MAEER's

BE Project Report

ON

ADVANCED OVERTAKE ASSISTANCE SYSTEM TO MAKE THE ROADS TRANSPARENT BY ESTABLISHING V2V COMMUNICATION

SUBMITTED BY,

Suyash Srivastava (B120023133)

Saumil Srivastava (B120023122)

Rohan Palkar (B120023089)

Project Guide:

Prof. Dr. B. S. Chaudhari (Internal Guide)

Sponsored by

College

Year: 2017-2018

Department of Electronics and Telecommunication

Maharashtra Institute of Technology, Pune - 38.

DECLARATION

We the undersigned, declare that the work carried under

Project Phase-II entitled

ADVANCED OVERTAKE ASSISTANCE SYSTEM TO MAKE THE ROADS TRANSPARENT BY ESTABLISHING V2V COMMUNICATION

Has been carried out by us and has not been implemented by any external agency/company that sells projects. We further declare that work submitted in the form of report is not been copied from any paper/thesis/site as it is. However existing methods/approaches from any paper/thesis/site are been cited and are been acknowledged in the reference section of this report.

We are aware that our failure to adhere the above, the Institute/University/Examiners can take strict action against us. In such a case, whatever action is taken, it would be binding on us.

Roll no	**Name of student**	**Signature with date**
405073	Suyash Srivastava	
405072	Saumil Srivastava	
405044	Rohan Palkar	

Academic Year **2017-18**

MAEER's

MAHARASHTRA INSTITUTE OF TECHNOLOGY, PUNE.

CERTIFICATE

This is to certify that the Project Phase-II entitled

ADVANCED OVERTAKE ASSISTANCE SYSTEM TO MAKE THE ROADS TRANSPARENT BY ESTABLISHING V2V COMMUNICATION

has been carried out successfully by

Suyash Srivastava (B120023133)

Saumil Srivastava (B120023122)

Rohan Palkar (B120023089)

during the Academic Year **2017-2018**
in partial fulfilment of their
course of study for Bachelor's Degree in
Electronics and Telecommunication as per the syllabus prescribed by the
SPPU.

Prof. Dr. B. S. Chaudhari
Internal Guide and Head of Department
(Electronics & Telecommunication)

ACKNOWLEDGEMENT

A project is an opportunity for the student to practically implement theoretical concepts. It proves to be a learning platform for the students so that they can compete successfully in their professional life. However, in this entire journey of completing the project, we needed proper guidance so as to avoid obvious mistakes.

We would like to thank our principal Prof. Dr. L. K. Kshirsagar for his constant encouragement. We would also like to thank our head of E&TC department and internal guide Prof. Dr. B. S Chaudhari for his invaluable guidance and support for making this project a success. We would also like to thank Prof. Dr. A. D. Anuse for his profound help.

Lastly, our sincere thanks to all the staff members and our friends Sarang Bachal, Kedar Remane, Vedant Salunke, Pushpak Rathi, Siddhant Sangai and Umang Wani who helped us in all possible ways to make this project a success.

ABSTRACT

Vehicle to Vehicle or V2V communication has recently been a hotbed for technological innovation in the Automotive Industry. V2V comprises of a network wherein in vehicles on roads are wirelessly connected to each other. There is real time transmission of data between vehicles in order to give drivers a richer and more vibrant driving experience. V2V forms the very basis of Intelligent Transport Assistance Systems (ITAS). These Assistance systems give the drivers a sixth sense, hence resulting in a safer drive.

Our project proposes to tackle the growing menace of hazardous overtaking on Indian Roads using the concepts of V2V and ITAS i.e. Advanced Overtake Assistance System. V2V wireless communication, computer vision and AI technologies form the core of the system.
It is a dual-frequency system which involves transmission of live video footage from the preceding vehicle to the following vehicle thereby making roads transparent. The goal of the project is to successfully establish a unique communication link with the vehicle in front at 2.4 GHz. The concept of Dynamic Number Plate Detection is used on the preceding vehicle to obtain a unique communication address for every Vehicle. Once the communication has been established, the video transmission takes place at 5.8 GHz.

The project has successfully been implemented on a pilot level on real cars. We hope that our innovation contributes towards reducing the loss of lives on Indian roads, and also serves as a stepping stone towards future V2V endeavours.

TABLE OF CONTENTS

List of Figures:

List of Tables:

List of Abbreviations:

1) V2V : Vehicle to Vehicle Communication
2) ADAS : Advanced Driver Assistance System
3) DSRC : Dedicated Short Range Communication
4) FPV : First Person View
5) ITS : Intelligent Transport System
6) VANETs : Vehicular Ad hoc Networks
7) CNN : Convolutional Neural Network
8) WAVE : Wireless Access in Vehicular Environment
9) RSSI : Received Signal Strength Indicator
10) VCF : Video Communication Frequency
11) USF : Universal Standard Frequency
12) ANPR : Automatic Number Plate Recognition
13) LCD : Liquid Crystal Display

1.1 BACKGROUND

Road accidents are one of the leading causes of life-loss in our country today. There are a number of factors responsible for casualties on our roads such as driving under influence, over speeding, but the most dangerous of them all is the manoeuvre of **overtaking**. According to a report by The Hindu, road-accidents claimed 1,30,000 lives in the year of 2014. The year 2015 showed the maximum increase in the number of road casualties in over a decade. And the numbers only increased in 2016. Numbers show that maximum crashes and deaths on roads took place in 2014 when drivers were overtaking. According to the same report by the Hindu on country's road accidents and deaths, over 48,000 people died in crashes caused due to overtaking and 'diverging' during 2014, which accounts for almost 40% of the total number of road-accident casualties that year. A more recent report by the Ministry of Road Transport and Highways states that 30,000 lives were lost due to overtaking alone in 2015, and further 26,000 were lost due to stationary vehicles and objects on the road which the driver could not see while overtaking. To sum it all up, a survey by the Brick's and Lane's revealed that an alarming every one in seven (14%) drivers in the past year have been compelled by another vehicle's rash overtaking to swerve, pull over, or break to avoid a collision. Thus, the numbers clearly show how much of a menace overtaking has become on Indian roads, and the dire need to curb this problem. Figure 1 shows the actual numbers involved and a graphical representation depicting exactly how big a menace overtaking is on Indian roads.

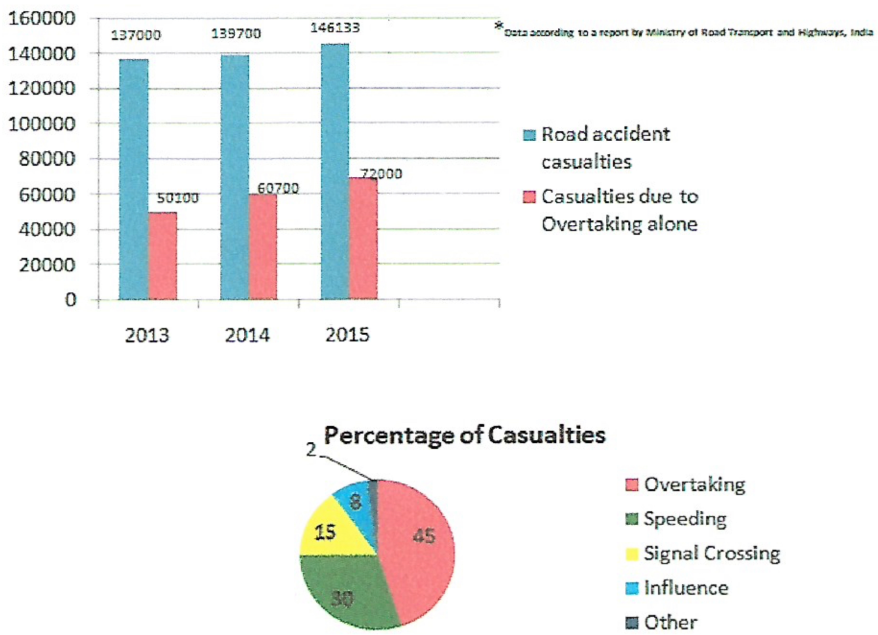


Fig. 1 Road accident casualties and contribution of overtaking in it

1.2 SCOPE OF THE PROJECT

The advent of vehicle-to-vehicle (V2V) communication has opened the opportunity to design driver-assistance systems that collect information from sensors residing in neighbouring vehicles. We have designed an advance driver assistance system (ADAS) which will help the driver during the critical overtaking manoeuvre by transmitting the footage of preceding car to his car. This will help the driver to take an informed decision about whether he should overtake or not. Dedicated Short Range Communication aka DSRC modules proposed for V2V communication are still under development and some of them which are developed are very costly and hence, we are planning to design our system by making use of the existing infrastructure. For video transmission, we have used First Person View (FPV) Transmitters and Receivers which are widely used in drone applications. The core idea of our system is to successfully establish a dedicated communication link between two cars. Once this communication has been established, it opens up a pathway into V2V. We sincerely hope that our proposed system will help to reduce the accidents on the road and will be beneficial to the society.

1.3 ORGANIZATION OF THE REPORT

- Chapter 1: Introduction

  This chapter provides an overview of the basic functionality of the system.

- Chapter 2: Literature Survey

  This chapter enlightens the literature survey of the work done in this field so far as well as the present technology in use.

- Chapter 3: System Development

  Explains in detail the design and development process of the system. Includes system requirements, flowchart and block diagram.

- Chapter 4: System Design

  It includes all the algorithms used for morphological operations, contour detection, character segmentation and recognition. It also includes hardware specifications.

- Chapter 5: Implementation of Design

  It includes sub-module algorithms used for licence plate localization, character segmentation, character recognition and flow of execution of transmitter and receiver sides.

- Chapter 6: Results and Conclusion

  It includes results of number plate recognition algorithm, comparison with existing technology, conclusion drawn from the results and the future scope of project.

2.1 LITERATURE SURVEY

When we decided to tackle the problem of road safety and overtaking, we had a look at several research papers in automated road safety systems to inspire our model. Our aim was to build a prototype which is dynamically useful, smart, and easy to interpret for the driver and real time, equipping the driver in situations where he needs more information to make a better informed decision.

Bourkerche et al. in [1] have mentioned the complexities involved in various applications of Intelligent transport System (ITS) and Vehicle to vehicle communication (V2V). The paper also focuses on the accuracy of location information of vehicle required by various applications. They have also proposed a data fusion system which can be used to improve location estimations in many sensor based systems.

Badr et al. in [2] have proposed a number plate recognition algorithm using morphological operations, histogram manipulation and edge detection techniques for plate localisation and character segmentation. It also introduced artificial neural networks for character classification and recognition.

Gomes et al. in [3] have proposed an idea of making the roads transparent using the video streaming between the two vehicles so that driver can take an informed decision about whether to overtake or not. They have also simulated their idea on a simulation platform which provides a suitable environment for validating such applications.

Olaverri-Monreal et al. in [4] proposed that the use of wireless technology based on Vehicular Ad hoc Networks (VANETs) for information exchange can influence the driver's behaviour towards improving driving performance and reduced road accidents. The information can be more relevant if it is there in the form of a video and can help the driver in challenging situations, such as overtaking a vision-obstructing vehicle.

Hijazi et al. in [6] have covered the basics of Convolutional Neural Network (CNN), the algorithms which can be used for character recognition and the procedure for training our CNN model adequately to get the best results.

We also looked into various Drones Live Streaming Technologies to study how we can effectively transmit live video footage over a certain distance in motion. We have been inspired by and adopted these concepts and used them to make our own similar model for car to car video transmission.

2.2 PRODUCT DEVELOPMENT SCENARIO

Road Safety concepts, namely Advanced Driver Assistance Systems or ADAS in conjunction with Vehicle to Vehicle Communication technologies like Dedicated Short Range Communication (DSRC) have been a hotbed for innovation for the past few years now. Many developed countries are aiming to establish V2V communication facilities in all new cars within the next five years and autonomous self-driving cars are expected to hit the roads within the next ten years or so. The DSRC modules mentioned above works on a standards specified in IEEE 802.11p, which is an approved amendment to 802.11 standard to add wireless access in vehicular environment (WAVE). It defines enhancements to 802.11 required to support the Intelligent Transport System (ITS) applications.

Leading automotive companies like Tesla and Ford have been actively working in this field for more than a decade and have come up with various concepts for V2V communication and autonomous cars.

NVIDIA has developed a dedicated hardware for vehicular automation and safety called DRIVEPX, which provides a platform and hardware support for implementing image processing and neural network applications. It is an AI platform that enables automakers, truck makers, tier 1 suppliers, and start-ups to accelerate production of automated and autonomous vehicles certified to International Safety Standards. The NVIDIA DRIVE platform combines deep learning, sensor fusion, and surround vision to change the driving experience. It is capable of understanding in real-time what's happening around the vehicle, precisely locating itself on an HD map, and planning a safe path forward. Tesla is planning to collaborate with NVIDIA and incorporate this hardware in their smart cars.

NXP Semiconductors have proposed a project in which they fly a drone over the car which sends real-time bird's eye view of the surroundings of the car, making the driver aware of what is around him. Figure 2 refers to this particular technology in practice. The ultimate aim of this technology is to provide a 360 degree view around the car to the driver. This project has been an inspiration for our pilot-level idea to make driver more aware of his surroundings via wireless real time video transmission.

Fig. 2 360 degree drone view for cars by NXP Semiconductors

Samsung has worked on the concept of making roads 'transparent' by displaying dash cam footage of the preceding vehicle on a large screen attached at the back, so that the following vehicle can see what is in front of the preceding vehicle. Samsung has majorly implemented this on large trucks, behind which the large LCD display screen can be attached. This project is called the Samsung Safety Trucks Project and is expected to hit the roads in the near future. Fig 3 shows the proposed implementation of the project. These, and many other technologies around the world, are either still under development, or proprietary.

Fig. 3 Samsung Transparent Trucks

We have taken inspiration from all these projects and tried to combine them into a single integrated system which makes roads 'transparent' and makes the overtaking manoeuvre safer, but which can also be implemented on today's cars itself, and is also cost efficient. This project will especially be useful on highways and ghats, where the overtaking manoeuvre is likely to be the most unsafe.

3.1 SYSTEM REQUIREMENTS

The main aim of this project is to develop a robust system which will achieve the V2V communication for exchange of data and video in real-time. The requirements of the system are:-

1) Software Requirements :-
   a) Opencv-Python (For image processing)
   b) TensorFlow-Python(For machine learning and neural network)
   c) Arduino Software
2) Hardware Requirements:-
   a) Eachine TS832 BoscamFPV 5.8G 48ch 600mW 7.4-16V Wireless AV Transmitter
   b) Eachine RC832 Boscam FPV 5.8G 48ch Wireless AV Receiver
   c) nRF24L01 single chip 2.4GHz Transceiver module
   d) Logitech C310 Webcam
   e) FPV monitor
   f) Arduino Mega 2560
   g) Push Buttons
   h) LM317 variable voltage regulator
   i) IC555 timer
   j) IC7476 Dual JK flip flop

3.2 SYSTEM FLOWCHART AND DESCRIPTION

Our solution assumes a classic overtaking situation where in a car B is following another preceding car A and wants to carry out an overtaking manoeuvre. Both cars are assumed to have a dash camera. When Car A is close enough and falls within the region of interest of Car B, Car B will perform Number plate Recognition on Car A. The number is used as a unique communication address for handshaking between the two vehicles. This address will be used to establish a V2V communication between the two cars, and the real time dash-cam footage of Car A will be transmitted and displayed on the dashboard of Car B, thus enabling the driver of Car B to see what is in front of Car A, and thus helping him make a better informed and safer decision. This is how we are making the roads 'transparent'.

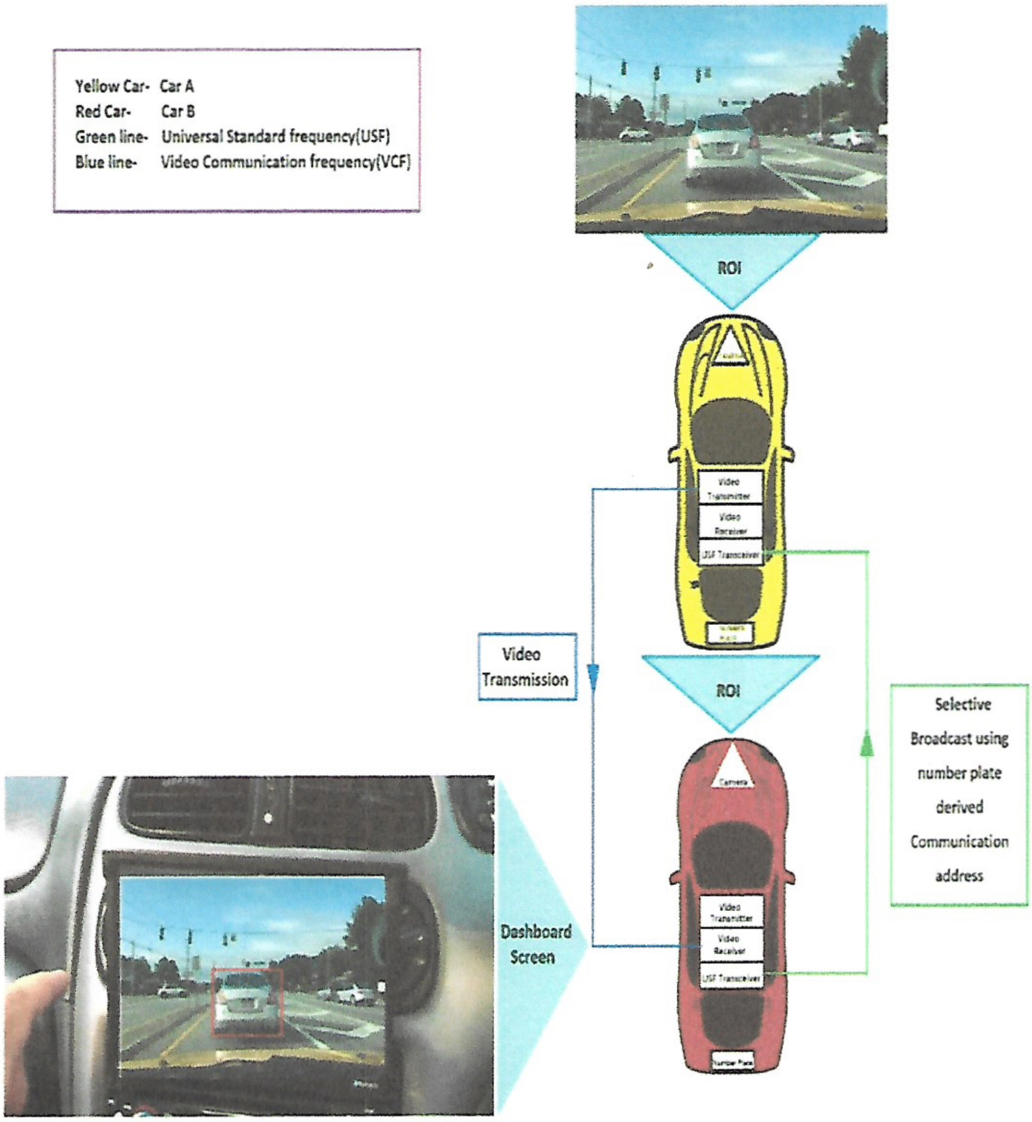


Fig.4 System schematic

In this dual frequency communication system, the first frequency will be a universal standard and common to all vehicles (used for handshaking between two vehicles) and the second frequency will be the video transmission frequency which will be decided by the two vehicles under consideration. The controller in Car B decides which frequency should be used for Video communication based on channel interference and accordingly tunes the video receiver in Car B for that frequency. Car B constructs a data frame which has the address (Number of Car A) and data payload (Value of Video Communication frequency). This data is selectively broadcasted by Car B at the universal frequency standard. The receiver in Car A is also tuned at the same frequency. The receiver of Car A then parses the data frame for address bits and data bits. The receiver only accepts data frames which bear its correct address and rejects the other data frames which it may receive from other surrounding vehicles. The data bits which have the value of the video communication frequency decided by Car B is used to tune the video transmitter in Car A. This ensures that the video communication frequency is unique for the two cars under consideration and only the video from Car A is transmitted to Car B. The

Universal standard frequency (USF) is 2.4 GHz frequency which will be used for data communication between the two vehicles while the Video Communication frequency will be 5.8 GHz. Figure 5 shows the algorithmic working of the system via a flowchart.

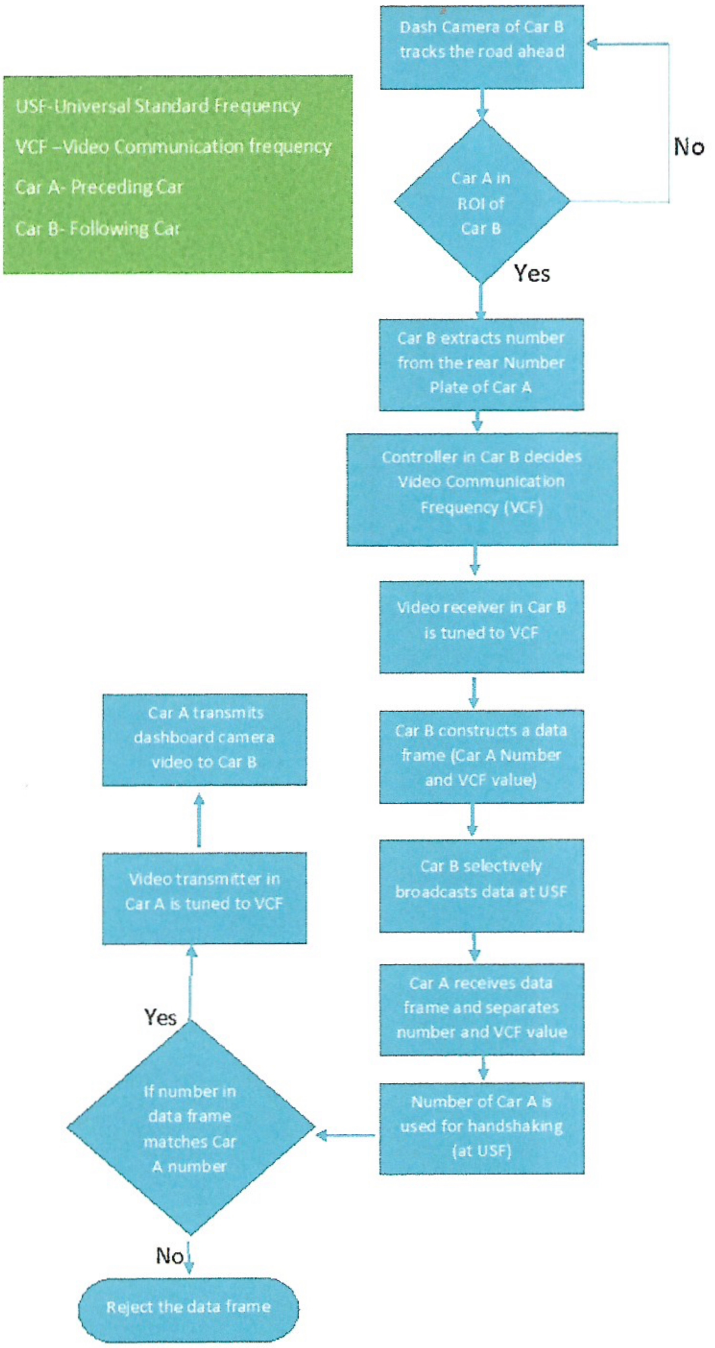


Fig. 5 System Flowchart

3.3 SYSTEM BLOCK DIAGRAM

The block diagrams of video transmitter side (car A) and video receiver side (Car B) are as shown below:

1) <u>Block Diagram of video receiver side (Car B):-</u>

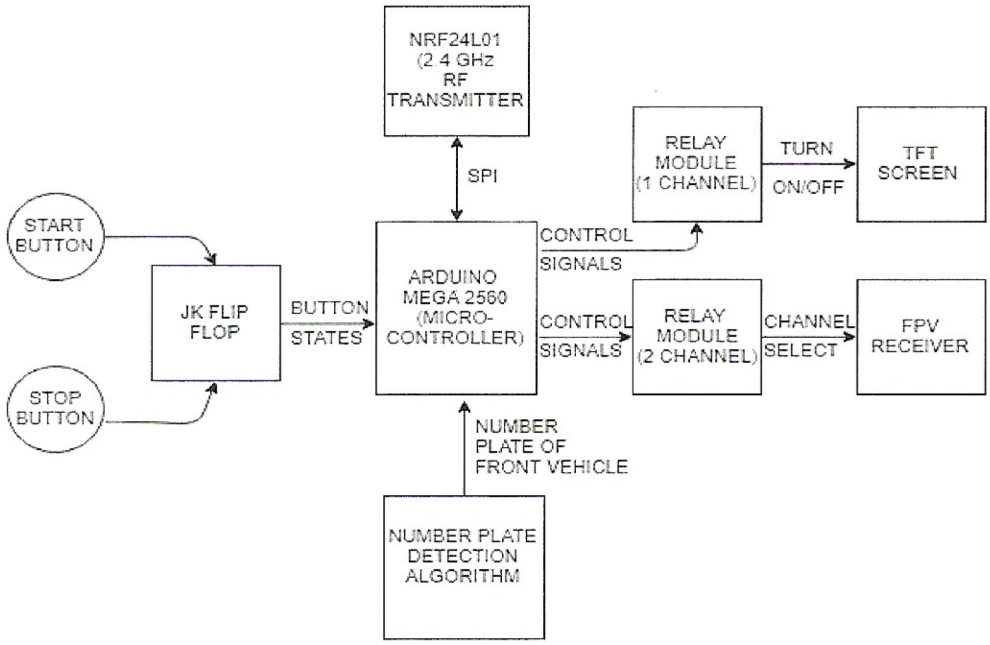


Fig. 6 Block diagram of video receiver side

The block diagram of the video receiver side (Car B) is as shown in fig 6. The camera continuously captures the video frames which are then analysed by the number plate detection algorithm to extract the vehicle number of the car in front. This extracted number is then passed to the controller. The controller then constructs a data packet which consists of extracted number as an address and a communication establishment request and ask nRF24L01 module to broadcast it. This module operates at 2.4 GHz which is our universal standard frequency (USF) as mentioned in system schematic. If the address present in the data packet matches with the address of the nRF module present in the car in front (Car A), nRF of car in front will send an acknowledgment. Now controller in car B will select a video communication channel of least interference out of available 48 cannel using Received Signal Strength Indicator (RSSI) pin of First Person View (FPV) Receiver module. FPV module is used for video communication at 5.8 GHz, which is video communication frequency (VCF). Controller then sends this channel number to nRF of car in front and after receiving the acknowledgment, it tunes FPV Receiver to that channel number using relays. Now when user press start button, he can see the video received by FPV receiver on TFT Screen.

The detailed description of each block is mentioned below:-

a) Controller Used :- **Arduino mega 2560**
   Need for controller: - To control the flow of execution of system
   - Takes input from ANPR algorithm in the form of a CNN encoded string and converts it into a 40-bit hex address required for NRF24L01 communication.
   - It provides an SPI interface to give the necessary commands to the NRF module. To electronically change the channel of the FPV Receiver to set it to the Video Frequency Channel Selected. It does so by giving the appropriate number of signals to the relay module.
   - Takes input from user interface (start/stop buttons) and takes necessary actions.

b) Universal Standard Frequency (USF) :- **nRF24L01 transceiver module**
   - nRF24L01 is a Radio Frequency (RF) transceiver module that operates at 2.4 GHz.
   - nRF transmitter module sends data packet that consists of address of receiver and data to send while nRF receiver checks the address field first when it receives the message and if address matches with its own address, then only it will accept the message else will reject it.
   - As working of this module is as per our requirement of USF device, we decided to select this module for our system.

c) Video Transmission Frequency(VF):-**First Person View (FPV) Receiver module**
   - FPV receiver module is most widely used in drone racing.
   - It consists of 48 frequency channels around 5.8 GHz. It also consists of RSSI pin that can help us to analyse the channels to find the channel of least interference.

d) Relay Modules :-
   - Relay module acts as an interface between the controller and FPV receiver. This is used to set FPV receiver to the desired video frequency channel. Also, it provides optical isolation between two connected devices.
   - We have also used a 5V 1 Channel Relay module to control the switch on and switch off of the TFT Screen.

e) Push Buttons :-
   - The output is high only for the duration that the switch is pushed for.
   - Start Button (Green) – To switch on the TFT Screen to display the transmitted video.
   - Stop Button (Red) – To switch off the TFT Screen and terminate the ongoing communication.

f) TFT Screen :-
   TFT screen is used to display the video received by FPV Receiver.

g) JK Flip Flop :- IC 7476

- JK Flip Flop is basically used to latch the state of push button inputs from the user. If the user presses a start button while the process of establishing communication is taking place, then it is necessary to store the state of the button so that it can be successfully read after the completion of the establishment of communication.
- The K input of 7476 is connected to ground and J is connected to button output. So when button output is high, the output of JK flip flop will be high (J=H, K=L, Q=H). Being a push button, the output remains high for a very short duration of time. After that, when J=L, K=L, it maintains the previous high input (Q=Q0). This indicates that the button was pressed.

h) Camera :- Logitech C310

We have used Logitech C310 HD webcam for capturing the number plate of the vehicle in front.

2) <u>Block Diagram of video transmitter side (Car A):</u>

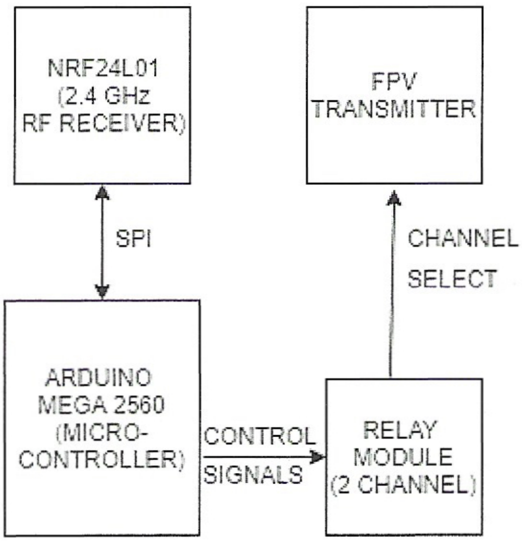


Fig. 7 Block diagram of video transmitter side

The block diagram of the video transmitter side (Car A) is as shown in fig. 7. It consists of a controller, nRF24L01 transceiver module, 2 Channel Relay module and FPV transmitter. The nRF module waits for other car's nRF module to send communication establishment request to it. As soon as it receives the request, it send acknowledgment to transmitter module. When transmitter nRF from other car sends the video frequency channel number, nRF sends that data to controller via SPI interface. The controller then tunes the FPV Transmitter to that channel using 2 channel relay module.

a) Controller Used :- Arduino mega 2560
   Need for controller: - To control the flow of execution of system
   - It provides an SPI interface to give the necessary commands to the NRF module. To electronically change the channel of the FPV Transmitter to set it to the Video Frequency Channel Selected. It does so by giving the appropriate number of signals to the relay module.

b) FPV Transmitter :-
   FPV Transmitter is a video transmitter module having 48 channels around 5.8 GHz. It transmits the camera footage to receiver module which is at same frequency channel as that of transmitter.

The complete project can be divided into following submodules.

1) ANPR:-

As mentioned is block diagram description, we are making use of number plate of preceding vehicle for addressing that vehicle and hence Automatic Number Plate Recognition (ANPR) becomes the most important module of our proposed system.

The typical stages in plate recognition algorithm includes:

a) Localizing the plate in an image

In this stage, we localize the number plate in a frame by making use of some image processing operations like grayscale conversion, morphological operations, filtering etc. At the output of this stage, we get only vehicle number and remaining part of image get masked.

b) Segmentation of plate characters

In this stage, we separate each character in previously localized image. At the output of this stage, we get each of the character in a number plate separately.

c) Character recognition

The characters separated in previous stage are recognised in this stage by making use of convolutional neural network (CNN).

2) ESTABLISHING COMMUNICATION: -

The licence plate detected in the previous stage is passed on from python code to Arduino with the help of PySerial package available, which encapsulates access for the serial port. Now, the Arduino at the data transmitter side(Car B) selects a video channel on which video communication shall take place and construct a frame which consists of address i.e. the number plate detected and video channel number as data and then transmits it to the car in front(Car A) on USF, which is 2.4GHz. The Nordic semiconductor's nRF24L01 transceiver module is used for this data communication. After receiving the acknowledgement from car A, the Arduino tunes the video receiver inside car B to the selected video communication channel with the help of relay module.

The Arduino inside car A receives the data frame transmitted by car B and parse it to get the video channel number. It then tunes the video transmitter to that channel with the help of relay module and video communication begins. The video transmitted can be seen on FPV monitor present inside car B which will assists the driver for the manoeuvre of overtaking.

4.1 AUTOMATIC NUMBER PLATE RECOGNITION (ANPR)

As described above, the typical ANPR algorithm can be divided into three stages which are localizing the number plate in an image, segment each character of the number plate and recognise the characters segmented.

4.1.1 Plate Localization

Each frame form the captured video is given as an input to this stage and the output of this stage is basically a region or contour containing the number plate with remaining area getting masked. Various image processing operations performed on the image are as follows:-

1) Converting camera image to grayscale:-

   The colour image is first converted to grayscale image with the help of convert colour function of opencv. The grayscale images basically reduces the complexity and increases the speed of computation. Figure 8 basically shows the greyscale converted video frame.

Fig. 8 grayscale image

2) Apply Top Hat operation to grayscale image

   The morphological operation Top Hat is the performed on the image. It returns an image containing objects or elements that are smaller than the structuring element and brighter than their surroundings. Since Indian license plates mostly have white background, this can help to localize the number plate well. This can be observed from figure 9 which shows the original image and its top hat equivalent.

   The top hat operation is defined as the difference between the original grayscale input image and its opening by some structuring element. Mathematically, it is defined as:

   $T_{hat}(f) = f - (f \, 0 \, b)$

   Where, f = input grayscale image

   0 = morphological opening

Fig. 9 top hat of greyscale image

3) Applying sobel operator:-

The Sobel operator, sometimes called the Sobel–Feldman operator or Sobel filter, is used in image processing and computer vision, particularly within edge detection algorithms where it creates an image emphasising edges. Edge detection is an image processing technique for finding the boundaries of objects within images. It works by detecting discontinuities in brightness. Edge detection is widely used in image segmentation operations. The edge detection can be done by calculating the gradient of an image in horizontal and vertical direction because edge is nothing but the points where there is change in brightness or pixel intensities.

There are various edge detection operators available like Roberts, Prewitt and Sobel operator. The sobel operators are most widely used because of the fact that they have better noise-suppression characteristics and noise suppression is an important issue while dealing with derivatives.

Here sobel operator is applied in horizontal direction to find the edges in that direction. Opencv has built in function called Sobel which gives us the output image with only edges.

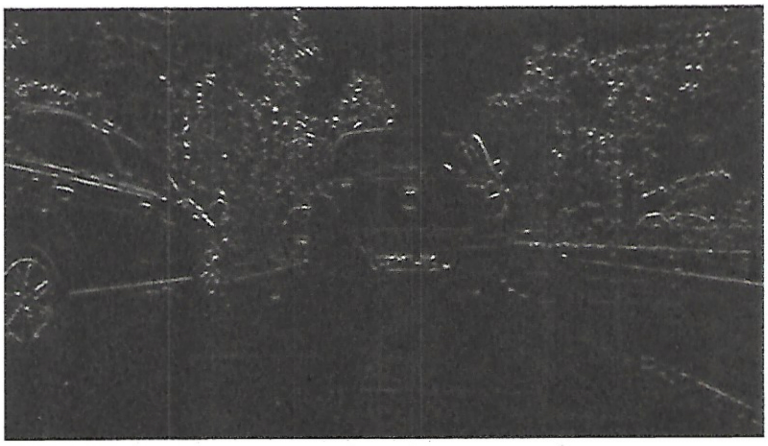


Fig. 10 sobel edge detection

4) Applying Gaussian Blur:-

In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise and reduce detail. Gaussian smoothing is commonly used with edge detection. Most edge-detection algorithms are sensitive to noise. Using a Gaussian Blur filter with edge detection aims to reduce the level of noise in the image, which improves the result of the following edge-detection algorithm. With more smoothening, fewer edges are detected. A Gaussian blur effect is typically generated by convolving an image with a kernel of Gaussian values.

In Opencv python, it is done with the function called cv2.GaussianBlur(). We should specify the width and height of kernel which should be positive and odd. We also should specify the standard deviation in X and Y direction, sigmaX and sigmaY respectively. If only sigmaX is specified, sigmaY is taken as same as sigmaX. If both are given as zeros, they are calculated from kernel size. Gaussian blurring is highly effective in removing Gaussian noise from the image. In the Gaussian kernel, the centre element has highest value and value goes on decreasing as the distance from the centre increases.

Fig.11 Gaussian blurred image

5) Closing:-

Closing is a morphological operation that is combination of erosion and dilation. The closing tends to smooth sections of the contours and fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in the contour. Closing is basically dilation followed by erosion by same structuring element. It can be seen in figure 12.

The closing of set A by structuring element B, denoted by A.B is defined as,

A.B = (A + B) – B = C - B

Where, A + B = C = dilation of A by B

C – B = erosion of previously dilated image by same structuring element B

In Opencv python, it is done with the function called cv2.morphologyEx(img,cv2.MORPH_CLOSE,kernel).

Here, kernel is nothing but the structuring element which is a matrix of odd dimensions.

Fig. 12 closed image

6) Thresholding:-

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images i.e. the image containing only two values required region i.e. the foreground contour and background gets masked. Figure 13 shows the original greyscale image and corresponding binary image with number plate contour detected.

In opencv python, various options are available for thresholding from simple binary to complex Otsu's thresholding. The grayscale image consists of pixel intensities from 0 to 255. In binary thresholding, the centre value of intensity i.e.127 is considered and pixels with intensity below that are made 0 and pixels with intensity more than 127 are made 255. This kind of approach is not well suited for our application and hence we have decided to make use of Otsu's thresholding which is an adaptive thresholding method where the centre pixel for comparison is calculated based on histogram of the image which gives the better results.

The function cv2.threshold () is used for this application. The parameters to this function are input image, lowest pixel intensity (generally 0), highest pixel intensity (generally 255), and which thresholding to be used. In our case the last parameter will be cv2.THRESH_BINARY + cv2.THRESH_OTSU.

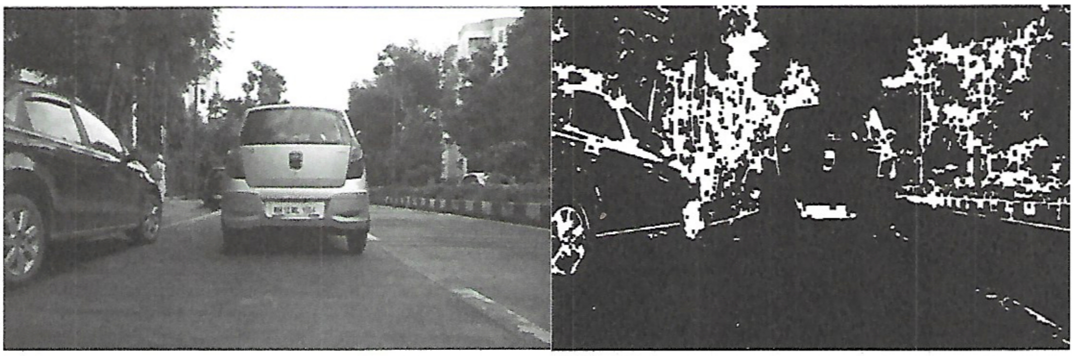


Fig. 13 grayscale and thresholded image

Figure 14 shows the number plate contour localised in the colour image which will be extracted in later stage for character segmentation and recognition.

Fig. 14 grayscale and processed image combined

7) <u>Localization :</u> -

The first step in localizing the number plate is to find contours in previously processed binary image (thresholded image).Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same colour or intensity. The contours are a useful tool for shape analysis and object detection and recognition. In Opencv, finding contours is like finding white object from black background. Opencv has a function called cv2.findContours(). It has got three arguments, the source image, contour retrieval mode, and contour approximation method and it outputs contours and hierarchy. Contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object.

Now, out of all contours, we need to select the contour having rectangular shape like the number plate. We ran a for loop over all the contours and with the help of cv2.boundingrect() function, we got the co-ordinates of top left corner of the contour as (x, y) and total width and height of the contour as (w, h).We also calculated the area of the contour with the help of cv2.contourArea() function. Now, we check for some h/w constraints with help us to extract the number plate. With the help of (x, y) and (w, h), we extracted the number plate rectangular part from the original image.
Some pre-processing is again done on the extracted image like adaptive histogram equalization and image is thresholded. Again, we find out the contours from thresholded image and contour having the maximum area is selected and extracted from the original grayscale image and passed on for character segmentation. The extracted image is as shown in the figure 15.

Fig. 15 extracted number plate

4.1.2 Character Segmentation

The input to this stage is localized and extracted grayscale number plate from the previous stage and output is all characters of the number plate being detected from left to right in a sequential order. This can also be done with the help of contour detection. We first converted the image into binary image by Otsu's thresholding and then invert the thresholded image because contours are basically white regions and characters in the number plate are mostly black. The detected number plate characters are then passed on to the convolutional neural network for recognition. Figure 16 shows the segmented characters of above extracted number plate which are then applied as an input to convolutional neural network for recognition.

Fig. 16 segmented characters of the number plate

4.1.3 Actual Implementation

The following are the outputs of ANPR algorithm when they have been implemented on a real-world video.

- Number Plate Localization: After performing the image processing steps as mentioned in the number plate localizing section above, we finally extract the number plate by bounding it with a box of the standard h/w ratio of the number plates.

Fig. 17 Number Plate Localization

- Having localized the number plate, each separate character is segmented out and stored in a serial order

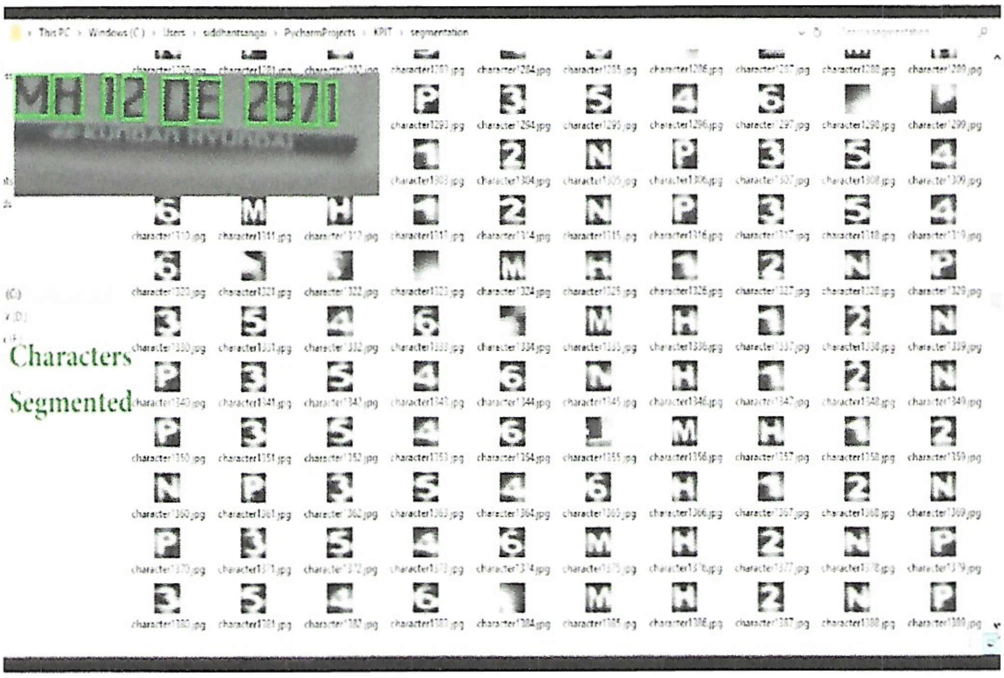


Fig. 18 Character segmentation and storage

4.1.4 Character Recognition: CONVOLUTIONAL NEURAL NETWORK

The individual segmented character image from the previous stage is applied as an input to the convolutional neural network and neural network predicts the character or digit present in the image.

In machine learning, a convolutional neural network (CNN) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analysing visual imagery. The major application of CNN is in image classification. The CNN inputs an image and pass it through a series of convolutional, nonlinear, pooling (down sampling), and fully connected layers, and get an output. The output can be a single class or a probability of classes that best describes the image. In our case, there will be total 36 classes, 10 for digits and 26 for alphabets. Our CNN will give us the prediction that in which class the input image belongs i.e.it will accurately identify the character present inside the segmented image. In the following sections, we have described the typical architecture of a convolutional neural network.

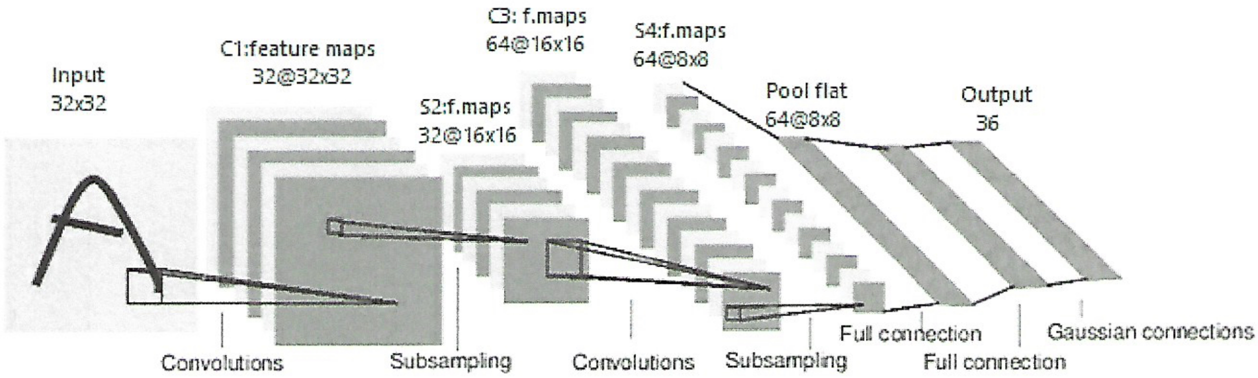


Fig. 19 A full convolutional neural network

The first layer in a convolutional neural network is always a convolutional layer. In this layer, the input image is convolved with a weight matrix. The weight matrix behaves like a filter in an image extracting particular information from the original image matrix. A weight combination might be extracting edges, while another one might a particular colour, while another one might just blur the unwanted noise. The weights are learnt such that the loss function is minimized. Therefore weights are learnt to extract features from the original image which help the network in correct prediction. When we have multiple convolutional layers, the initial layer extract more generic features, while as the network gets deeper, the features extracted by the weight matrices are more and more complex and more suited to the problem at hand.

The convolutional layers are generally followed by subsampling or pooling layers. Pooling is done for the sole purpose of reducing the spatial size of the image. Pooling is done independently on each depth dimension, therefore the depth of the image remains unchanged. The most common form of pooling layer generally applied is the max pooling. The max pooled image still retains the information but the dimensions of the image are reduced. This helps to reduce the parameters to a great extent.

After multiple layers of convolution and pooling, we would need the output in the form of a class. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. However, to generate the final output we need to apply a fully connected layer to generate an output equal to the number of classes we need. It becomes tough to reach that number with just the convolution layers. Convolution layers generate 3D activation maps while we just need the output as whether or not an image belongs to a particular class. The output layer has a loss function like categorical cross-entropy, to compute the error in prediction. Once the forward pass is complete the backpropagation begins to update the weight and biases for error and loss reduction. The complete working of a convolutional neural network is as follows:

- We pass an input image to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolution layer extract relevant features from the input image to pass further.
- Each filter shall give a different feature to aid the correct class prediction. In case we need to retain the size of the image, we use same padding (zero padding), otherwise valid padding is used since it helps to reduce the number of features.
- Pooling layers are then added to further reduce the number of parameters
- Several convolution and pooling layers are added before the prediction is made. Convolutional layer help in extracting features. As we go deeper in the network more specific features are extracted as compared to a shallow network where the features extracted are more generic.
- The output layer in a CNN as mentioned previously is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network.
- The output is then generated through the output layer and is compared to the output layer for error generation. A loss function is defined in the fully connected output layer to compute the mean square loss. The gradient of error is then calculated.
- The error is then back propagated to update the filter (weights) and bias values.
- One training cycle is completed in a single forward and backward pass.

Before implementation, the convolutional neural network must be trained and tested. During training, the images from the training dataset are applied as an input to the CNN. The output class for these images is known. The predicted output is then compared with the known output and error is calculated. The weights and biases of the network are then modified in such a way that the error should get minimized. After training, the testing dataset images are applied to the network for determining the accuracy of the CNN. Some of the parameters of the network are modified to improving the accuracy. This training and testing is a cyclic process and carried out till desired accuracy is achieved and then the neural network can be used in real world applications.

Implementation

INPUT:

An image processing algorithm is used to localize the number plate of a vehicle from an arbitrary image/video and segment the individual characters.

These individual character images are fed as input to the convolutional neural network.

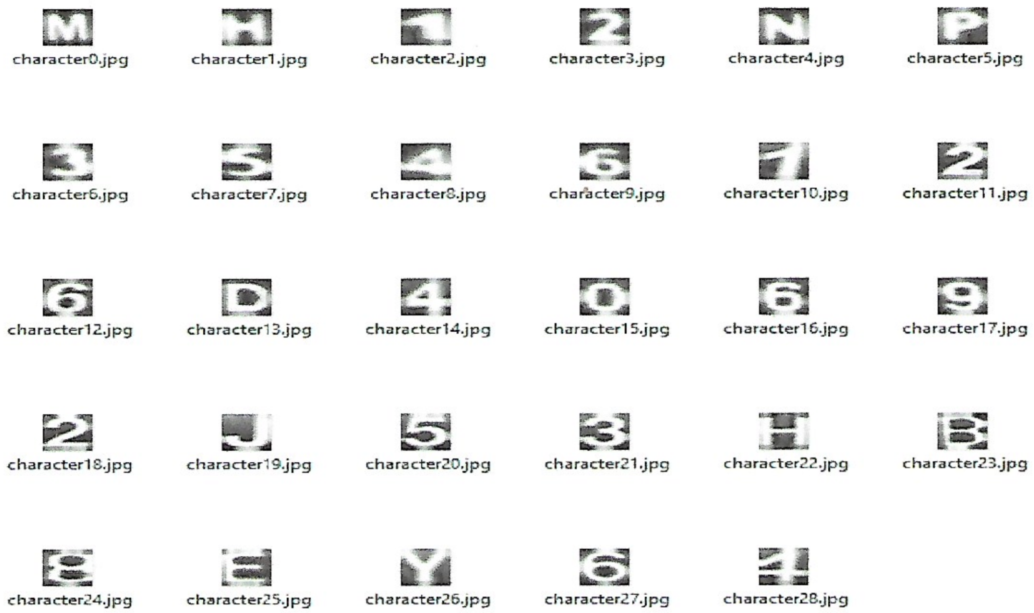


Fig. 20 CNN input

OUTPUT:

The convolutional neural network classifies the input images as one of the 36 output classes: 0 to 9 and A to Z

The output encoding scheme as follows:

0	0	I	18
1	1	J	19
2	2	K	20
3	3	L	21
4	4	M	22
5	5	N	23
6	6	O	24
7	7	P	25
8	8	Q	26
9	9	R	27
A	10	S	28
B	11	T	29
C	12	U	30
D	13	V	31
E	14	W	32
F	15	X	33
G	16	Y	34
H	17	Z	35

Table 1 CNN Encoding Scheme

```
C:\Python35\python.exe C:/Users/siddhantsangai/PycharmProjects/KPIT/SoftCOmputing.py
INFO:tensorflow:Using default config.
INFO:tensorflow:Using config: {'_save_checkpoints_steps': None, '_save_summary_steps': 100,
 '_save_checkpoints_secs': 600, '_model_dir': '/tmp/char_cnn_model_new7',
 '_keep_checkpoint_max': 5, '_log_step_count_steps': 100, '_keep_checkpoint_every_n_hours':
  10000, '_session_config': None, '_tf_random_seed': 1}
WARNING:tensorflow:Input graph does not contain a QueueRunner. That means predict yields
 forever. This is probably a mistake.
2018-04-06 21:21:02.403827: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\35
 \tensorflow\core\platform\cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled
 to use AVX instructions, but these are available on your machine and could speed up CPU
 computations.
2018-04-06 21:21:02.457345: W C:\tf_jenkins\home\workspace\rel-win\M\windows\PY\35
 \tensorflow\core\platform\cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled
 to use AVX2 instructions, but these are available on your machine and could speed up CPU
 computations.
INFO:tensorflow:Restoring parameters from /tmp/char_cnn_model_new7\model.ckpt-208610
New Samples, Class Predictions:    [22, 17, 18, 2, 23, 25, 19, 28, 4, 6, 1, 2, 6, 13, 4, 2,
 14, 9, 2, 21, 5, 3, 17, 11, 14, 14, 34, 6, 4]


Process finished with exit code 0
```

Fig. 21a CNN output

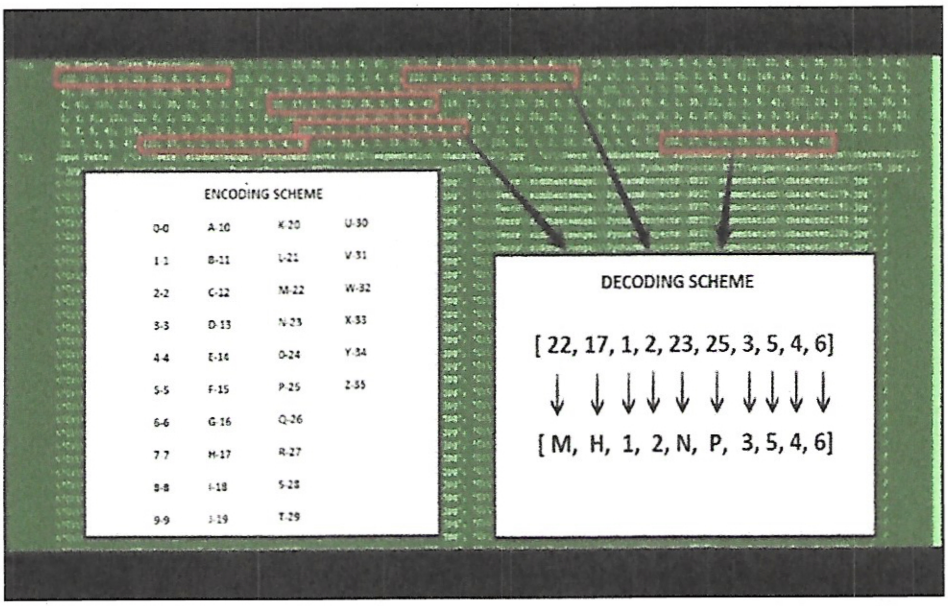


Fig. 21b CNN encoding and decoding scheme

TOOLS: TensorFlow library is used for implementing the convolutional neural network. TensorFlow 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. TensorFlow originated as an internal library that Google developers used to build models in-house, and we expect additional functionality to be added to the open source version as they are tested and vetted in the internal flavour. At a high level, TensorFlow is a Python library that allows users to express arbitrary computation as a graph of *data flows*. Nodes in this graph represent mathematical operations, whereas edges represent data that is communicated from one node to another. Data in TensorFlow are represented as tensors, which are multidimensional arrays. Although this framework for thinking about computation is valuable in many different fields, TensorFlow is primarily used for deep learning in practice and research.

TRAINING AND RESULTS:

We have currently trained the model on 1,36,000 images.

- One of the dataset used for training is the char74K dataset which includes around 74k natural scene images.
- We have also created our own database for training purpose which is nothing but the output of character segmentation stage. The images of number plates have been clicked manually from the natural scene.

ACCURACY: We have achieved an accuracy of approximately 90 percent.

- 700 segmented images were given as input to the CNN.
- 634 correct predictions were achieved.

4.2 ESTABLISHING COMMUNICATION

As mentioned above, our project uses a dual frequency system for establishing a unique communication link between the two cars.

Universal Standard frequency or USF (2.4 GHz): The detected number plate is used to establish a two way, acknowledgement based communication link with only the concerned car in front, even if there are a multiple cars in the vicinity.

Video Communication frequency or VSF (5.8 GHz): Once the communication has been established between the two cars, video transmission occurs at a mutually decided frequency in the 5.8 GHz band. In the following section, we have explained the detailed working of hardware that we have used in our system.

4.2.1 2.4 GHz RF Transceiver: The NRF24L01 module

- The nRF24L01 is a highly integrated, ultralow power (ULP) 2Mbps RF transceiver IC for the 2.4GHz ISM (Industrial, Scientific and Medical) band.
- It has peak RX/TX currents lower than 14mA, a sub µA power down mode, advanced power management, and a 1.9 to 3.6V supply range
- High battery life and low power consumption
- The nRF24L01 integrates a complete 2.4GHz RF transceiver, RF synthesizer, and baseband logic including the Enhanced ShockBurst™ hardware protocol accelerator supporting a high-speed SPI interface for the application controller.

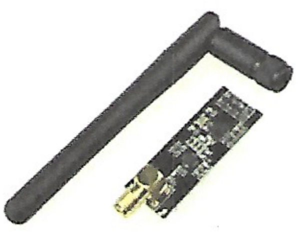


Fig. 22 nRF24L01 module

Piping in Nrf24L01:

Piping is the basis of communication in NRF24L01. NRF24L01 has 126 channels. And the radio communication can take place on any of these channels. But **piping** is an address based protocol that allows the NRF transmitter to communicate with multiple receivers on the same RF channel. A transmitter can be connected to 6 receivers at a time. Therefore, each pipe is essentially just an address that you can receive data on (when in RX mode). This means that any 24L01 can actually act as 6 different receivers at the same time. All pipes use the same RF channel.

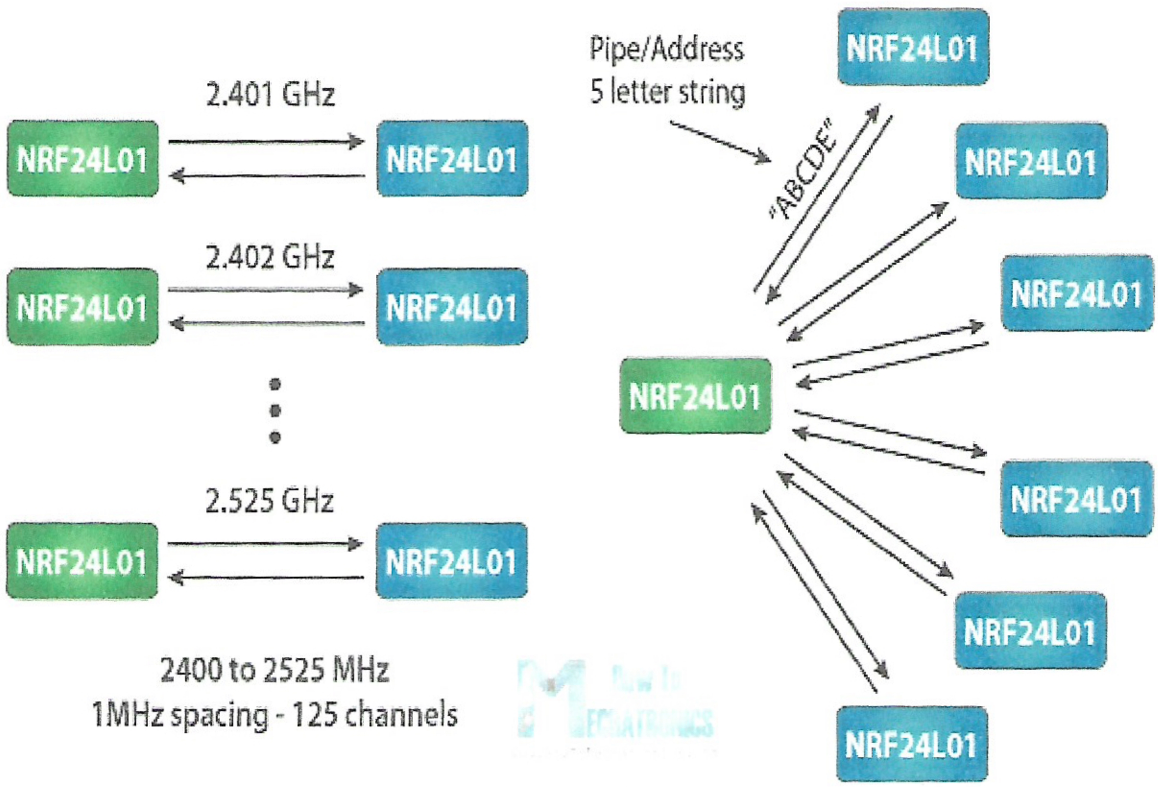


Fig. 23 nrF24L01 pipe concept

How we are using Piping in NRF24L01:

- In our system, each car can act as either a transmitter or receiver. If car B is trying to overtake Car A, it will read the number plate of Car A and obtain a 40 bit encoded address at its transmitter side.
- Car A, or every car, will have the 40 bit encoded address based on their own number plate as their receiver address.
- So when Car B reads number plate of Car A, their transmitter and receiver addresses will match, and a communication will be established between the two cars.

This is how piping is used to establish a unique communication link with the required car:

How to establish communication with only the car in front of us?

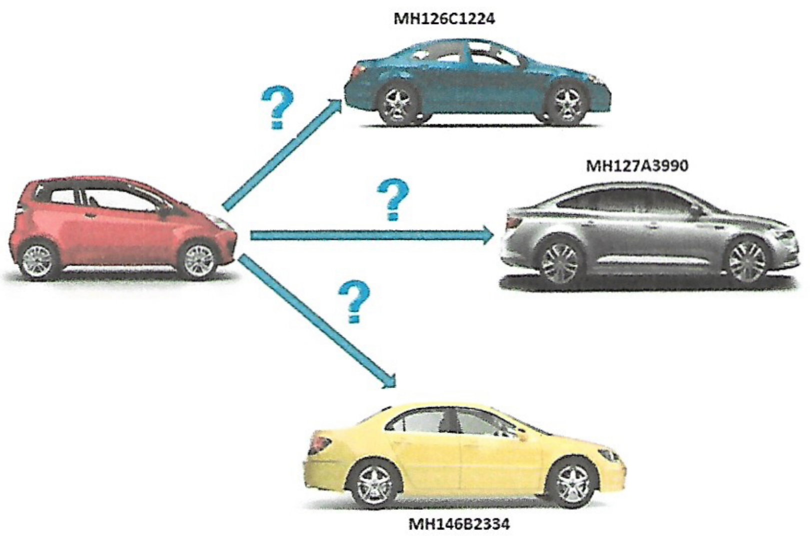


Fig. 24 How to identify correct car

When there are multiple cars in front of our car, it is important to establish a dedicated communication link at 2.4 GHz with the car in front of us only. So how do we identify the car in front of us?

Fig. 25 Number Plate Detection

In the field of view of camera of our car, only the car in front of us is completely visible. So the number plate of that car is detected. Hence we have the unique communication address of the car in front of us.

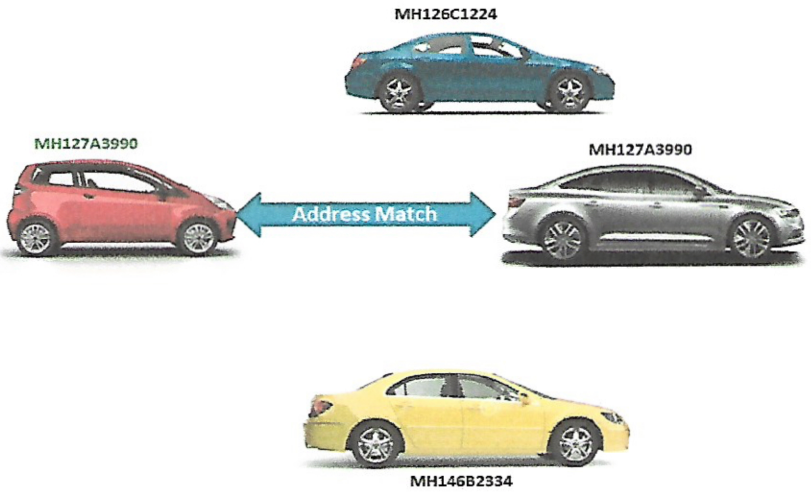


Fig. 26 Address Match

Once we have the address of the car in front of us, now the address at both the nRF transmitter in our car as well as the nRF receiver in the preceding car is same. Hence there is an address match, and a pipe will be formed between the designated nRf Tx-Rx pair.

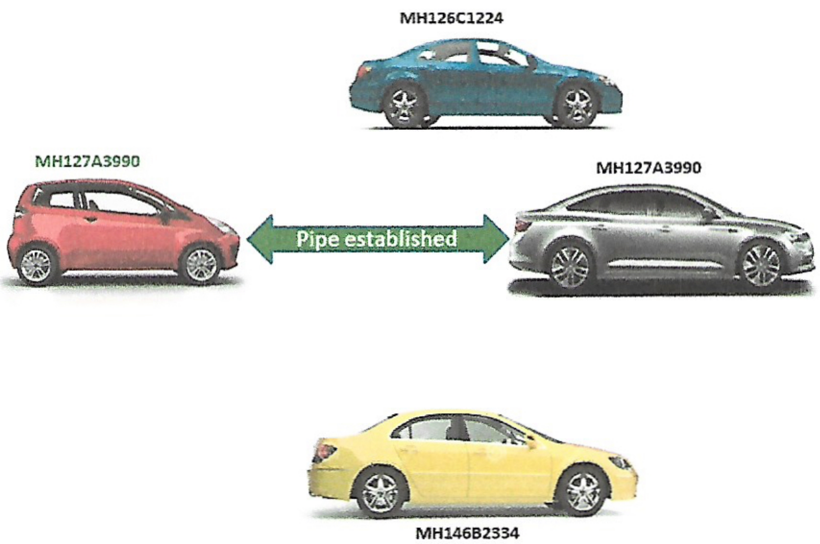


Fig. 27 Communication Established

Once the pipe has been established, it means that a dedicated communication link has successfully been established between the two cars at 2.4 GHz. The V2V communication has finally been completed. Now, the video transmission part can take place.

4.2.2 Video Frequency: 5.8 GHz

Once the communication has been established between the two cars at 2.4 GHz, the preceding car or the transmitter node will select a Video transmission frequency i.e. an empty channel in the 5.8 GHz band and communicate it to the receiver node or the following car. The 5.8 GHz transceivers of both the cars will then be mechanically tuned through hardware to the selected frequency, and then the video transmission will take place at this frequency for as long as the connection is maintained

The module which we used for transmitting video at 5.8 GHz is the FPV module.

The FPV module:

FPV stands for First Person View. FPV modules are 5.8 GHz transceivers which are widely used in drones and quadcopters to provide real time first person footage for various applications. We will be using this module for the video transmission of live dash-cam footage between the cars. Fig 28 specifies the active part of the FPV transceiver (Transmitter and Receiver) in the preceding and the following car.

The module that we have used are **Eachine TS832 (Transmitter) and Eachine RC832 (Receiver):**

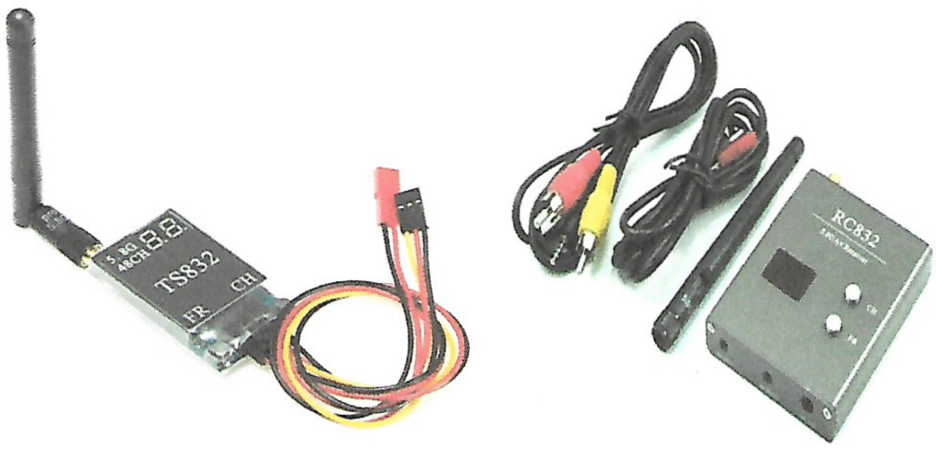


Fig. 28 FPV Tx and Rx

Features of Eachine TS832 and RC832:

1)48 channels choice to get the best transmitting quality
2)Compatible with A,B,E and F frequency bands
3)Smaller size with lighter weight

4)48 CH compatible to all FPV 5.8g transmitter.
5) Super small 200mA current for 600mW wireless transmitter power.
6)600mW transmitter power assure 5KM distance in open area, 5-8 KM is available if work with the bigger gain antenna.

The process of video transmission:

- When the communication has been successfully established between the cars at 2.4 GHz, it is now time for the actual video transmission. For the transmission of video, it is necessary that the FPV Transmitter in the preceding car and the FPV Receiver in the following car are tuned to the same radio frequency channel.
- Once the communication has been established at 2.4 GHz, the following car scans the 5.8 GHz spectrum and chooses one of the available channels. It then forms another data packet and sends this packet to the preceding car. This data packet contains the transmission frequency. This car then tunes its FPV receiver to the selected frequency channel
- When the preceding car receives the data packet containing the transmission frequency, it parses it, and then tunes its FPV Transmitter to the received frequency. It then sends an acknowledgement to the falling car
- The FPV transmitter in the preceding car and the FPV receiver in the following car have now been tuned to the same channel. Now video transmission takes place from the preceding car to the following car.

Adjusting the video communication channel:

1) The transmitter and receiver FPV modules have push button switches for changing the video communication channel. But in our system, we need to change or set the video communication channel with the help of microcontroller.
2) For doing so, we have used 5V relay modules and we are changing the channels by applying PWM pulses to relay module.
3) The relay module that we have used is 5V 2-channel relay module which is optically isolated to protect the devices.
4) We have written the code in such a way that it remembers the current channel number and when controller wants to jump to new channel, the pulses equal to the difference between the two channel numbers are applied to relay module.

4.2.3 THE FOUR WAY HANDSHAKE

The whole communication process between the two cars, starting from establishing of communication using the number plate to transmission of video, can be seen as a **four-step protocol**, or a **four-way handshake**, with the following steps:

Let Car A be the preceding car and Car B be the following car which is trying to overtake Car A.

- Car B detects the number plate of Car A. It forms a data packet containing the detected address and broadcasts it in the immediate vicinity at Universal Frequency 2.4 GHz
- Every car in the vicinity receives and parses the data packet. If the address in the data packet matches its own address, it accepts the frame and sends an acknowledgement. If not, it ignores it. Hence if Car B detects the number plate of Car A and broadcasts it, Car A will accept the data packet and send an acknowledgement back to Car B, saying that it has acknowledged the efforts of Car B to communicate with it.

Now, Car B has successfully identified Car A, and a dedicated communication has been established between the two cars at 2.4 GHz

- Now, Car B scans the 5.8 GHz spectrum and selects an available channel for video transmission. It then transmits this selected Video Frequency channel to Car A in another data frame.
- Once Car A has the transmission Video Frequency, it transmits live video to Car B at the designated channel.

Figure 29 summarizes the four-way handshake:

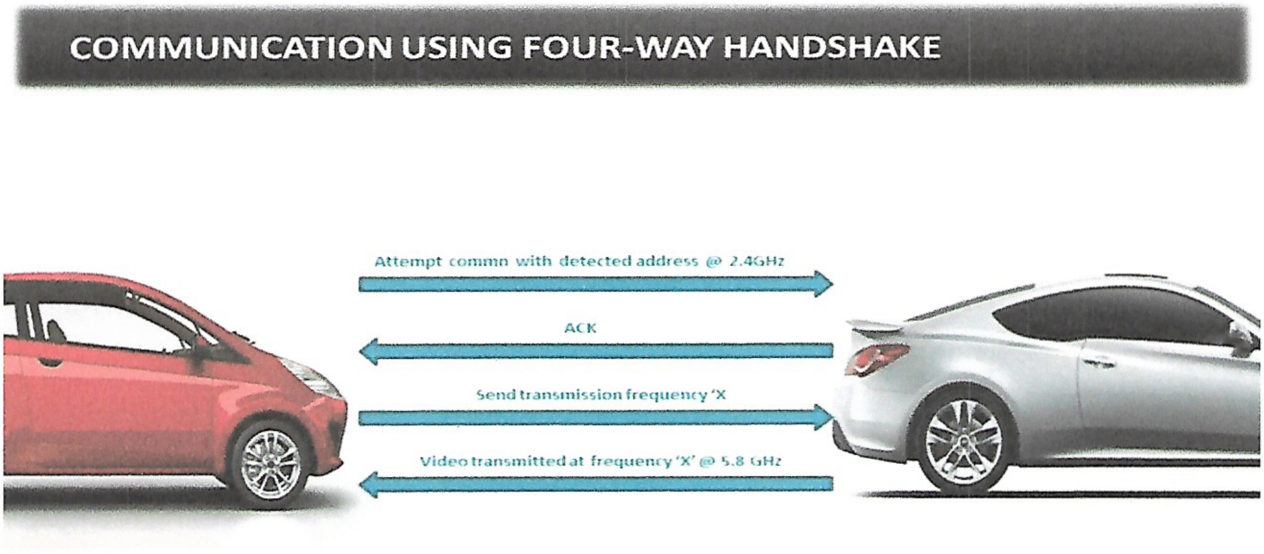


Fig. 29 the Four Way Handshake

4.2.4 RSSI: SELECTING THE VIDEO TRANSMISSION CHANNEL WITH LEAST INTERFERENCE

In telecommunications, received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal.

RSSI is usually invisible to a user of a receiving device. However, because signal strength can vary greatly and affect functionality in wireless networking, IEEE 802.11 devices often make the measurement available to users. In an IEEE 802.11 system, RSSI is the relative received signal strength in a wireless environment, in arbitrary units. RSSI is an indication of the power level being received by the receive radio after the antenna and possible cable or atmospheric loss. Therefore, the higher the RSSI number, the stronger the signal.

<u>Need For RSSI in our project</u>: In a real world scenario, where multiple cars will be equipped with our system, there is a possibility of interference during video transmission. Suppose a pair of cars, Car A and Car B are transmitting video at say channel number 11 of FPV transceiver. Now if another Car D adjacent or in vicinity of Cars A and B, wants to view the video of Car C in front of it, it will need to select a Video Frequency channel for transmission. If by chance, Car D also decides the channel for transmission as channel number 11, then both pairs of cars in vicinity of each other will be transmitting videos at the same channel/frequency. This can possibly lead to interference or crosstalk.

<u>RSSI Implementation</u>: The FPV Receiver module provides a RSSI pin. This RSSI pin gives output in the form of voltage in Volts. Before the selection of Video Frequency Channel for every connection of cars, we traverse through all the available channels in real time, and read the corresponding voltage level on the RSSI pin. A high voltage indicates the presence of noise on the channel. And low voltage indicates a noise-free channel. So for every video transmission, we choose the channel with lowest voltage level/lowest RSSI reading. Hence, this is the channel with the least noise, which is then transmitted to the car in front in a data frame, and then the video transmission take place.

Figure 30 shows the RSSI pin Output values in volts for all the channels traversed for a particular establishment of communication. Out of all these channels, the channel with the least voltage value is selected for video transmission as it is the channel with the least noise at that instance of time.

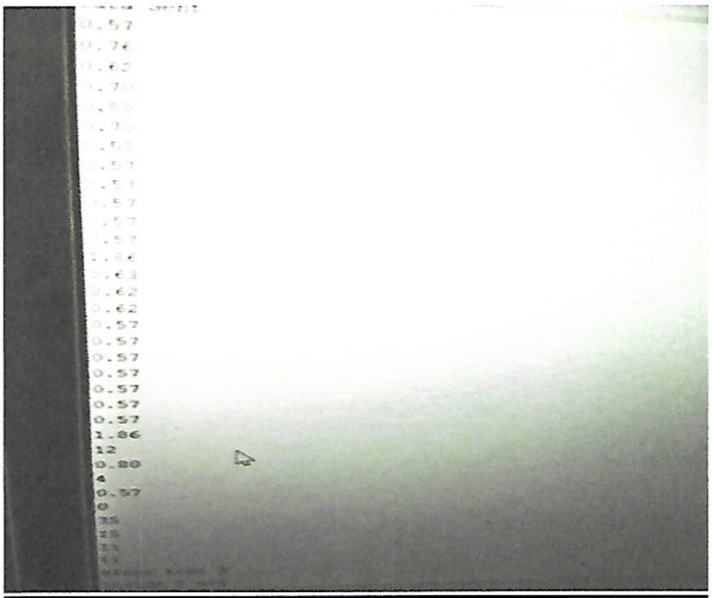


Fig.30 RSSI Implementation (values represents the voltage measured by RSSI pin of RC832 module for each channel)

4.3 CIRCUIT DESIGN

4.3.1 IC555 AS 50% DUTY CYCLE ASTABLE OSCILLATOR

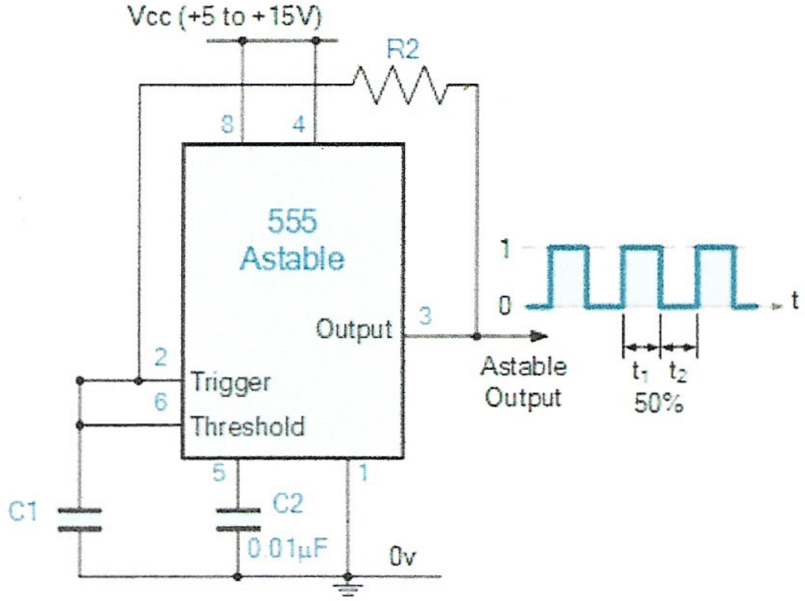


Fig. 31 IC555 Oscillator

Working: The 555 oscillator circuit shown in fig.31 produces a 50% duty cycle as the timing capacitor, C1 is now charging and discharging through the same resistor, R2 rather than discharging through the timer's discharge pin 7. When the output from the 555 oscillator is HIGH, the capacitor charges up through R2 and when the output is LOW, it discharges through R2. The equation for the 50% duty cycle astable oscillator is given as: -

$$f = \frac{1}{0.693(2R_2).C} \text{ Hz}$$

Component Selection: The output of IC555 oscillator is connected as a clock to the JK flip flop which is used to latch the high level logic state of the push button. Hence, the clock frequency should be chosen such that at least one rising edge of the clock should be there for the duration for which the push button is pressed. Here we have chosen clock frequency as 2 KHz to ensure that there as more than one rising edges of clock in the duration for which the button is pressed so as to change the output of flip flop.

Now, if we assume the value of capacitor C to be 10nF and substitute in the equation, the value of resistor R2 comes out to be 35.075 KΩ. Assuming the standard resistor value near this which is 33KΩ, we get a clock frequency of 2.186 KHz.

Hence, the components selected are:

R2 = 33KΩ and **C = 10nF** and resulting output frequency, **f = 2.186 KHz.**

4.3.2 LM317 AS ADJUSTABLE VOLTAGE REGULATOR

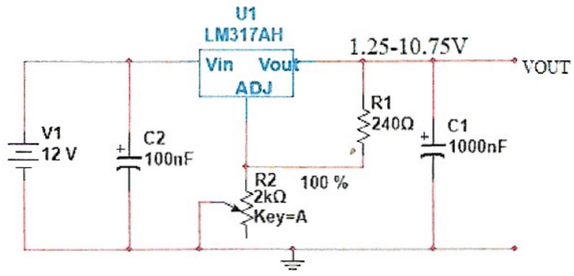


Fig. 32 LM317 as regulator

The JK flip flop requires 5V supply voltage and FPV receiver requires about 7.4V supply voltage which are drawn from the car battery which is of 12V. Hence, there is a need of voltage regulator. LM317 is an adjustable voltage regulator with maximum output current of 1.5A which is sufficient for both the devices. The output voltage of LM317 is given as:-

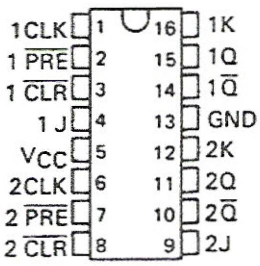
$$V_{OUT} = 1.25\left(1+\frac{R_2}{R_1}\right)$$

Hence, if we assume the value of R1 to be 240Ω and use a 2KΩ potentiometer in place of R2, we get an output voltage range from about 1.25V to a maximum output voltage of 10.75V.

Therefore, the components selected are:

R1=240Ω and **R2= 2KΩ pot.**

4.3.3 IC7476 DUAL INPUT JK FLIP FLOP



'LS76A
FUNCTION TABLE

INPUTS					OUTPUTS	
$\overline{PRE}$	$\overline{CLR}$	CLK	J	K	Q	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H†	H†
H	H	↓	L	L	Q_0	$\overline{Q}_0$
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	
H	H	H	X	X	Q_0	$\overline{Q}_0$

Fig. 33 IC7476 pin diagram and truth table

Function: JK Flip Flop is basically used to latch the state of push button inputs from the user. If the user presses a start button while the process of establishing communication is taking place, then it is necessary to store the state of the button so that it can be successfully read after the completion of the establishment of communication.

Implementation: The truth table of JK flip flop is as shown in the fig. 33. We have designed circuit in such a way that the K input of 7476 is connected to ground and J is connected to button output. So when button output is high, the output of JK flip flop will be high, as seen from the table (J=H, K=L). Being a push button, the output remains high for a very short duration of time. After that, when J=L, K=L, it maintains the previous high input. This indicates that the button was pressed.

4.3.4 CIRCUIT SCHEMATIC:

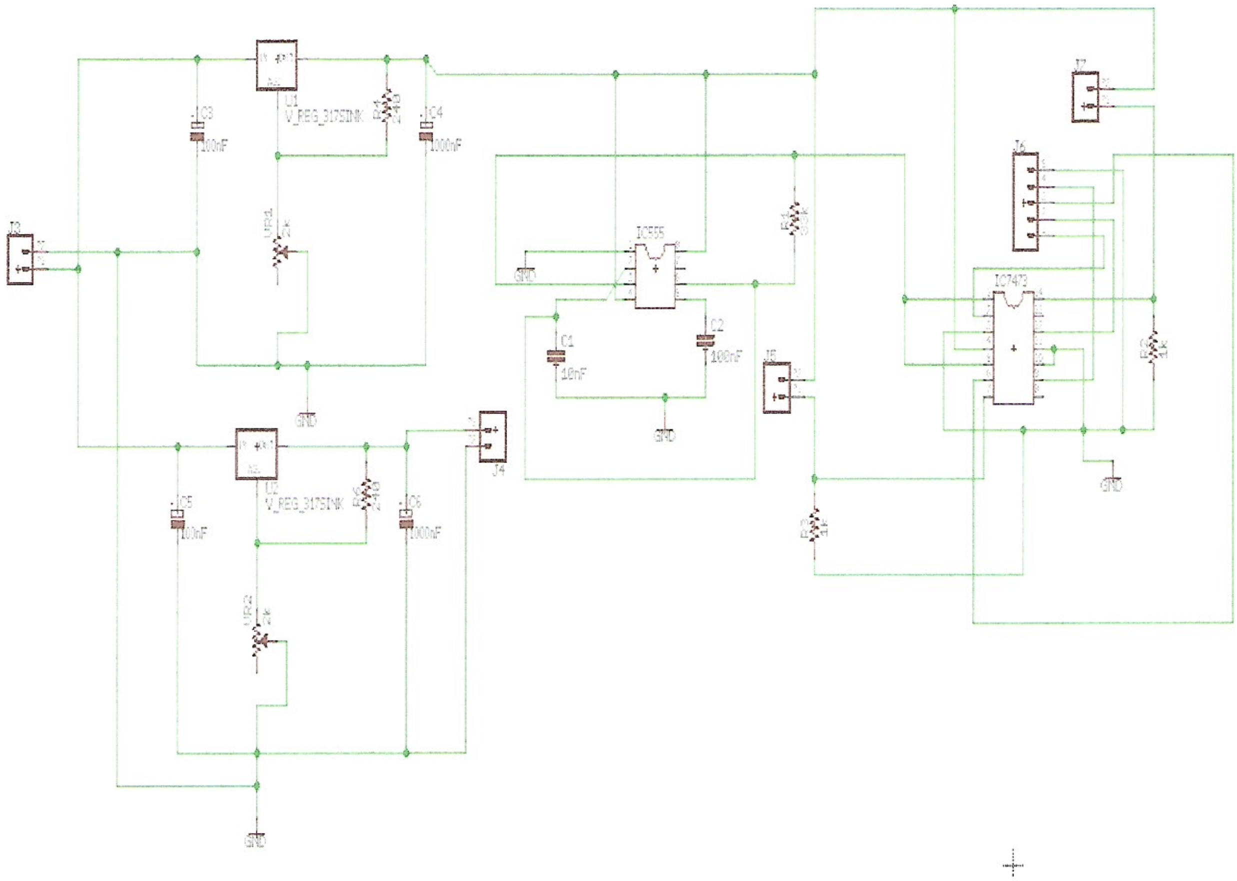


Fig. 34 Circuit schematic

4.3.5 PCB LAYOUT:

1) BOTTOM SIDE:

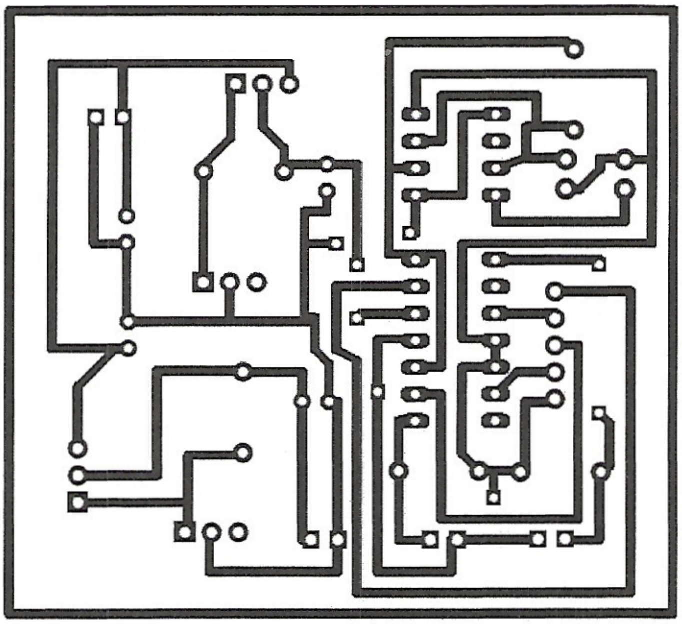


Fig. 35 PCB layout bottom side

2) TOP SIDE:

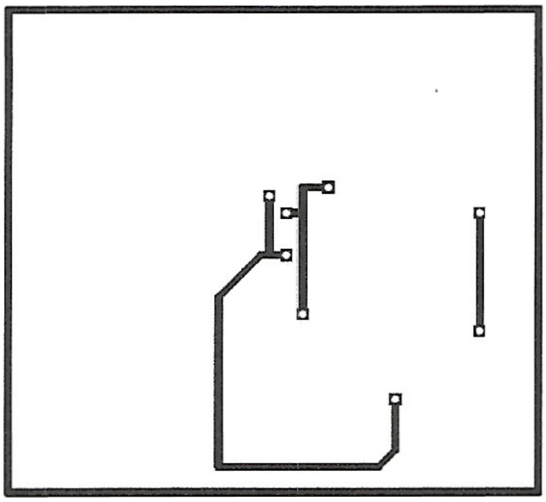


Fig. 36 PCB layout top side

4.4 HARDWARE SETUP

4.4.1 HARDWARE SETUP IN PRECEDING CAR (CAR A)

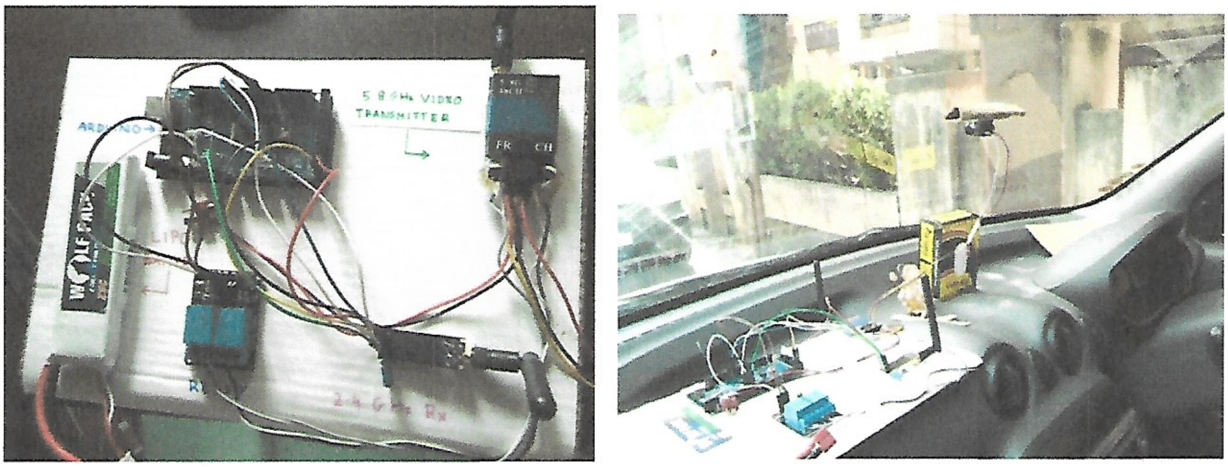


Fig. 37 Hardware setup in car A

4.4.2 HARDWARE SETUP IN FOLLOWING CAR (CAR B):-

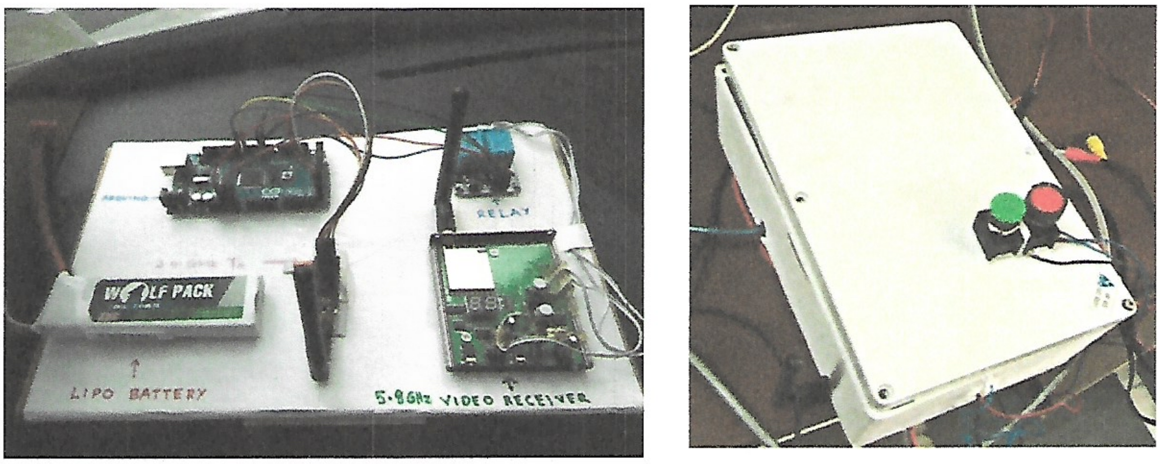


Fig. 38 Hardware setup in car B and the final plug and play kit box.

5.1 SUB MODULE ALGORITHMS

5.1.1 Localization of number plate

Input: Video Frames

Output: Localized number plate

Method Begins

Step 1: Read each frame of video being recorded by the webcam.

Step 2: Convert the image to grayscale.

Step 3: Apply morphological operation top hat to the grayscale image.

Step 4: Apply sobel operator to get edges in that image.

Step 5: Apply Gaussian blur to smoothen the image.

Step 6: Apply morphological operation closing on the image.

Step 6: Threshold the image using Otsu's adaptive thresholding.

Step 7: Detect contours in that thresholded image.

Step 7: With suitable (height/width) condition, select the rectangular contour in thresholded image which corresponds to rectangular number plate in original grayscale image. Extract that contour from original grayscale image.

Step 8: Apply adaptive histogram equalization and adaptive Otsu's thresholding to extracted image.

Step 9: Now, find contours in extracted thresholded image and select the contour with maximum area.

Step 10: Again extract this contour from previously extracted grayscale image. This new extracted segment is nothing but the number plate.

Method ends.

5.1.2 Character Segmentation

Input: Localized number plate.

Output: Segmented number plate character.

Method Begins.

Step 1: The extracted grayscale output image from previous stage is given as an input to this stage.

Step 2: The colour image is converted to grayscale.

Step 3: The grayscale image is then thresholded using binary thresholding method.

Step 4: Thresholded image is then inverted for detection of contours.

Step 5: Detect contours in that inverted image.

Step 6: For every contour, we have defined a bounding rectangle and furthermore, these bonding rectangles are filtered out according to conditions bases on area, height, width of the contour so as to get only characters of the number plate and neglect other false negatives.

Step 7: The characters of the number plate are then extracted with the help of these contours and they are stored for further processing.

Method ends.

5.1.3 Convolutional Neural Network

Input: Segmented number plate character.

Output: Predictions in the form of string.

Method Begins.

Step 1: The segmented number plate images are read from the stored locations and are passed to the trained convolutional neural network model.

Step 2: The convolutional neural network model predicts the character present in the image and gives the corresponding output class.

Step 3: Predictions of each individual character from the number plate are sequentially grouped together to an output in string format.

Step 4: This String is then passed to Arduino via PySerial.

Method ends.

5.1.4 Controller on video receiver side

Input: Vehicle number plate in string format.

Method Begins.

Step 1: Wait for ANPR code to send the number plate string on serial port.

Step 2: As soon as it receives the encoded number plate in string format, it converts the number plate string into 40-bit hex address required by nRF24L01 for establishing communication.

Step 3: It compares this address with previous address. If they doesn't match, it means that a new vehicle has been detected.

Step 4: It then sends a communication establishment request to this new address and wait for acknowledgement.

Step 5: If it receives an acknowledgement, it means that the car in front is also ready to establish communication.

Step 6: It then traverses through all the video communication channels to find the channel of least RSSI value. It ensures that there is a least probability that there is a communication going on this channel in the vicinity.

Step 7: It then sends this channel number to nRF24L01 of car in front. The nRF of car in front then tunes its video transmitter on that channel and sends an acknowledgement.

Step 8: After receiving acknowledgement, controller then tunes the video receiver to same channel number.

Step 9: Now checks the status of start button. When user presses the start button, controller turn on the TFT Screen through relay module. User can now see the video transmitted by car in front.

Step 10: When user presses the stop button, TFT Screen is turned off and a STOP message is sent to car in front to break the communication completely. The video receiver in car is then tuned to default channel number which is 61.

Method ends.

5.1.5 Controller on video transmitter side

Method Begins.

Step 1: It waits for nRF24L01 module to receive a communication establishment request.

Step 2: As soon as it receives the request, it acknowledges the requesting nRF24L01 indicating that it is ready to establish a video communication.

Step 3: Now it waits for the requesting nRF24L01 module to send video communication channel number.

Step 4: It then tunes its video transmitter on that channel number using relay module and acknowledges back indicating that video transmitter is tuned to desired channel. Video communication has now begin.

Step 5: Now when it receives a STOP message, it tunes video transmitter to default channel number which is 48 and breaks the ongoing video communication.

Method ends.

6.1 RESULTS:

6.1.1 Results of Number Plate Recognition Algorithm

6.1.1.1 Input video frame = 324

Fig. 39 video frame captured by camera

6.1.1.2Thresholded video frame = 324

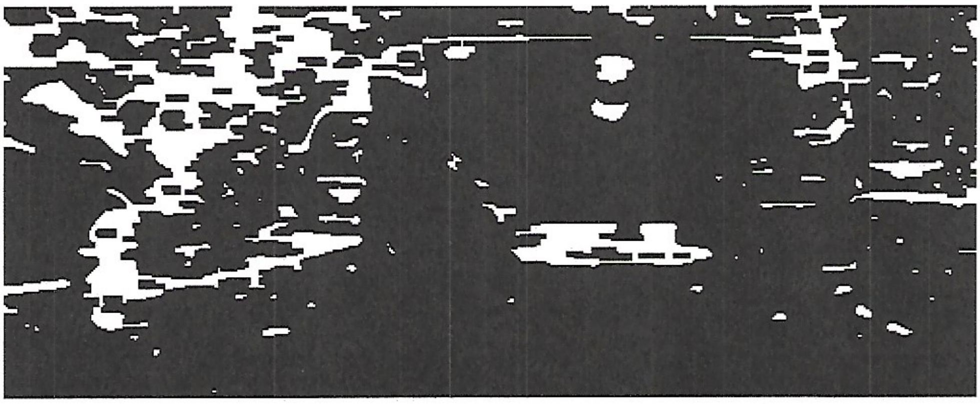


Fig. 40 thresholded video frame showing contour of number plate

6.1.1.3 Localized in video frame = 324

Fig. 41 Number plate localised in video frame

6.1.1.4 Extracted Number plate

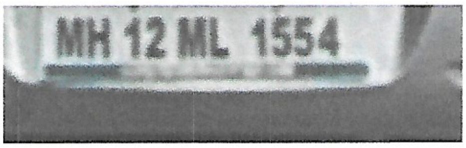


Fig. 42 Number plate extracted for segmentation

6.1.1.5 Segmented characters of the number plate

Fig. 43 Number plate characters segmented

6.1.1.6 Results of CNN

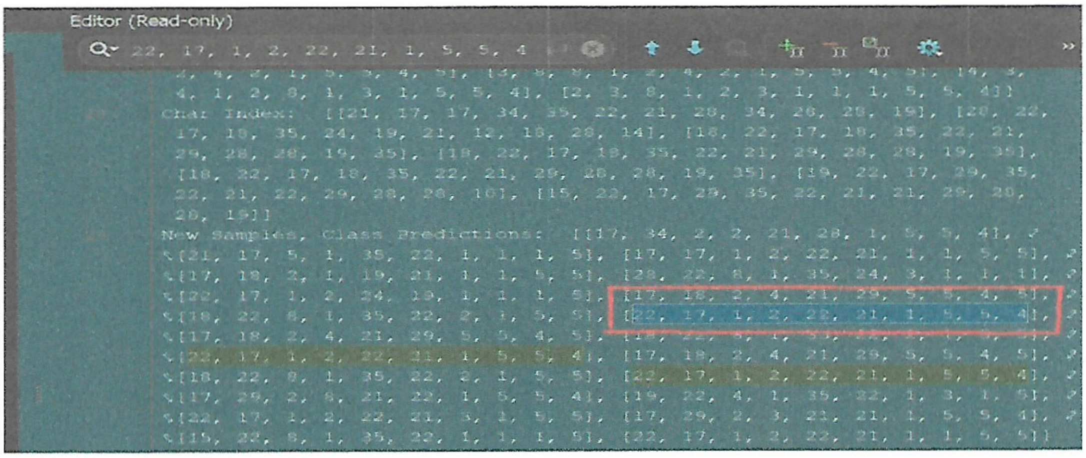


Fig. 44 CNN model output for input number plate characters

6.1.2 Final Results:

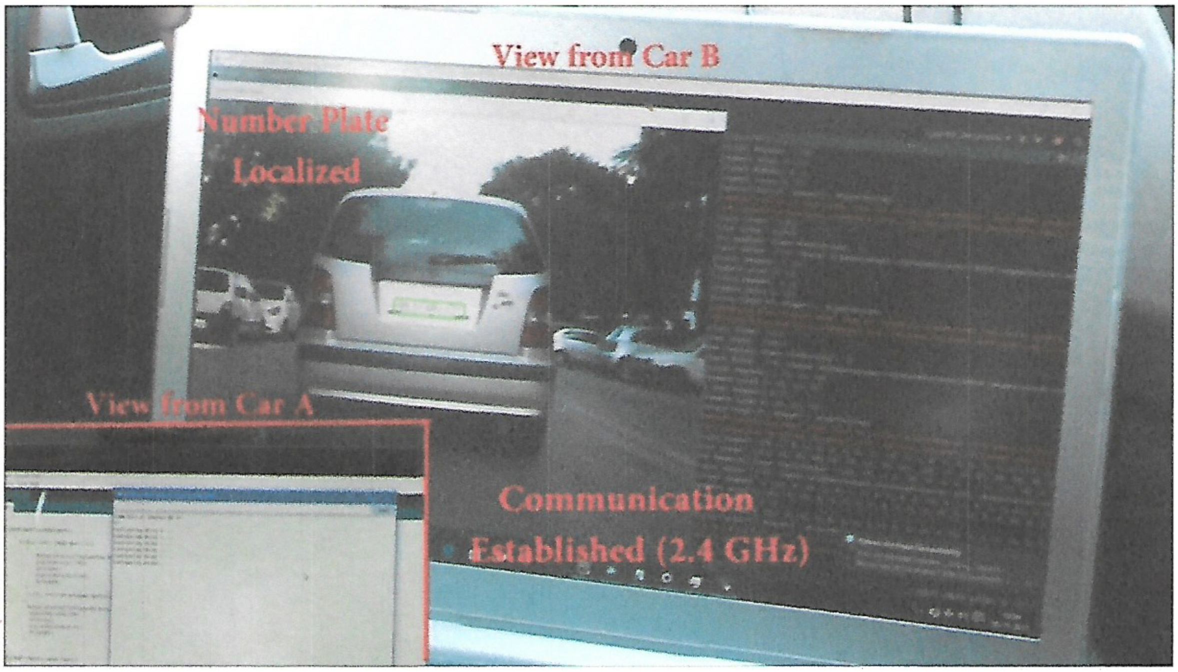


Fig. 45 Number plate detection and establishing communication

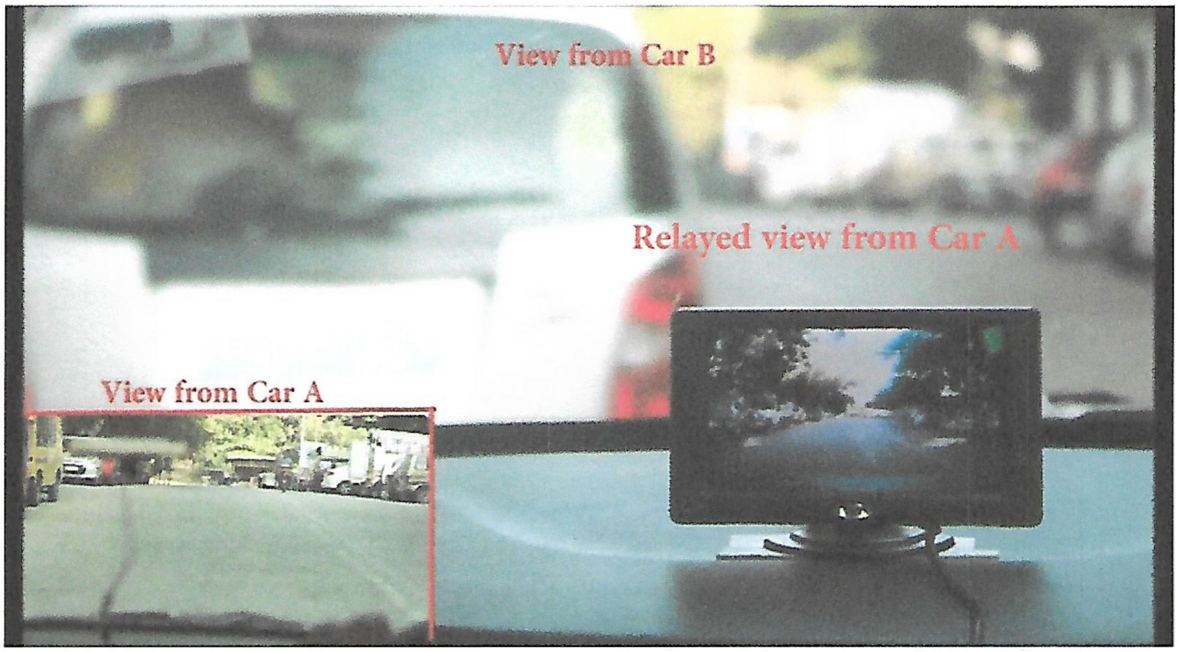


Fig. 46 Final video transmission

6.1.3 Comparison with existing solution:

The following comparison table compares our solution with one of the existing solutions in the market: The Samsung Transparent Truck. It gives a comprehensive analysis of the pros and cons of both the solutions

Comparison with existing solutions with advantages: No comparable technologies have hit the roads yet apart from a concept from Samsung called the Transparent Truck

Parameter	Current Solution	Your Solution	Advantages
Communication model	No Vehicle to Vehicle communication involved	Solution based on Vehicle to Vehicle communication.	Makes the system usage broader.
Video Display	Huge LCD screens on the back of the truck	Dashboard or custom display inside the car	More economically feasible
Ease of system implementation	Specialized hardware needs to be installed in vehicle. Not practically feasible.	Vehicle independent hardware, as our system contains a pair of Rx and Tx.	Can be implemented on **existing** cars
Application	Developed specifically for	Can be implemented on	Broader usage scope

Table 2 Comparison between Samsung's transparent Truck and our system

6.2 CONCLUSION:

For this project, we have tried to come up with an innovative solution to tackle a problem which causes a large loss of life on our roads. We have delved into futuristic concepts of Vehicle to Vehicle Communication and Smart Cars. We have used number plate localization using image processing and Convolutional Neural Networks to detect the characters of number plate with the help of which, we are transmitting live dash-cam footage uniquely from car to car using a dual frequency system. We have successfully implemented our prototype on a pilot level i.e. two real cars. We hope to convert our prototype into a complete product which can be implemented commercially. From the testing that we have performed so far, we can conclude following things:

1) For better segmentation of the characters and proper localization of number plate, the vehicle should come within a permissible range. On our current camera, the range is 3-5 meters.
2) The hardware setup has been working quite well for the duration it takes to complete the overtaking manoeuvre and at the appropriate distance. But still, more test samples are required to obtain a definitive range for our hardware setup.

6.3 FUTURE SCOPE

The grander aim of this project has been to delve into the concept of vehicle to vehicle communication. In an ideal world in the future, every vehicle will be connected with and will communicate with every other vehicle on the road to provide an extremely rich and safe driving experience. But at the present, we are very far away, both in terms of the technology and the connectivity infrastructure, from achieving this goal. Our project aims to operate at an introductory level of this concept. It aims to establish a dedicated communication link with only the vehicle in front, and taking live, real time inputs from that vehicle to ensure a safer experience for the driver. In this project, we have transmitted live video footage between two vehicles to make roads transparent. This concept can be further extended, and once the communication has been established, more inputs like instantaneous and relative speed, acceleration, relative distance, location and inputs from other vehicular sensors can be transmitted. These inputs can be used to develop a full-fledged overtaking assistance algorithm with speech assistance for driver. The video-transmission technology can also be useful in the autonomous car technology.

Our project can be implemented in existing cars as a standalone kit, but widespread implementation can be a little impractical. A more practical approach is the integration of our system with the infotainment system of cars.

Through this project, we have tried to simulate, on real existing cars, how V2V can be implemented in the future. V2V is the future of automotive and is a technology which will lead to a decrease in loss of life and increase in driver comfort, as depicted by our project.

REFERENCES

RESEARCH PAPERS

[1] Boukerche A., Oliveira, H., Nakamura, E. and Loureiro, A. (2008). Vehicular Ad Hoc Networks: A New Challenge for Localization-Based Systems. *Computer Communications*, 31(12), pp.2838-2849.

[2]Badr A., Abdelwahab M., Thabet A., and Abdelsadek A., "Automatic Number Plate Recognition System", Annals of the University of Craiova, Mathematics and Computer Science Series Volume 38(1), 2011, Pages 62–71.

[3]Gomes P., Olaverri-Monreal C., and Ferreira M., "Making vehicles transparent through V2V video streaming,"*IEEE Trans. Intell. Transp. Syst.*,vol. 13, no. 2,pp.930-938, June 2012.

[4]Olaverri-Monreal C., Gomes P., Fernandes R., Vieira F., and Ferreira M., "The see-through system: A VANET-enabled assistant for overtaking maneuvers," in *Proc.IEEE IV Symp.*, 2010, pp. 123-128.

[5]Mandellos N., Keramitsoglou I. and Kiranoudis C. (2011). A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*, 38(3), pp.1619-1631.

[6] Hijazi S., Kumar R., and Rowen C., "Using Convolutional Neural Networks for Image Recognition," IP Group, Cadence.

WEBSITES

[1] https://www.tensorflow.org/programmers_guide/

[2] https://www.tensorflow.org/api_docs/python/

[3] http://tmrh20.github.io/RF24/classRF24.html

[4] https://www.azoft.com/blog/license-plate-recognition-ios/

[5] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

[6] https://www.codeproject.com/Articles/16419/AI-Neural-Network-for-beginners-Part-of