate open automatically by using the motor and also helpful in the LCD.

We use Arduino ATMEGA 2560 as micro controller. The Arduino Mega 2560 is a microcontroller board based on the AT-mega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the micro-controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

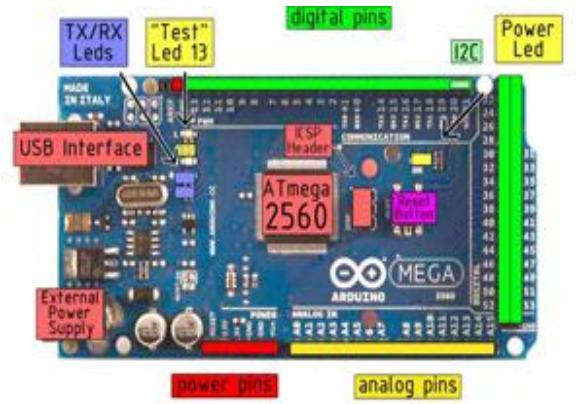

Fig. 3. Arduino ATMEGA 2560

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- VIN: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V: The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-

board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3: A 3.3 volt supply generated by the onboard regulator. Maximum current draw is 50 mA.
- GND: Ground pins

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the boot loader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18(TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 0 to 13. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although pro-vided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- I2 C: 20 (SDA) and 21 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I2 C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog Reference() function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analog Reference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being trans-mitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).A SoftwareSerial library allows for serial communication on any of the Mega's digital pins.The ATmega2560 also sup-ports I2C (TWI) and SPI communication. The Arduino soft-ware includes a Wire library to simplify use of the I2C bus.

For communication between the Arduino board and a computer or other devices we use serial communication. All Arduino boards have at least one serial port (also known as a UART or USART): Serial. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, we cannot also use pins 0 and 1 for digital input or output. You can use the Arduino environment's built-in serial monitor to communicate with an Arduino board. Click the serial monitor button in the toolbar and select the same baud rate used in the call to begin().

Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board). Don't connect these pins directly to an RS232 serial port; they operate at +/- 12V and can damage your Arduino board. The Arduino Mega has three additional serial ports: Serial1 on pins 19 (RX) and

18 (TX), Serial2 on pins 17 (RX) and 16 (TX), Serial3 on pins 15 (RX) and 14 (TX). To use these pins to communicate with your personal computer, it will need an additional USB-to-serial adaptor, as they are not connected to the Mega's USB-to-serial adaptor. To use them to communicate with an external TTL serial device, connect the TX pin to your device's RX pin, the RX to your device's TX pin, and the ground of your Mega to your device's ground.

The Arduino DUE has three additional 3.3V TTL serial ports: Serial1 on pins 19 (RX) and 18 (TX); Serial2 on pins 17 (RX) and 16 (TX), Serial3 on pins 15 (RX) and 14 (TX). Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip, which is connected to the USB debug port. Additionally, there is a native USB-serial port on the SAM3X chip, SerialUSB'.The Arduino Leonardo board uses Serial1 to communicate via TTL (5V) serial on pins 0 (RX) and 1 (TX). Serial is reserved for USB CDC communication. For more information, refer to the Leonardo getting started page and hardware. So that we use serial communication between the vehicle and toll server.

The Arduino Mega2560 can be programmed with the Arduino software. The Atmega2560 on the Arduino Mega comes preburned with a bootloader that allows to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header.

Rather then requiring a physical press of the reset button be-fore an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the AT-mega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot-loader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it com-municates waits a second after opening the connection and before sending this data. The Mega contains a trace that can be cut to disable the auto - reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". it may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Al-though most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is re-moved.

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by con-trolling lights, motors, and other actuators. The micro-controller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino

development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP). Arduino is a crossplatoform program.

A. Conclusion. The project proposes about the image processing technique to implement the automated toll collection in order to reduce congestion and fraudulent behaviour at the toll checkpoints. The automated deduction of the toll from the vehicle owner's bank account. The proposed system will help in reduce the human intervention at the toll collection areas. The purpose was to improvise the toll collection as well as implement a system for smart card users as smart cards is being used by many people in order to avoid physical cash. Automated Toll payment system is used to provide flexibility and reliability in paying toll amount at toll plaza

These system involves the use of high capacity cameras on toll plaza capable of capturing number plate of car. The image processing is performed on captured car number plate image. The image processing is used to detect the owner of car using car number plate. The amount can be paid by several ways such as registered user using wallet on Automated toll payment system website for users. The system for Image Processing Based Automatic Toll Booth in Indian Condition which is secure and highly reliable can be obtained. It can be used to remove all drawbacks with the current system such as time and human effort and it also doesn't require any tag only required best quality camera and fixed font number plate.

B. Future Enhancements. In the system toll payment is done automatically that is toll amount will be deducted from the user's account. The same idea can be used to improve car parking and security system.

In future perspective, the police department should be involved in this system to deduct various fine including chalan, alcoholic motorist, non-helmate wearer, no seat belt put, high speed motors etc.

Through the toll collection application we are making the toll collection possible from either mobile balance or bank ac-count associated with registered number of the vehicle. Features of the proposal include:

- User can get the detail of tax amount.
- User can pay the tax before actually passing through the toll booth.
- Irregularities of toll collection by the employees can be controlled.

The system can be extended to be used to determine the unauthorized vehicle detection system. Since the proposed Mosel uses image processing technique based on license plates, the vehicles which are unauthorized will get blocked at the toll gates. The system also has an alarm facility for the unregistrerd vehicles. This will help to prevent unauthorized vehicles being entering or passing throuhgh the toll plaza.

This system can be deviced to aid to traffic signal breaking detection system. This means that since the images of the li-cense plates are captured by the camera placed at the toll plaza, the vehicles those have violated the traffic signals can be blocked at the particular checkpoint. Or if a user registered with the system violates the traffic, the details of the vehicle owner can be easily fetched by using the license plate number.

Another future expansion for the system include the parking lot automation. In the present scenario traffic congestion due to the parking of vehicles are a matter of concern. The toll system can be modified so that for the registered users, it would automatically locates a parking lot via the use of GPS or other technologies.

BIBLIOGRAPHY

[1] Namrata Shirodkar, Preksha Uchil, (2015), "Number Plate Detection using Image Processing for Automated Toll Collection to prevent fraudulent behaviour ", International Journal of Advanced Research in Computer Engineering Technology (IJARCET) Volume 4 Issue 5.

[2] Omarhommadi1, Hussain Asiri, Mamdoohalzahrani, (2015) "Comparison of Different toll collections system's and RFID tool collection system", American Journal of Engineering Research (AJER) Volume-6, Issue-1, pp-118-121.

[3] Mosam Sangole, Yogesh Risodkar, Sampada Kulkarni, Rahul Kushare, Vijay Pawar, (2015) "Automatic Toll col-lection and Antitheft system".

[4] Satyasrikanth P, Mahaveer Penna, Dileep Reddy Bolla,(2007) "Automatic Toll Collection System using RFID ", International Journal of Computer Science and Mobile Computing , Vol. 5, Issue. 8.
[5] Persad, Khali and Walton, C. Michael, (2007), "Toll Col-lection Technology and Best Practices", Center for Transporatation Research, 2007. Project 0-5217.
[6] C. Singh, N. Bhatia and A. Kaur,(2008), "Hough transform based fast skew based Fast skew detection and accurate skew correction methods", Pattern Recognition, vol.41, pp.3528-3546.
[7] Rafiya Hossain, Moonmoon Ahmed, Md. Mozadded Al-fasani, Hasan U. Zaman,(2008), Advanced Security System Integrated With RFID Based Automated Toll Collection System", Third Asian Conference on Defence Technology.
[8] Nima Farajian, Morteza Rahimi B,(2014), "Algorithms for licenseplate detection :A survey ", First International Congress on Technology, Communication and Knowledge (ICTCK 2014) .
[9] Ling Hu and Qiang Ni,(2017), IOT-Driven Automated Object Detection Algorithm for Urban Surveillance System in Smart Cities", IEEE Joural.
[10] Siham Bouchelaghem and Mawloud Omar,(2018), Reliable and Secure Distributed Smart Road Pricing System for Smart Cities", IEEE transactions on intelligent transportation systems.
[11] http://www.ijerd.com/paper/vol13-issue11/Version-1/I131116871.pdf/
[12] http://www.fastag.org/.