AUTOMATIC LICENSE NUMBER PLATE RECOGNITION SYSTEM USING DEEP LEARNING

A Project Report submitted in the partial fulfilment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

CSE (ARTIFICIAL INTELLIGENCE)

Submitted By

K. ESWAR CHARAN ROHITH	**(20471A4328)**
I. KRISHNA SAI RAM	**(20471A4321)**
J.YASWANTH ABHISHEK VARMA	**(20471A4322)**
T.PRASANTH KIMAR	**(20471A4357)**

Under the Esteemed Guidance of

Mrs. P. NEELIMA M.Tech

Asst. Professor

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET (AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Cycle -1 NIRF rank in the band of 251-320 and ISO 9001:2015 Certified Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601.

2023-2024

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPET (AUTONOMOUS)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

CERTIFICATE

This is to certify that the project that is entitled with the name **"AUTOMATIC LICENSE NUMBER PLATE RECOGNITION SYSTEM USING DEEP LEARNING"** is a bonafide work done by the team **K. ESWAR CHARAN ROHITH (20471A4328), I. KRISHNA SAI RAM (20471A4321), J. YASWANTH ABHISHEK VARMA (20471A4322), T. PRASANTH KUMAR (20471A4357)** in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the Department of **CSE (ARTIFICIAL INTELLIGENCE)** during 2023-2024.

PROJECT GUIDE

Mrs. P. Neelima, M.Tech.,
Asst. Professor

PROJECT COORDINATOR

Mr. K. Jaya Prakash, M.Tech, (Ph.D.),
Asst. Professor

HEAD OF THE DEPARTMENT

Dr. B. Jhansi Vazram, M.Tech, Ph.D.,
Professor & Head

EXTERNAL EXAMINER

Viva Voce Date: ………

DECLARATION

We hereby declare that the work described in this project work, entitled **"Automatic License Number Plate Recognition System Using Deep Learning"** which is submitted by us in partial fulfilment for the award of **Bachelor of Technology** in the Department of **CSE (Artificial Intelligence)** to the **Narasaraopeta Engineering College**, is the result of work done by us under the guidance of **Mrs. P. Neelima**, Asst. Professor, Dept of IT & CSE (AI)

The work is original and has not been submitted for any Degree/ Diploma of this or any other university.

K. Eswar Charan Rohith	**(20471A4328)**
I. Krishna Sai Ram	**(20471A4321)**
J. Y.Abhishek Varma	**(20471A4322)**
T. Prasanth Kumar	**(20471A4357)**

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairperson **Sri M.V. Koteswara Rao** **B.Sc.,** who took keen interest on us in every effort throughout this course.We owe our gratitude to our principal **Dr. M. Sreenivasa Kumar** **M. Tech., Ph.D., MISTE, FIE,** for his kind attention and valuable guidance throughout the course.

We express our deep-felt gratitude to **Dr. B. Jhansi Vazram** **M. Tech, Ph.D.,** Professor & Head Department of IT & CSE (AI) and also to Project Coordinator **Mr. K. Jaya Prakash** **M. Tech, (Ph.D.), Asst. Prof**, Department of IT & CSE (AI) whose unstinting encouragement enabled us to accomplish our project successfully and in time.

We extend our sincere thanks to our guide, **Mrs. P.Neelima** **M. Tech., Asst. Prof,** of IT & CSE (AI), for extending her valuable guidance and encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department fortheir cooperation and encouragement during our B. Tech degree. We have no words to acknowledge the warm affection, constant inspiration and encouragement that we receive from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts, which had really helped us in successfully completing our project.

By

K.Eswar Charan Rohith	**(20471A4310)**
I.Krishna Sai Ram	**(20471A4340)**
J.Yaswanth Abhishek Varma	**(20471A4342)**
T.Prasanth Kumar	**(20471A4357)**

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF CSE (ARTIFICIAL INTELLIGENCE)

VISION OF THE DEPARTMENT

To be renowned department that imparts creative, learning and research skills to students in the domain of artificial intelligence.

MISSION OF THE DEPARTMENT

To Impart Strong foundation of statistics for understanding Artificial Intelligence.

M1: To establish high performance computational facilities and tools to develop innovative and intelligent solutions.

M2: To collaborate with renowned companies for multidisciplinary research and development.

M3: To guide the students in learning and creative for developing intelligent technology-based solutions to societal problems.

Program Specific Outcomes (PSO's)

PSO1: Ability to analyze and apply the knowledge of human cognition, Artificial Intelligence, Machine Learning and data engineering in terms of real world problems to meet the challenges of the future.

PSO2: Ability to develop computational knowledge and project development skills using innovative tools and techniques to solve problems in the areas related to Deep Learning, Machine learning, Artificial Intelligence.

PSO3: Ability to lead a product development company/team and use the acquired knowledge to identify real-world research problems.

Program Educational Objectives (PEO's)

The graduates of the programme are able to:

PEO1: To Formulate, analyze and solve Engineering problems with strong foundation in Mathematical, Scientific, Engineering fundamentals and modern computing practices through advanced curriculum.

PEO2: Analyze the requirements, realize the technical specification and design the Engineering solutions by applying artificial intelligence theory and principles.

PEO3: Demonstrate technical skills, competency in AI and promote collaborative learning and teamwork spirit through multi -disciplinary projects and diverse professional activities.

PEO4: Equip the graduates with strong knowledge, competence and soft skills that allows them to contribute ethically to the needs of society and accomplish sustainable progress in the emerging computing technologies through life-long learning.

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Course Outcomes (CO'S):

CO421.1: Analyse the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO’s and PO’s:

1. **Low level**
2. **Medium level**
3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are applied in this project	Description of the Model	Attained PO
CC2204.2	Gathering the requirements and defining the problem, plan to develop a smart model for predicting genre of the song.	PO1, PO3
CC421.1, CC2204.3	Each and every requirement is critically analyzed and the process model is identified.	PO2, PO3
CC421.2, CC2204.2	Logical design is done by using the unified modelling language which involves individual teamwork.	PO3, PO5, PO9
CC421.3, C2204.3	Each and every model is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4	Documentation is done by all our three members in the form of a group.	PO10
CC421.5, CC2204.2	Each and every phase of the work in group is presented periodically.	PO10, PO11
CC2202.2, CC2203.3, CC1206.3, CC3204.3, CC4104.2	Implementation is done and the project id deployed by using a frontend.	PO4, PO7
CC32SC4.3	The design includes models like convolutional Neural Networks and Artificial Neural Networks.	PO5, PO6

ABSTRACT

Automatic Number Plate Recognition (ANPR) is an image processing technology which uses number (license) plate to identify the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a highly restricted area like military zones or area around top government offices e.g., Parliament, Supreme Court etc. The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address etc. Traffic control and vehicle owner identification has become major problem in every country. Sometimes it becomes difficult to identify vehicle owner who violates traffic rules and drives too fast. Therefore, it is not possible to catch and punish those kinds of people because the traffic personal might not be able to retrieve vehicle number from the moving vehicle because of the speed of the vehicle. Therefore, there is a need to develop Automatic Number Plate Recognition (ANPR) system as a one of the solutions to this problem. There are numerous ANPR systems available today. These systems are based on different methodologies but still it is really challenging task as some of the factors like high speed of vehicle, non-uniform vehicle number plate, language of vehicle number and different lighting conditions can affect a lot in the overall recognition rate. Most of the systems work under these limitations.

The License Plate detection and recognition is a challenging task that plays a significant role in intelligent transportation systems. Where it could be used as a core in various applications, such as security, traffic control, and electronic payment systems (e.g., freeway toll payment and parking fee payment). A variety of algorithms are developed for this work and each one has advantages and disadvantages for extracting plates in images under different circumstances. However, the complexity of some methods requires a high calculation cost and this could be time-consuming. In the current paper, a simple and efficient method is proposed to tackle the issue of license plate detection and character recognition. The license plate is detected first based on the two-dimensional wavelet transform to extract the vertical edges of the input image. The high density of vertical edges is computed first to detect the potential areas of the license plate. Then these potential areas are verified by using a plate/non-plate CNN classifier. After the license plate is detected, the characters are segmented by using a simple

method that is based on the empty distance between the characters. Finally, these character candidates are classified by training another CNN classifier.

In the result analysis, it was found that among the three algorithms evaluated for automotive security, Number Plate Recognition stood out with an impressive accuracy rate of 96%. Its exceptional performance in accurately extracting alphanumeric characters from license plates demonstrated its suitability for the project's objectives. Consequently, Number Plate Recognition was selected as the preferred algorithm, ensuring robust functionality and reliability in real-world applications.

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

The Automatic Number Plate Recognition (ANPR) was invented in 1976 at the Police Scientific Development Branch in the UK. However, it gained much interest during the last decade along with the improvement of digital camera and the increase in computational capacity. It is simply the ability to automatically extract and recognition a vehicle number plate's characters from an image. In essence it consists of a camera or frame grabber that has the capability to grab an image, find the location of the number in the image and then extract the characters for character recognition tool to translate the pixels into numerically readable character. ANPR can be used in many areas from speed enforcement and tool collection to management of parking lots etc. It can also be used to detect and prevent a wide range of criminal activities and for security control of a highly restricted areas like military zones or area around top government offices. The system is computationally inexpensive compare to the other ANPR systems. Besides the robustness, the earlier methods use either feature based approached using edge detection or Hough transform which are computationally expensive or use artificial neural network which requires large training data.

The ANPR system works in three steps, the first step is the detection and capturing a vehicle image, the second steps is the detection and extraction of number plate in an image. The third section use image segmentation technique to get individual character and optical character recognition (OCR) to recognize the individual character with the help of database stored for each and every alphanumeric character. The major steps to accomplish the proposed work can be given as

- Image acquisition
- Number plate recognition
- Edge Detection
- Character Segmentation
- Character Recognition

The first step is the capturing of an image using the USB camera connected to the PC. The images are captured in RGB format so it can be further process for the number plate extraction.

The second step of the ANPR algorithm is the extraction of the number plate in an image. A yellow search algorithm is used to extract the likelihood ROI in an image. As the official number plate of Sindh has yellow background with alphanumeric character written in black, it is easy to detect the plate area by searching for yellow pixels. The image is search for the yellow color pixels or some which are closer to yellow in value. If pixel value is of yellow color the pixel is set to 1, otherwise the pixel value is set to 0. The image obtained after the search algorithm is in black and white format. After identify the ROI, image is then filtered using two different filtering techniques. The first technique involves removing of all white patches that are connected to any border and set their pixel value to 0. The second filtering technique use pixel count method to remove the small regions in an image other than the plate region. The number of consecutive white pixels is inspected and regions that contain number of white pixels less than the predefined threshold are set to 0. At this stage the image contains only the vehicle number plate.

The third step of the developed ANRP algorithm uses Optical Character Recognition (OCR) algorithm to recognize the vehicle number. The resultant cropped image obtained after the second step is inverted i.e., all white pixels are converted to black and black pixels to white. Now the text is in white and the plate background is black. Before applying the OCR the individual lines in the text are separated using line separation process. The line separation adds the each pixels value in a row. If the resultant sum of row is zero that means no text pixel is present in a row and if the resultant sum of row is greater than zero that means the text is present in row. The first resultant sum greater than zero represents the start of the line and after this the first resultant sum equal to zero represents the end of the line. The start and end values of the line is used to crop the first line in the text. The same process continues to separate the second line in the text.

Once the lines in an extracted vehicle number plate are separated, the line separation process is now applied column wise so that individual character can be separated. The separated individual characters are then stored in separate variables. The OCR is now used to compare 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 string is then compared with the stored database for the vehicle authorization. The resultant signals are given according to the result of comparison.

1.2 RELATED WORK

This section includes the work already done on this system by various researchers using different methodologies and algorithms. Following is the brief description of some of them:

Car Plate Recognition Using the Template Matching Method is proposed by M. I. Khalil. Generally, LPR system consists of 4 modules: Image acquisition, licensed plate extraction, segmentation & recognition of individual character. But template matching method does not need the "segmentation" process of input image. After the license plate extraction phase, INFORMATION RECOGNITION PHASE (IPR) is applied. For this phase "moving window technique" is used. To recognize the image the country name, the license plate image is loaded as main image. Then the first image entry of country image set is loaded as an object. The moving window technique is applied to detect that object within the image. If answer is "YES" then the name of country corresponding to country name is retrieved from the country names table. And if answer is "NO" then the next country name image is loaded as the object & this procedure is repeated till the end of the characters.

An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network is proposed by Kaushik Deba, Md. Ibrahim Khana, Anik Sahaa, and KangHyun Job. In this system they are using segmentation technique named as sliding concentric windows (scw). This method helps is analyze road images which are often contain vehicles And extract license plate from natural Properties by finding vertical and horizontal edges from vehicle region. On the Basis of a novel adaptive image segmentation technique is for detecting candidate region and Color verification for candidate region by using HSI color model on the basis of using hue and intensity in HSI colour model verifying green and yellow LP and white LP, respectively. Mainly there focus on artificial neural network (AAN) new algorithm which is based on Korean number plate system. If we try to follow they diagram above you will get clear idea about who this system working taking place. How the candidate region selection taking place and how grey image conversion taking in there.

SVM Based License Plate Recognition System is proposed by Kumar Parasuraman. SVM is a supervised learning technique, which takes Statistical Learning Theory (SLT) as its theoretical foundation, and the structural risk minimization as its optimal object to realize the best generalization. Two main approaches have been suggested for applying SVMs for multiclass classification. They are “one against all” and “one against one”. A number plate region is located by using mean shift method and extracted; the histogram projection method in horizontal direction is applied for a simple segmentation only. Then it is normalized into size of 140x36. Then 315 dimensional feature vectors are obtained by averaging values in 4x4 windows of the normalized sub-images. The feature vectors are used to train SVMs with RBF kernel.

1.3 EXISTING SYSTEM

The existing system usually for a license plate detection and character recognition (LPDR) system has mainly three phases. The first phase is image pre-processing, once the image is captured further processing of the image is carried out like converting the image from a color space to another, resizing the image resolution, and removing noises. The second phase is license plate localization, the region of interest is detected based on some license plate characteristics and image features. The final phase is the optical character recognition, this phase is considered the most crucial step because it helps to read the plate number and identify the vehicle.

DISADVANTAGES

A variety of algorithms are developed for this work and each one has advantages and disadvantages for extracting plates in images under different circumstances. However, the complexity of some methods requires a high calculation cost and this could be time-consuming.

From literature survey it has been observed that there are certain limitations about proposed algorithms like:

- Poor image resolution
- Less Accuracy
- Poor lighting and low contrast
- Higher Computational Cost
- Lack of standards of the plate of the vehicles
- Improperly segmented characters will result in misrecognized characters.

1.4 PROPOSED SYSTEM

The systematic study of the existing ALPR systems, the basic algorithm used, the variations in the existing algorithm are done to improve the overall system. Concentration on localization of license plates

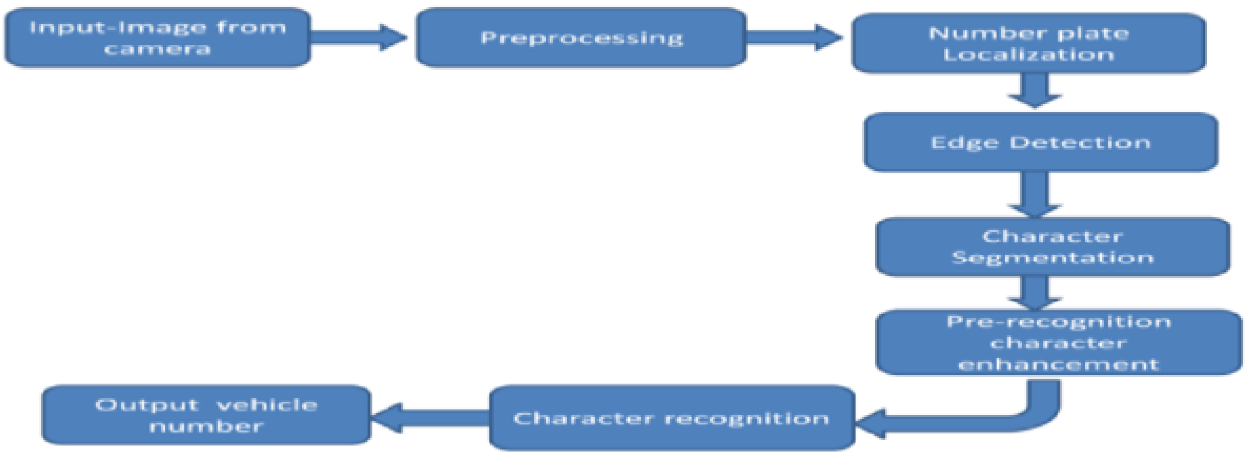


Fig 1.1 Proposed System

then go on to extract the characters by using morphological operations such as dilation, eroding the image, dilating, filtering etc. All these morphological operations leads to the efficiency of overall system.

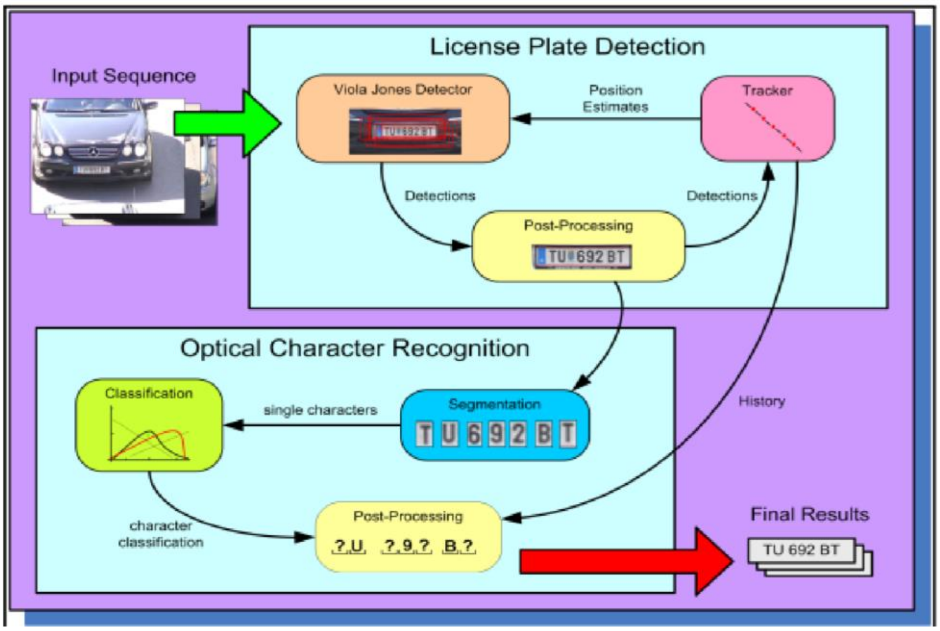


Fig 1.2 Overall view

ADVANTAGES

- Faster traffic management.
- Enhanced parking management.
- Better security and prevention of crimes like car thefts.
- Provides better evidence and lines of inquiry.

- Automates access control systems.
- Allows modern and effective law enforcement.

1.5 SYSTEM REQUIREMENTS

1.5.1 Hardware Requirements:

- Processor : Intel(R) Core™2 i7-5500U CPU @2.50GHz
- RAM : 8GB(gigabyte)
- System Type : 64-bit operating system, x64-based processor

1.5.2 Software Requirements:

- Browser : Any Latest browser like Chrome
- Operating System : Windows 10
- Language : Python
- Platform : Google COLAB

2. LITERATURE SURVEY

2.1 Deep Learning

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

Deep Learning architectures such as deep neural networks, deep belief networks, graph neural networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analogue.

The adjective "**deep**" in deep learning refers to the **use of multiple layers in the network**. Early work showed that a linear perceptron cannot be a universal classifier, but that a network with a non-polynomial activation function with one hidden layer of unbounded width can. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed connectionist models, for the sake of efficiency, trainability and understandability, whence the "structured" part.

2.2 Some Deep Learning Types

1. **Feedforward neural network:**
   - This type of neural network is the very basic neural network where the flow control occurs from the input layer and goes towards the output layer.
   - These kinds of networks are only having single layers or only 1 hidden layer.

- Since the data moves only in 1 direction there is no backpropagation technique in this network.

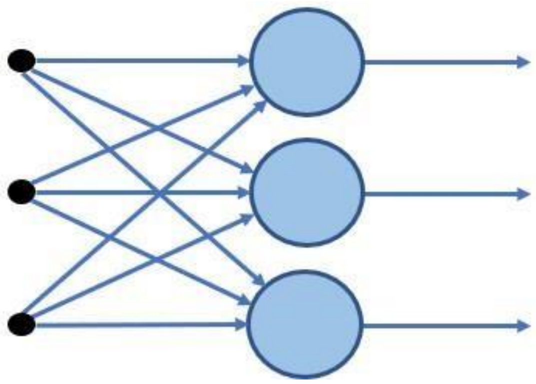


Fig: 2.1 Feed Forward Neural Network

- From the above fig 2.1 we can say that the network, the sum of the weights present in the input is fed into the input layer.
- These kinds of networks are used in the facial recognition algorithm using computer vision.

2. Radial basis function neural networks:

- This kind of neural networks have generally more than 1 layer preferably two layers.
- In this kind of networks, the relative distance from any point to the center is calculated and the same is passed towards the next layer
- Radial basis networks are generally used in the power restoration systems to restore the power in the shortest span of time to avoid the blackouts.
-

3. Multi-Layer perceptron:

- You can see the fig 2.2 where it shows the type of network are having more than 3 layers and its used to classify the data which is not linear.
- These kinds of networks are fully connected with every node.
- These networks are extensively used for speech recognition and other machine learning technologies.

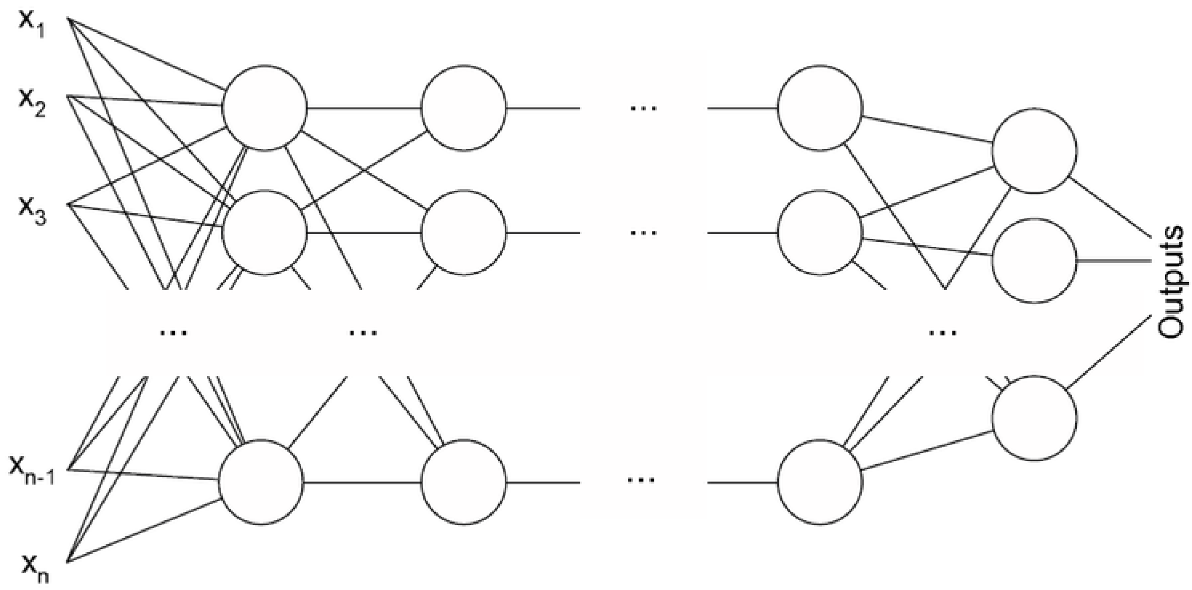


Fig: 2.2 Multi-Layer Perceptron

4. Convolution Neural Network

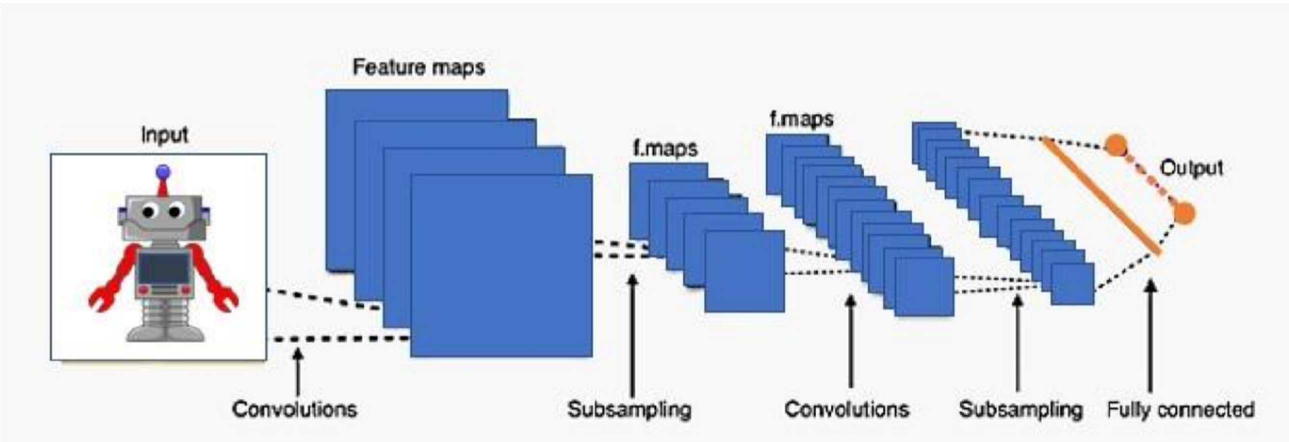


Fig: 2.3 Convolution Neural Network

- The entire process of CNN is been shown on fig 2.3 where it is one of the variations of the multilayer perceptron.
- CNN can contain more than 1 convolution layer and since it contains a convolution layer the network is very deep with fewer parameters.
- CNN is very effective for image recognition and identifying different image patterns.

5. **Recurrent Neural Network:**

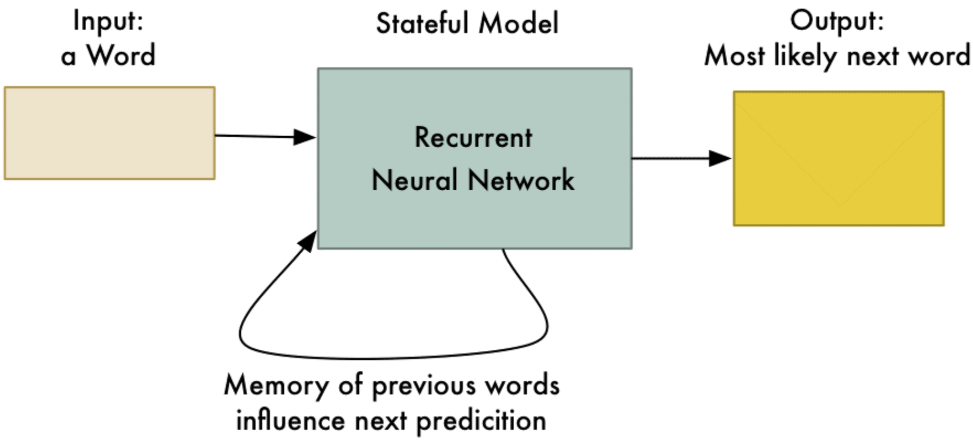


Fig: 2.4 Recurrent Neural Network

- RNN is a type of neural network where the output of a particular neuron is fed back as an input to the same node.
- These method helps the network to predict the output which is shown in the fig 2.4
- This kind of network is useful in maintaining a small state of memory which is very useful for developing the chatbot
- This kind of network is used in chatbot development and text to speech technologies.

6. **Modular Neural Network:**

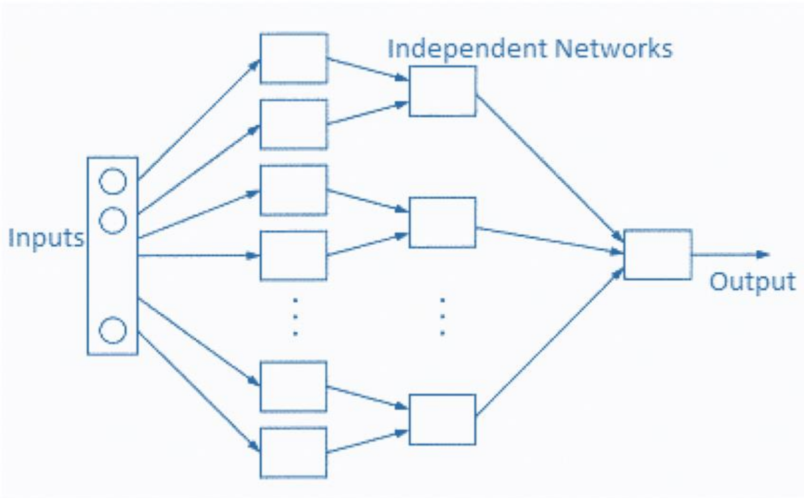


Fig: 2.5 Modular Neural Network

- This kind of network is not a single network but a combination of multiple small neural networks.

- All the sub-networks make a big neural network and all of them work independently to achieve a common target.
- These networks are very helpful in breaking the small-large problem into small pieces and then solving it which is shown in the fig 2.5

7. **Sequence to sequence models:**

- This type of network is generally a combination of two RNN networks.
- The network works on the encoding and decoding that is it consists of the encoder which is used to process the input and there is a decoder which processes the output.
- Generally, this kind of network is used for text processing where the length of the inputted text is not as same as output.

2.3 Applications of Deep Learning:

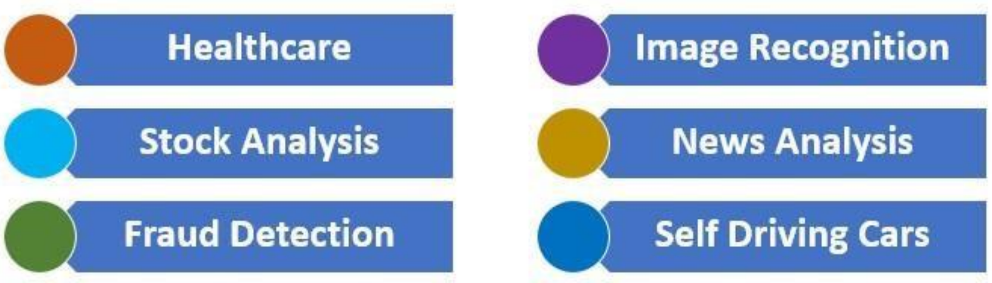


Fig: 2.6 Applications of Deep Learning

- Self-Driving Cars
- News Aggregation and Fraud News Detection
- Natural Language Processing
- Virtual Assistants
- Entertainment
- Visual Recognition
- Fraud Detection
- Healthcare
- Personalizations
- Detecting Developmental Delay in Children
- Colourisation of Black and White images
- Adding sounds to silent movies

- Automatic Machine Translation
- Automatic Handwriting Generation
- Automatic Game Playing
- Language Translations
- Pixel Restoration
- Photo Descriptions
- Demographic and Election Predictions
- Deep Dreaming

2.4 Characteristics of Deep Learning

1. Supervised, Semi-Supervised or Unsupervised

When the category labels are present while you train the data then it is Supervised learning. Algorithms like Linear regression. Logistic regression, decision trees use Supervised Learning. When category labels are not known while you train data then it is unsupervised learning. Algorithms like Cluster Analysis, K means clustering, Anomaly detection uses Unsupervised Learning.

The data set consists of both labeled and unlabelled data then we call it is Semi-Supervised learning. Graph-based models, Generative models, cluster assumption, continuity assumption use Semi-Supervised learning.

2. Huge Amount of Resources

It needs advanced Graphical Processing Units for processing heavy workloads. A huge amount of data needs to be processed like Big data in the form of structured or unstructured data. Sometimes more time also required to process the data, it depends on the amount of data fed in.

3. Large Amount of Layers in Model

A huge amount of layers like input, activation, the output will be required, sometimes the output of one layer can be input to another layer by making few small findings and then these findings are summed up finally in the softmax layer to find out a broader classification for final output.

4. Optimizing Hyper-parameters

Hyperparameters like no of epochs, Batch size, No of layers, Learning rate, needs to be tuned well for successful Model accuracy because it creates a link between layer predictions to final output prediction. Over-fitting and under-fitting can be well handled with hyper-parameters.

5. Cost Function

It says how well the model performance in prediction and accuracy. For each iteration in Deep Learning Model, the goal is to minimize the cost when compared to previous iterations. Mean absolute error, Mean Squared Error, Hinge loss, Cross entropy are different types according to different algorithms used.

2.5 Advantages of Deep Learning

- Solve Complex problems like Audio processing in Amazon echo, Image recognition, etc, reduce the need for feature extraction, automated tasks wherein predictions can be done in less time using Keras and Tensorflow.
- Parallel computing can be done thus reducing overheads.
- Models can be trained on a huge amount of data and the model gets better with more data.
- High-Quality Predictions when compared with humans by training tirelessly.
- Works well-unstructured data like video clips, documents, sensor data, webcam data, etc.

2.6 Deep Learning Algorithms

To create a deep learning model, one must write several algorithms, blend them together and create a net of neurons. Deep learning has a high computational cost. To aid deep learning models, there are deep learning platforms like Tensor flow, Py-Torch, Chainer, Keras, etc. In deep learning, we have tried to replicate the human neural network with an artificial neural network; the human neuron is called perceptron in the deep learning model. We connect these perceptron units together to create a neural network; it has 3 sections:

- Input layer
- Hidden layers
- Output layer

A perceptron has input nodes (dendrites in the human brain), an actuation function to make a small decision and output nodes (axon in the human brain). We will see how one perceptron works; connecting them together will create a deep learning model. Input information (number of input variables/features) are assigned some weight and fed to the actuation function. The actuation function makes a decision and sends output. This perceptron's output will be input to other neurons. Once the batch is processed, with backpropagation error is calculated at each neuron, with the help of a cost function/ cross-entropy. In this way, input weights are reassigned, and the whole process continues until cross-entropy satisfies the condition.

We have different actuation functions like Sigmoid functions, hyperbolic tangent function, Rectified Linear Unit (ReLU) to take a small decision. A deep learning model needs a vast amount of data to build a good model. Generally, a model with more than 3 hidden layers is treated as a deep neural network. Basically, Deep learning is a set of neurons with a number of parameters defined for each layer. To create the Deep Learning model, the popular architectures are RNN, CNN, etc.

2.7 Architectural Methods for Deep Learning Algorithms

To build this architecture following algorithms are used:

1. Back Propagation

In this algorithm, we calculate partial derivatives. In general, the gradient descent method for optimization, derivatives (gradients) are calculated at each iteration. In deep learning, functions are not simple; they are the composition of different functions. In this case, it is hard to calculate gradients, so we use approximate differentiation to calculate derivatives. The more the number of parameters, the more expensive approximate differentiation will be.

2. Stochastic Gradient Descent

In Gradient descent, the goal is to find global minima or optimum solution. But to get that, we have to consider local minima solutions (not desirable) also. If the objective function is a convex function, it is easy to find the global minima. The initial value for the function and learning rate are deciding parameters for finding global minima. This can easily be understood by considering a river from the mountain top and searching for a foothill (global minima). But in the way, there will be some ups and downs (local minima) which must be avoided. The river originating point and speed (initial value and learning rate in our case) are deciding factors to find global minima.

3. Learning Rate

The learning rate is like the speed of the river; it can reduce training time and increase performance. In general, to learn any technique/sport, in the beginning, the learning rate is relatively high than at the end when one is to master it. After the intermediate stage, the learning will be slow; the focus will be on fine-tuning. The same is applied in deep learning; too large changes are tackled by a higher learning rate and by slowly decreasing the learning rate later for fine-tuning.

4. Batch Normalization

In deep learning initial value of weight (randomly chosen) and learning rate is defined for a minibatch. In the beginning, there would be many outliers, and during backpropagation, these outliers must be compensated to compute the weights to get output. This compensation results in extra epochs. So to avoid it, we use batch normalization.

5. Drop Out

In deep learning, we generally encounter the problem of over fitting. Over fitting in large networks with several parameters makes it difficult to predict on test data. So, to avoid that, we use the dropout method, which drops random units during training by creating different 'thinned networks'. When testing these thinned networks' predictions are averaged, which helps to avoid over fitting.

6. Bag of Words

We use a continuous bag of words to predict the next word. For e.g., we see in email writing the autosuggestion for completing the sentence is part of NLP. This is done by considering lots of sentences and for a specific word surrounding words that are captured. These specific words and surrounding words are fed to the neural network. After the training model, it can predict the specific word based on the surrounding words.

7. Long Short Term Memory

LSTM is very useful in sequence prediction problems like language translation, predicting sales and finding the stock price. LSTM has the edge over other techniques because it is able to consider previous data. LSTM makes modification by cell states mechanism. It remembers to forget things. The 3 main aspects of LSTM make it stand out from other deep learning techniques. First is when the neuron should have input, second when to remember previous data and what to forget and third is when to pass output.

2.8 Libraries of Deep Learning

All the libraries which are generally used for deep learning are open source and few of them are as follows:

- TensorFlow
- deeplearning4j
- Torch
- Caffe
- Microsoft CNTK

- ML.NET
- Theano
- Deepmat
- Neon

1. TensorFlow

- TensorFlow is the machine learning and deep learning library developed by Google and it came into the market around the 2016 march.
- TensorFlow grew out of an in-house library of google brain known as DistBelief.
- Currently, TensorFlow is the leading and most used library in the market.
- Different types of deep nets can be developed and also the various packages available in this library are used to attain and address most of the tasks and problems in the field of deep learning.
- This library is completely written in python and so it’s easy to use for the python programmers.
- Due to a flexible computational graphical structure of TensorFlow, this library is not only limited to deep learning operations it can be used for many different operations and applications.
- TensorFlow provides a layer or we can say more of a wrapper, known as Keras which is used to access the different packages and methods easily of TensorFlow.
- Google provides a very well documentation for this library where every small intricacies and usage are mentioned anyone can refer that and use the library.
- TensorFlow is a very fast-evolving library, this library can be used for educational purposes as well as huge large scale commercial application can also be built.
- Google has developed this library for Mobil platforms as well which is known as TensorFlow lite.
- TensorFlow is the only library which provides support for Python, Java, C++, javascript and swift programming language, for Javascript TensorFlow.js
- TensorFlow has also support for GPU and big data.

2. Deeplearning4j

- Deeplearning4j is the open-source java library which only supports java programming language and this library is written in Java.
- This was developed by Adam Gibson to provide distributed multimode capabilities for deep neural networks.

- This library is very much use full for the application which is having build on top of big data.
- This library works with Scala and also provide inbuilt GPU support.

3. Torch

- This open-source deep-learning library was developed by Facebook and Twitter.
- This library is written in Lua programming language.
- However PyTorch is the library which is widely used, and it's written in a python programming language.

4. Caffe

- Caffe is an open-source deep-learning library written in C++/CUDA and developed by Yangqing Jia of Google.
- This library was first developed for computer vision tax.
- Caffe gives permission to the user to configure the hyperparameters for a deep net.
- The layer configuration is very robust and very much sophisticated.

5. Theano

- This is the open-source deep-learning library written in Python and CUDA.
- This library is very similar to the TensorFlow library but the implementation and usage are not that simple as that of TensorFlow.
- This library is generally used for educational and research purposes.
- Theano is not that easy to use and many deep learning libraries extend the features of this library to help ease the life of the developer for coding the deep learning models.
- Theano is fastest amongst most of the libraries mentioned because it makes use of vectors and matrices for all the functions and the vectorized code runs faster since parallel processing for the multiple values makes the things faster.

6. Microsoft CNTK

- This is a cognitive toolkit developed by Microsoft to venture in the field of Artificial intelligence.
- This library is written in python and its supports the other packages and libraries which python programming language supports, and it comes with Microsoft visual studio.
- CNTK is used to describe neural networks as a series of computational directed graphs.

7. ML.NET

- ML.NET is the open-source library which is also developed by Microsoft for the dot net developers.

- This library is written in C# and F# and it uses the Microsoft dot net platform.
- With the library, it becomes easy to create desktop as well as large scale web applications which can bring the vast possibility of the machine learning algorithm to the end-user.

8. Deepmat

- This library is developed in MATLAB.
- With the use of this library, we can implement deep learning using MATLAB.
- with this library GSN, CNN, Restricted Boltzmann machine, Deep belief networks,multi-layer perceptron, and many more artificial neural networks.

9. Neon

- Neon is a deep learning framework created by the Nervana systems to deliver industry-leading cutting edge technologies.
- This framework has been depreciated as of 2018 and further research has been carried out by Intel corporation on the same.
- As per the Intel corporation website, alternative frameworks are asked to be used such as Intel optimization for tensorFlow, Intel optimization for Caffe, pytorch etc.

Layers in Convolutional Neural Networks

Below are the Layers of convolutional neural networks:

1. **Image Input Layer:**

The input layer gives inputs (mostly images), and normalization is carried out. Input size has to be mentioned here.

2. **Convolutional Layer:**

Convolution is performed in this layer. The image is divided into perceptrons(algorithm); local fields are created, leading to the compression of perceptrons to feature maps as a matrix with size m x n.

3. **Non-Linearity Layer:**

Here feature maps are taken as input, and activation maps are given as output with the help of the activation function. The activation function is generally implemented as sigmoid or hyperbolic tangent functions.

4. **Rectification Layer:**

The crucial component of CNN, this layer does the training faster without reducing accuracy. It performs element-wise absolute value operation on activation maps.

5. **Rectified Linear Units (ReLU):**

ReLU combines non-linear and rectification layers on CNN. This does the threshold operation where negative values are converted to zero. However, ReLU doesn't change the size of the input.

6. **Pooling Layer:**

The pooling layer is also called the downsampling layer, as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. This layer ignores less significant data; hence image recognition is done in a smaller representation. This layer reduces overfitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions, and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.

7. **Dropout Layer:**

This layer randomly sets the input layer to zero with a given probability. More results in different elements are dropped after this operation. This layer also helps to reduce overfitting. It makes the network to be redundant. No learning happens in this layer. This operation is carried out only during training.

8. **Fully Connected Layer:**

Activation maps, which are the output of previous layers, is turned into a class probability distribution in this layer. FC layer multiplies the input by a weight matrix and adds the bias vector.

9. **Output Layer:**

FC layer is followed by softmax and classification layers. The softmax function is applied to the input. The classification layer computes the cross-entropy and loss function for classification problems.

10. **Regression Layer:**

Half the mean squared error is computed in this layer. This layer should follow the FC layer.

Common steps for any Tensorflow based Algorithms:

The basic steps of TensorFlow algorithm are:

Step 1: Data is Imported/Generated: TensorFlow Models depends heavily on the huge amount of Data. Either you can import your own dataset or TensorFlow also comes with the collection of Type this command to check out available datasets in TensorFlow.

import TensorFlow as tf

import TensorFlow datasets as tendata

#This command will generate a list of datasets available in the TensorFlow

print(tfds.list_builders())

Step 2: Data Normalization or Transformation: If the data is not in the appropriate forum. The Batch

Normalization is the command approach used to normalize data in the TensorFlow.

Step 3: Set the Parameters of the Algorithm: For eg; the number of Iterations, Learning rate, etc.

Step 4: Set and initialize the variables and Placeholders: Variables and Placeholders are two basic programming Elements of the TensorFlow. Variables hold the state of the graph and placeholders are used to feed the data in the graph at the later date.

Step 5: Create Model structure: What operations will be performed on the data is defined.

Step 6: Define the Loss Function: It calculates the difference between predicted values and actual values. It tells how well your model is trained basically used to evaluate the output.

Step 7: Train Model: Initialize computational graph and create an Instance of a graph. Feed data into the model with the help of placeholders and let the TensorFlow do the rest of the processing for better predictions.

Step 8: Evaluate the performance: Evaluate the model by checking with new data.

Step 9: Predict the Outcome: Also checks your model on new and unseen data.

To better visualize model TensorFlow provides Tensorboard. It helps us to visualize any statistics of the neural network, debug and optimize them. You can check what happens in the code and will give you a detailed understanding of the inner working. You can fix problems very easily with the help of this tool. Tensorboard provides five types of Visualizations:

- Scalars
- Images
- Audio
- Histograms

The summary function of the TensorFlows gives us a detailed summary according to the specified format. To allocate resources, hold intermediate results and variables and execute graphs or part of graph session function is used.

3. SYSTEM ANALYSIS

3.1 Scope of the project

The scope of this system is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate.

3.2 Analysis

The system consists of two modules: License Plate Detection and License Plate Character Recognition.

As we are providing the link for the dataset we have took from the kaggle to train the model with the best algorithm. Dataset link: https://www.kaggle.com/datasets/andrewmvd/car-plate-detection.

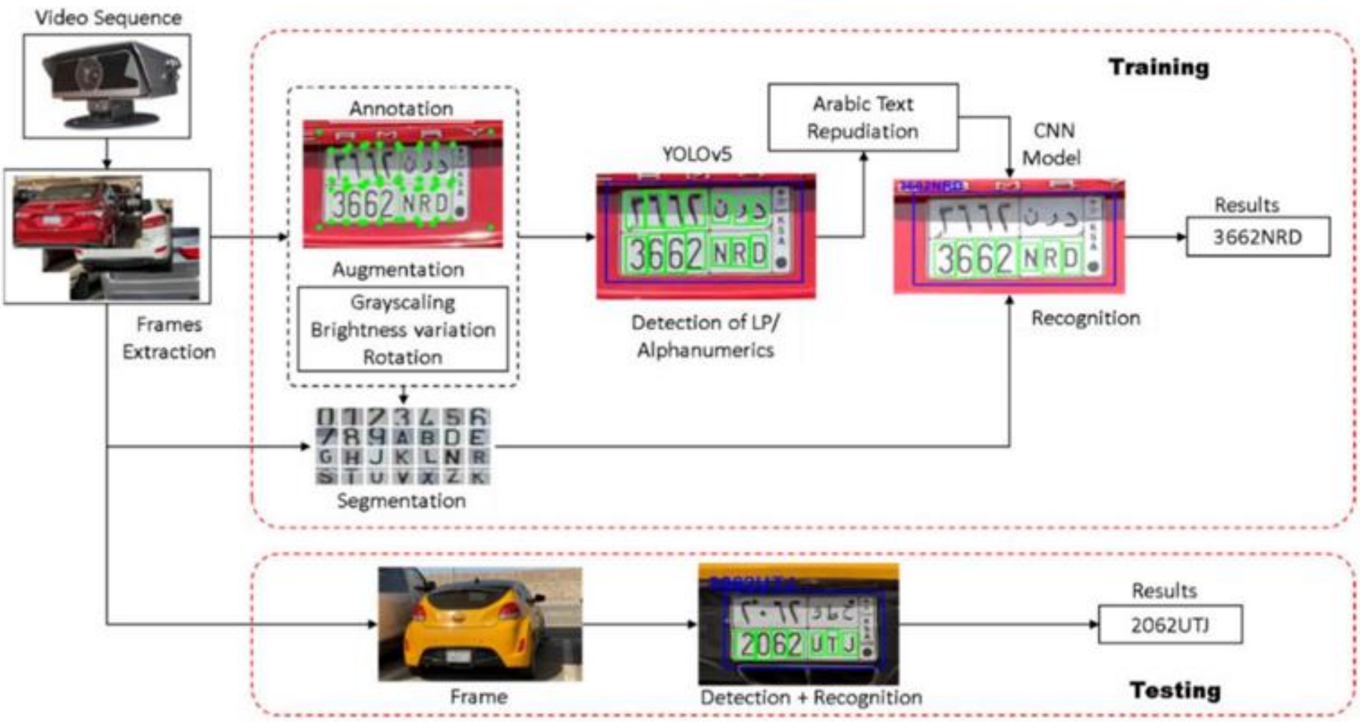


Fig: 3.1. Analysis of the System

The dataset was taken from the Kaggle repository which contain 432 images out of which 402 belongs to train category and 30 are test category images. All the images are of jpeg format.

From the above fig 3.1 we can clearly see the entire process of the system .Two experiments have been done to test the performance of the proposed detection and recognition system. The first experiment was done on various License Plates using real traffic video sequences collected from different situations in different lighting conditions. The second experiment was done using the dataset with car images, the AOLP dataset. The cars dataset consists of 126 images with a resolution of 890x592 pixels taken during the day from complex outside scenes, each of which contains only one vehicle.

For the classification, we have used a pre-trained CNN model, so that we do not need a large training dataset for successful classification.

3.3 Data preprocessing:

Data preprocessing to be done for the following Neural Network Models.

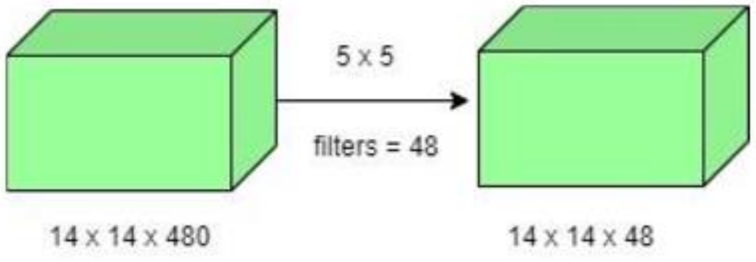
About Some Neural Network Models:

Google Net Model: Google Net (or Inception V1) was proposed by research at Google (with the collaboration of various universities) in 2014 in the research paper titled "Going Deeper with Convolutions". This architecture was the winner at the ILSVRC 2014 image classification challenge. It has provided a significant decrease in error rate as compared to previous winners AlexNet (Winner of ILSVRC 2012) and ZF-Net (Winner of ILSVRC 2013) and significantly less error rate than VGG (2014 runner up). This architecture uses techniques such as *1×1* convolutions in the middle of the architecture and global average pooling.

Features of GoogleNet:

The GoogLeNet architecture is very different from previous state-of-the-art architectures such as AlexNet and ZF-Net. It uses many different kinds of methods such as *1×1* convolution and global average pooling that enables it to create deeper architecture. In the architecture, we will discuss some of these methods:

- ***1×1* convolution :** The inception architecture uses *1×1* convolution in its architecture. These convolutions used to decrease the number of parameters (weights and biases) of the architecture. By reducing the parameters we also increase the depth of the architecture. Let's look at an example of a *1×1* convolution below:
- For Example, If we want to perform *5×5* convolution having 48 filters without using *1×1* convolution as intermediate:



- Total Number of operations : *(14 x 14 x 48) x (5 x 5 x 480) = 112.9 M* • With 1×1 convolution :

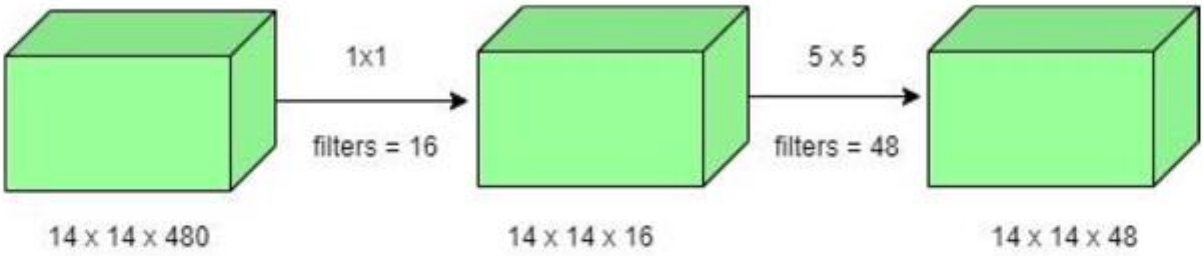


Fig: 3.2 1*1 convolution layer of GoogleNet

- *(14 x 14 x 16) x (1 x 1 x 480) + (14 x 14 x 48) x (5 x 5 x 16) = 1.5M + 3.8M = 5.3M* which is much smaller than 112.9M.

Global Average Pooling:

In the previous architecture such as AlexNet, the fully connected layers are used at the end of the network. These fully connected layers contain the majority of parameters of many architectures that causes an increase in computation cost.

In GoogLeNet architecture, there is a method called global average pooling is used at the end of the network. This layer takes a feature map of *7×7* and averages it to *1×1*. This also decreases the number of trainable parameters to 0 and improves the top-1 accuracy by 0.6%.

Inception Module:

The inception module is different from previous architectures such as AlexNet, ZF-Net. In this architecture, there is a fixed convolution size for each layer. In the Inception module *1×1, 3×3, 5×5* convolution and *3×3* max pooling performed in a parallel way at the input and the output of these are stacked together to generated final output which is shown in fig 3.3. The idea behind that convolution filters of different sizes.

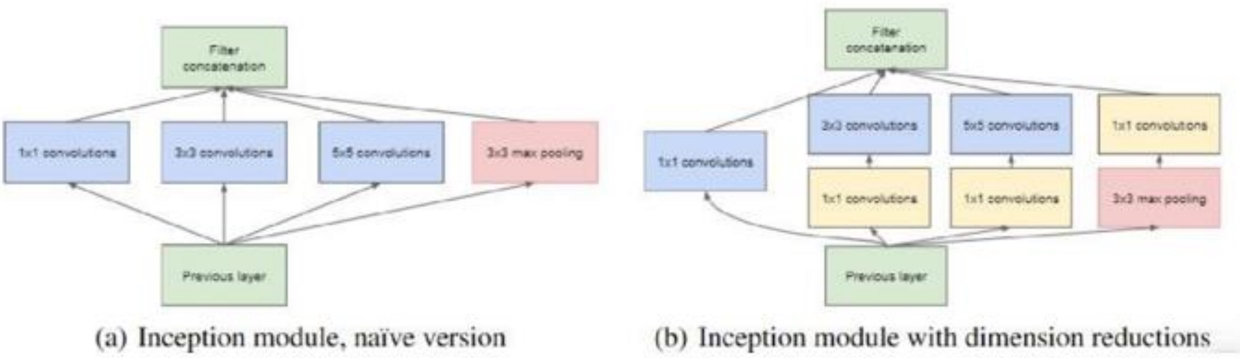


Fig: 3.3 Inception Module of GoogleNet

Auxiliary Classifier for Training:

Inception architecture used some intermediate classifier branches in the middle of the architecture, these branches are used during training only. These branches consist of a 5×5 average pooling layer with a stride of 3, a *1×1* convolutions with *128* filters, two fully connected layers of 1024 outputs and 1000 outputs and a softmax classification layer. The generated loss of these layers added to total loss with a weight of 0.3. These layers help in combating gradient vanishing problem and also provide regularization.

Model Architecture:
Below is Layer by Layer architectural details of GoogeNet.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Fig: 3.4 Architecture of GoogleNet with Layered Details

The overall architecture is 22 layers deep. The architecture was designed to keep computational efficiency in mind. The idea behind that the architecture can be run on individual devices even with low computational resources. The architecture also contains two auxiliary classifier layer connected to the output of Inception (4a) and Inception (4d) layers. The architectural details of auxiliary classifiers as follows:

- An average pooling layer of filter size 5×5 and stride 3.
- A 1×1 convolution with 128 filters for dimension reduction and ReLU activation.
- A fully connected layer with 1025 outputs and ReLU activation
- Dropout Regularization with dropout ratio = 0.7
- A softmax classifier with 1000 classes output similar to the main softmax classsifier.

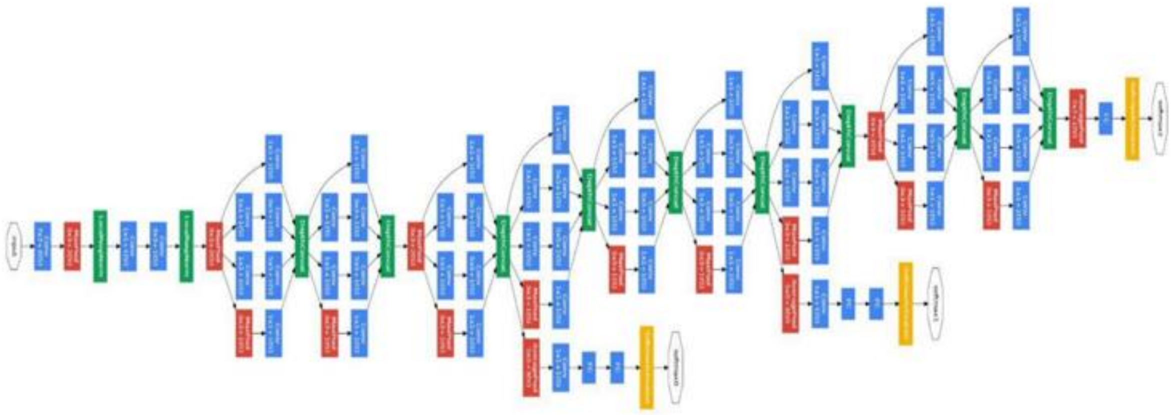


Fig: 3.5 Architecture of GoogleNet

This architecture takes image of size *224 x 224* with RGB color channels. All the convolutions inside this architecture uses Rectified Linear Units (ReLU) as their activation functions.

Results:

GoogleNet was the winner at ILSRVRC 2014 taking 1st place in both classification an detection task. It has top-5 error rate of 6.67% in classification task. An ensemble of 6 GoogLeNets gives 43.9 % mAP on ImageNet test set.

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Fig 3.6 Results of GoogleNet at ILSRVRC

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Fig: 3.7 Results of GoogleNet at ILSRVRC with details of data

Residual Networks:

After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases.

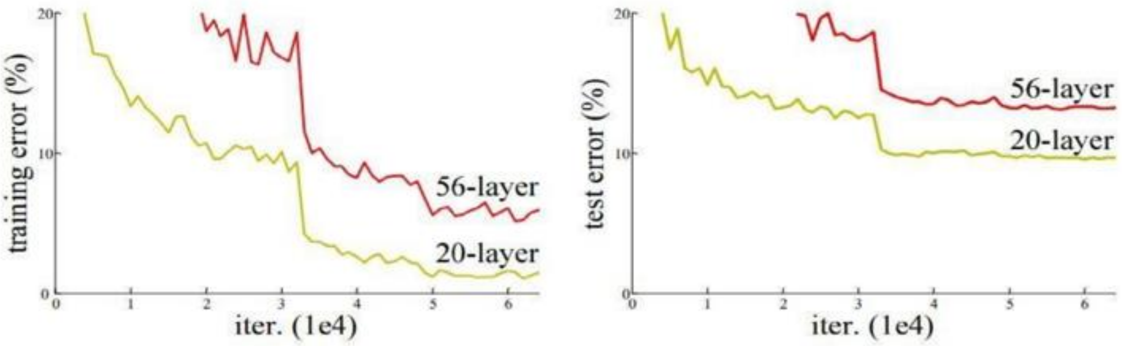


Fig: 3.8 Residual Networks with training and testing rates

In the above plot, we can observe that a 56-layer CNN gives more error rate on both training and testing dataset than a 20-layer CNN architecture, If this was the result of over fitting, then we should have lower training error in 56-layer CNN but then it also has higher training error. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient. ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

Residual Block:

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. In this network we use a technique called ***skip connections***. The skip connection skips training from a few layers and connects directly to the output.

The approach behind this network is instead of layers learn the underlying mapping, we allow network fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit,

$F(x) := H(x) - x$ which gives $H(x) := F(x) + x$.

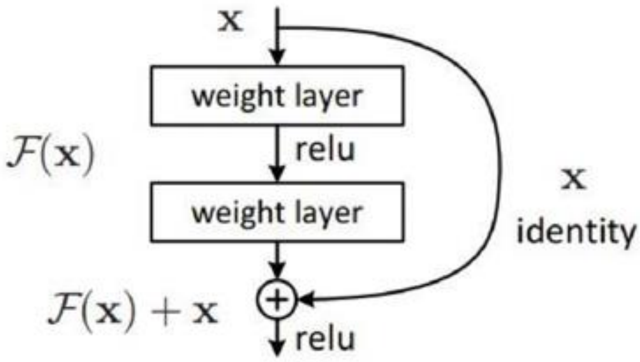


Fig: 3.9 Residual Block with ReLu

The advantage of adding this type of skip connection is because if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training very deep neural network without the problems caused by vanishing/exploding gradient. The authors of the paper experimented on 100-1000 layers on CIFAR-10 dataset.

There is a similar approach called "highway networks", these networks also uses skip connection. Similar to LSTM these skip connections also uses parametric gates. These gates determine how much information passes through the skip connection. This architecture however has not provide accuracy better than ResNet architecture.

Network Architecture:

This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into residual network.

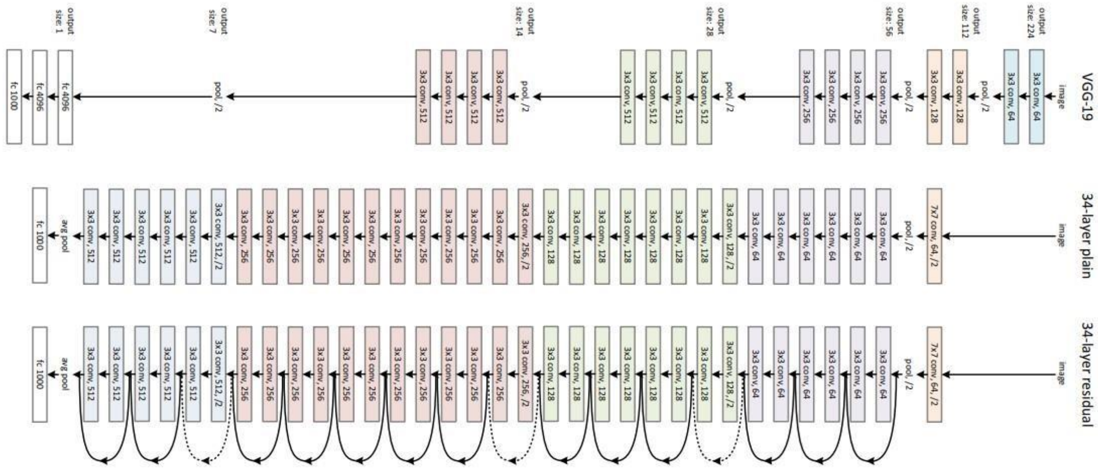


Fig: 3.10 VGG-34 Architecture

VGG-16 Model:

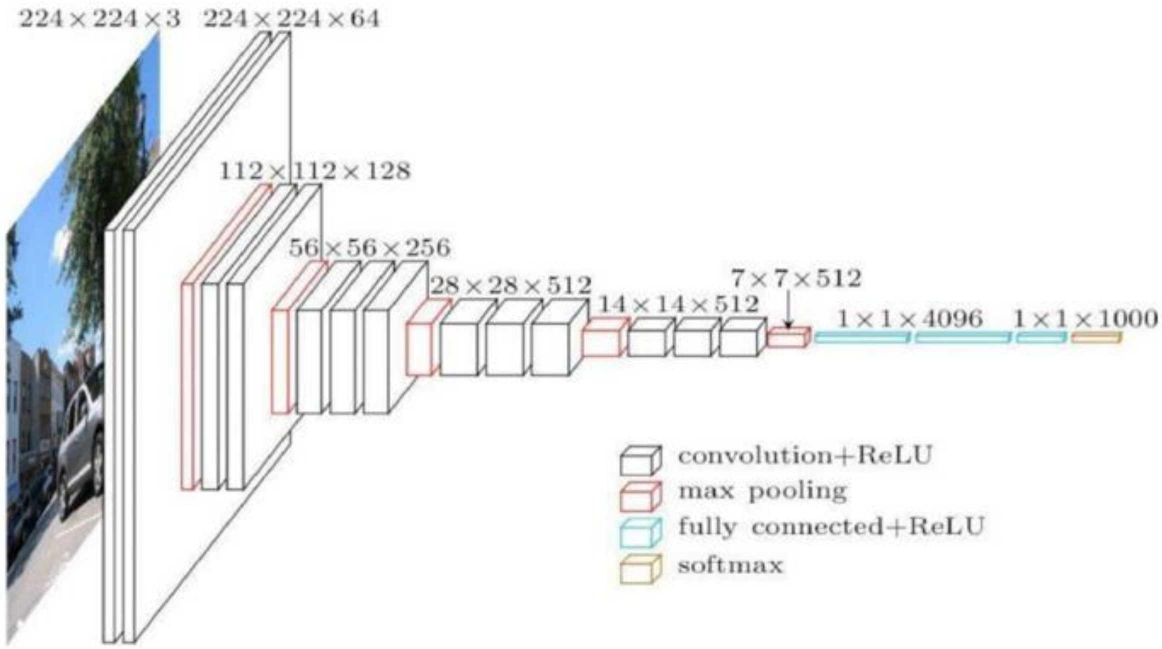


Fig: 3.11 VGG-16 Architecture

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA Titan Black GPU's.

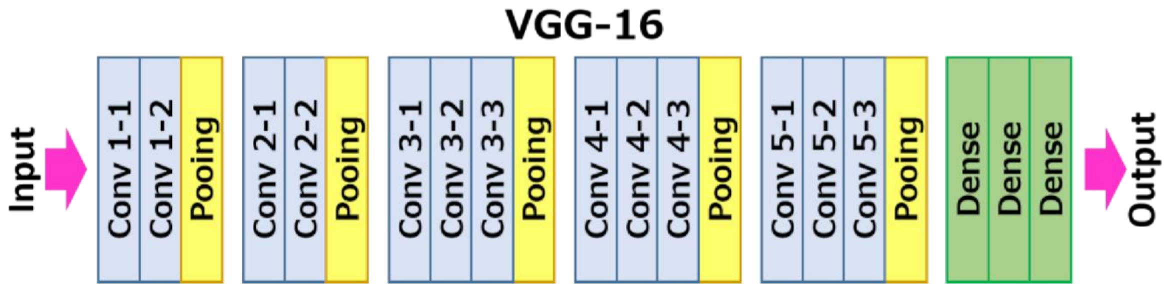


Fig: 3 .12 VGG-16 Overview of layer

The Architecture

The input to cov1 layer is of fixed size 224 x 224 RGB image. The image is passed through a stack of convolutional (conv.) layers, where the filters were used with a very small receptive field: 3×3 (which is the smallest size to capture the notion of left/right, up/down, center). In one of the configurations, it also utilizes 1×1 convolution filters, which can be seen as a linear transformation of the input channels (followed by non-linearity). The convolution stride is fixed to 1 pixel; the spatial padding of conv. layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 conv. layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the conv. layers (not all the conv. layers are followed by max-pooling). Maxpooling is performed over a 2×2 pixel window, with stride 2.

Three Fully-Connected (FC) layers follow a stack of convolutional layers (which has a different depth in different architectures): the first two have 4096 channels each, the third performs 1000-way ILSVRC classification and thus contains 1000 channels (one for each class). The final layer is the soft-max layer. The configuration of the fully connected layers is the same in all networks.

All hidden layers are equipped with the rectification (ReLU) non-linearity. It is also noted that none of the networks (except for one) contain Local Response Normalisation (LRN), such normalization does not improve the performance on the ILSVRC dataset, but leads to increased memory consumption and computation time.

Configurations

The ConvNet configurations are outlined in figure 2. The nets are referred to their names (A-E). All configurations follow the generic design present in architecture and differ only in the depth:

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig: 3.13 ConvNet Configuration

from 11 weight layers in the network A (8 conv. and 3 FC layers) to 19 weight layers in the network E (16 conv. and 3 FC layers). The width of conv. layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

AlexNet Model:

AlexNet is the name of a convolutional neural network which has had a large impact on the field of machine learning, specifically in the application of deep learning to machine vision. It famously won the 2012 ImageNet LSVRC-2012 competition by a large margin (15.3% VS 26.2% (second place) error rates). The network had a very similar architecture as LeNet by Yann LeCun et al but was deeper, with more filters per layer, and with stacked convolutional layers. It consisted of 11×11, 5×5,3×3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer. AlexNet was trained for 6 days simultaneously on two Nvidia Geforce GTX 580 GPUs which is the reason for why their network is split into two pipelines.

Key Points

- Relu activation function is used instead of Tanh to add non-linearity. It accelerates the speed by 6 times at the same accuracy.
- Use dropout instead of regularisation to deal with overfitting. However, the training time is doubled with the dropout rate of 0.5.
- Overlap pooling to reduce the size of the network. It reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively.

The Architecture

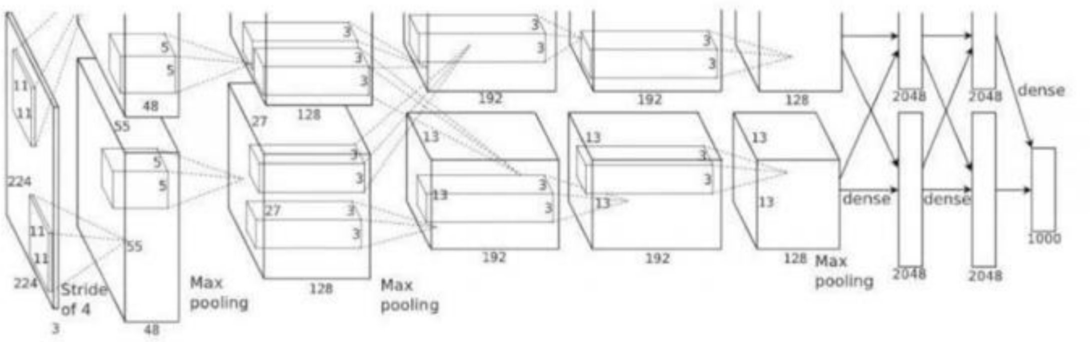


Fig: 3.14 AlexNet Model

AlexNet contains eight layers with weights; the first five are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. The network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the logprobability of the correct label under the prediction distribution. The kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer which reside on the same GPU. The kernels of the third convolutional layer are connected to all kernel maps in the second layer. The neurons in the fully-connected layers are connected to all neurons in the previous layer.

In short, AlexNet contains 5 convolutional layers and 3 fully connected layers. Relu is applied after very convolutional and fully connected layer. Dropout is applied before the first and the second fully connected year. The network has 62.3 million parameters and needs 1.1 billion computation units in a forward pass. We can also see convolution layers, which accounts for 6% of all the parameters, consumes 95% of the computation.

Training

AlexNet takes 90 epochs which were trained for 6 days simultaneously on two Nvidia Geforce GTX 580 GPUs which is the reason for why their network is split into two pipelines. SGD with learning rate 0.01, momentum 0.9 and weight decay 0.0005 is used. Learning rate is divided

$$\begin{aligned} v_{i+1} &:= 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i} \\ w_{i+1} &:= w_i + v_{i+1} \end{aligned}$$

by 10 once the accuracy plateaus. The learning rate is decreased 3 times during the training process.

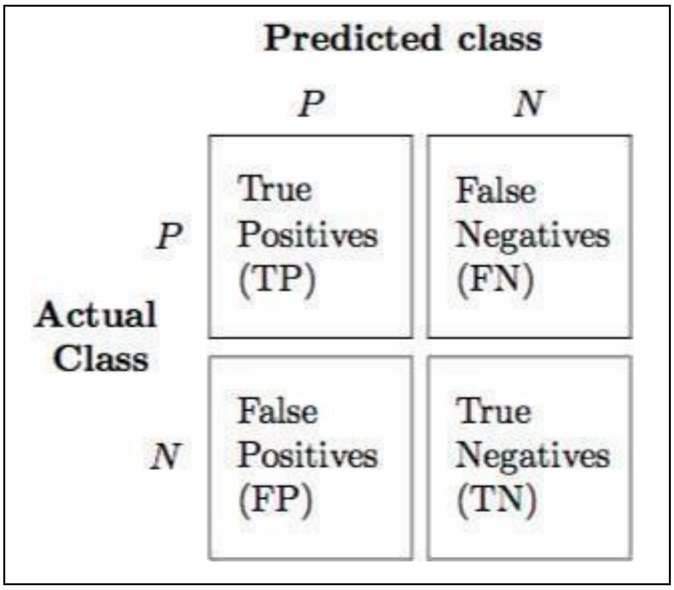
The update rule for *w* was where *i* is the iteration index, *v* is the momentum variable and *epsilon* is the learning rate. Equal learning rate for all layers, which was adjusted manually throughout training. The heuristic which was followed was to divide the learning rate by 10 when the validation error rate stopped improving with the current learning rate.

Result

The network achieves top-1 and top-5 test set error rates of 37.5% and 17.0%. The best performance achieved during the ILSVRC-2010 competition was 47.1% and 28.2% with an approach that averages the predictions produced from six sparse-coding models trained on different features, and since then the best published results are 45.7% and 25.7% with an approach that averages the predictions of two classifiers trained on Fisher Vectors (FVs) computed from two types of densely sampled features.

Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.



A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class.

A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease. TPR= tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

TNR= tn / (tn + fp) =1-FPR

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the patient actually has the disease.

PPV= tp / (tp + fp)

Negative predictive value (NPV)

If the test result is negative what is the probability that the patient does not have disease.

NPV= tn/ (tn+fn)

Miss rate or false negative rate (FNR) :

It is the proportion of the individuals with a known positive condition for which the test result is negative.

FNR=fn/ (fp+tn)

Fall-out or false positive rate (FPR) :

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

FPR=fp/ (fp+tn)

False discovery rate (FDR)

It is the proportion of all the people identified as having the disease who do not have the disease.

FDR=fp/ fp+tp

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true condition is positive.

FOR=fn/ (fn+tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.

ACC= (tp+tn) / (tp+tn+fp+fn)

F1 score

It is the harmonic mean of precision and sensitivity sensitivity

F1=2tp/ (2tp+fp +fn)

4. DESIGN

The most important part of this system is the software design. The software design uses series of image processing techniques which are implemented in Android mobile platform which is supported minimum API 5 or android 2.0 (cupcake). The ANPR algorithm designed in this paper is roughly divided into three parts:

- Capture number plate image
- Image filtering
- Segmentation of the number plate image
- Recognize the numbers plate image using OCR algorithm.

The design of this model is as follows:

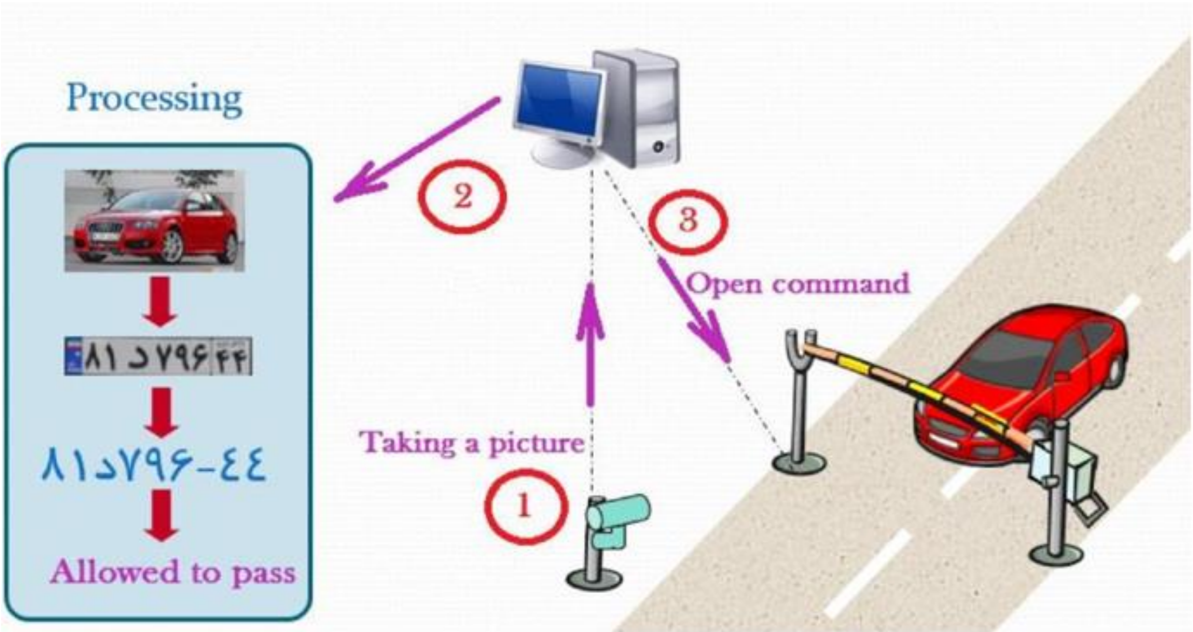


Fig: 4.1 Overview of the model

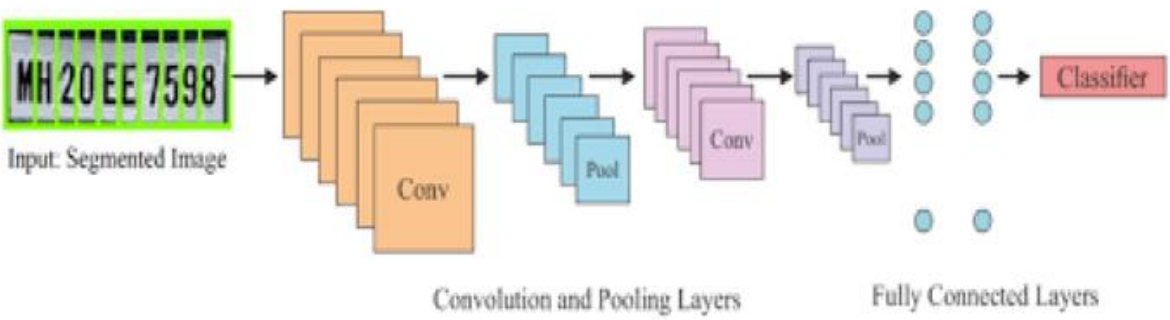


Fig: 4.2 Convolutional Neural Network (CNN)

UML Diagrams

USE CASE DIAGRAM

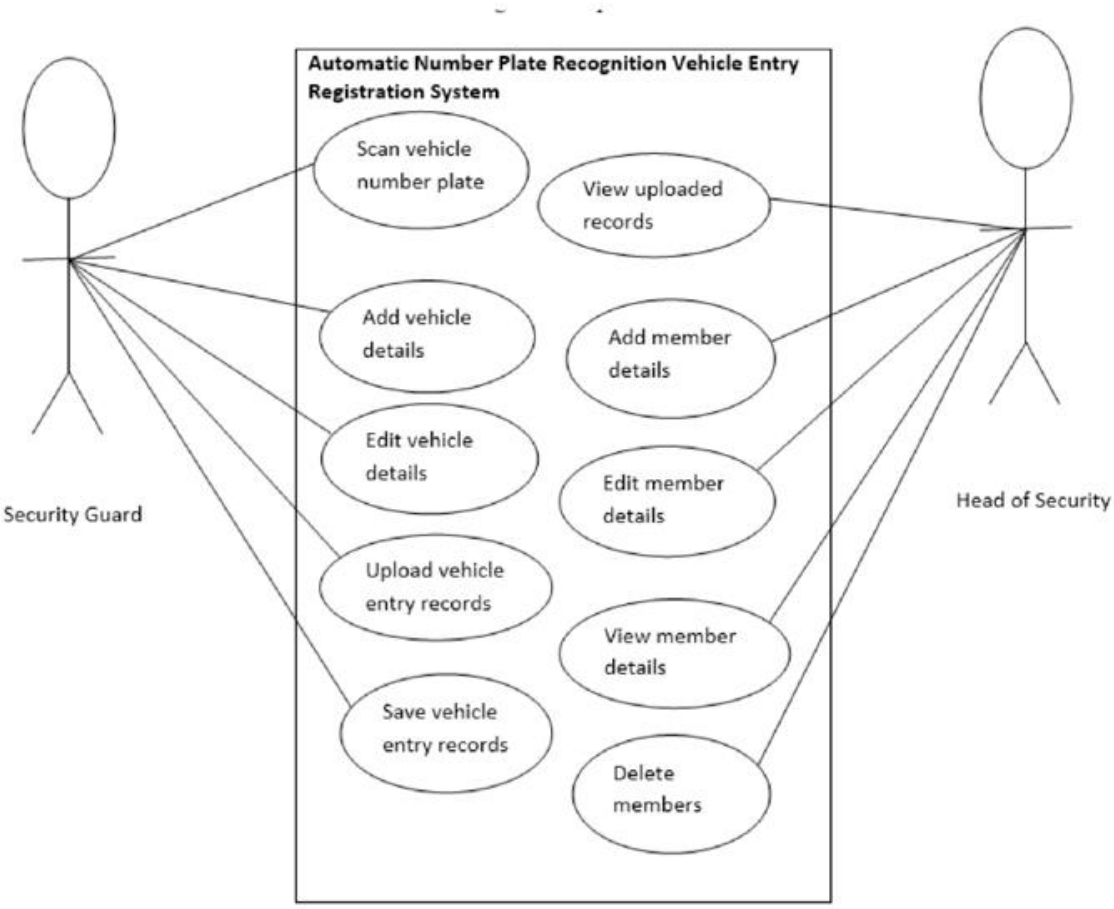


Fig 4.3 UML DIAGRAM

EXPLANATION

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the case and will often be accompanied by other types of diagrams as well.

ACTIVITY DIAGRAM

EXPLANATION

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. It captures the dynamic behavior of the system. Activity is a particular operation of the system.

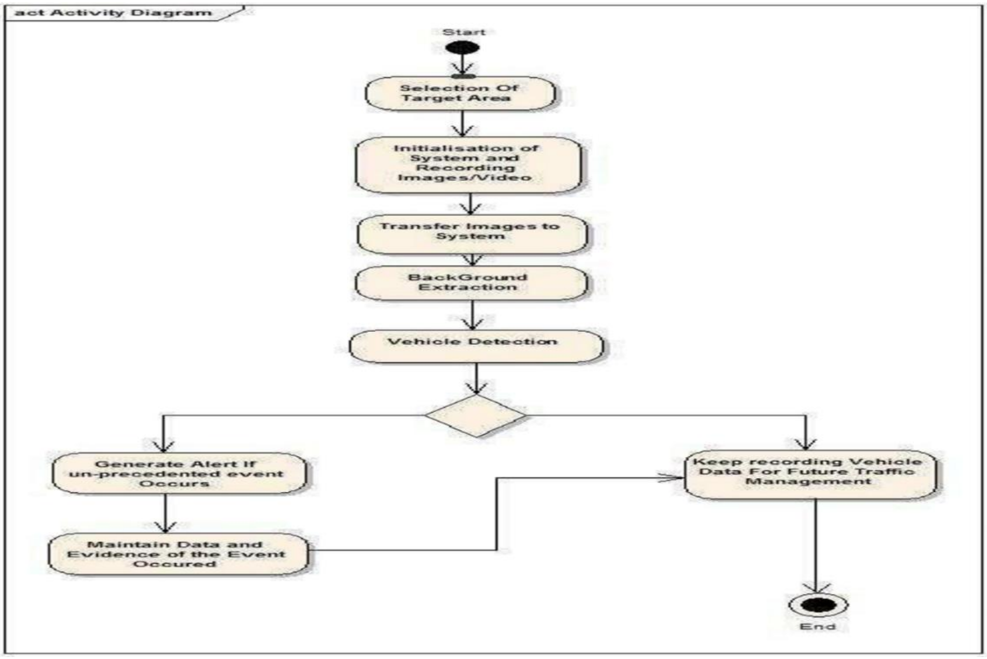


Fig 4.4

SEQUENCE DIAGRAM

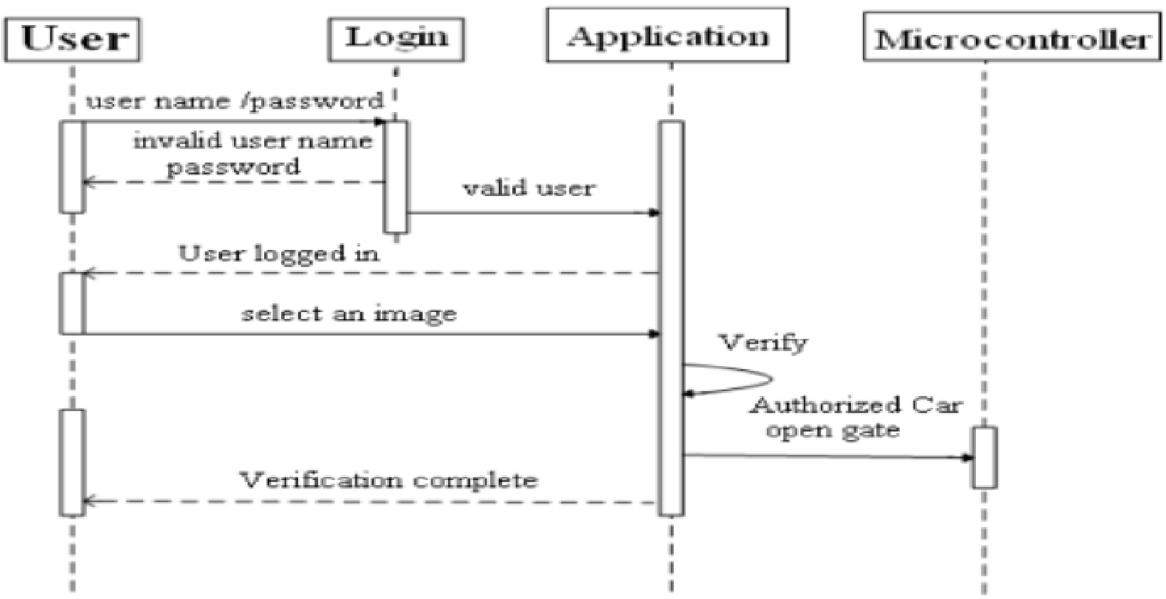


Fig 4.5 Sequence Diagram

EXPLANATION

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

CLASS DIAGRAM

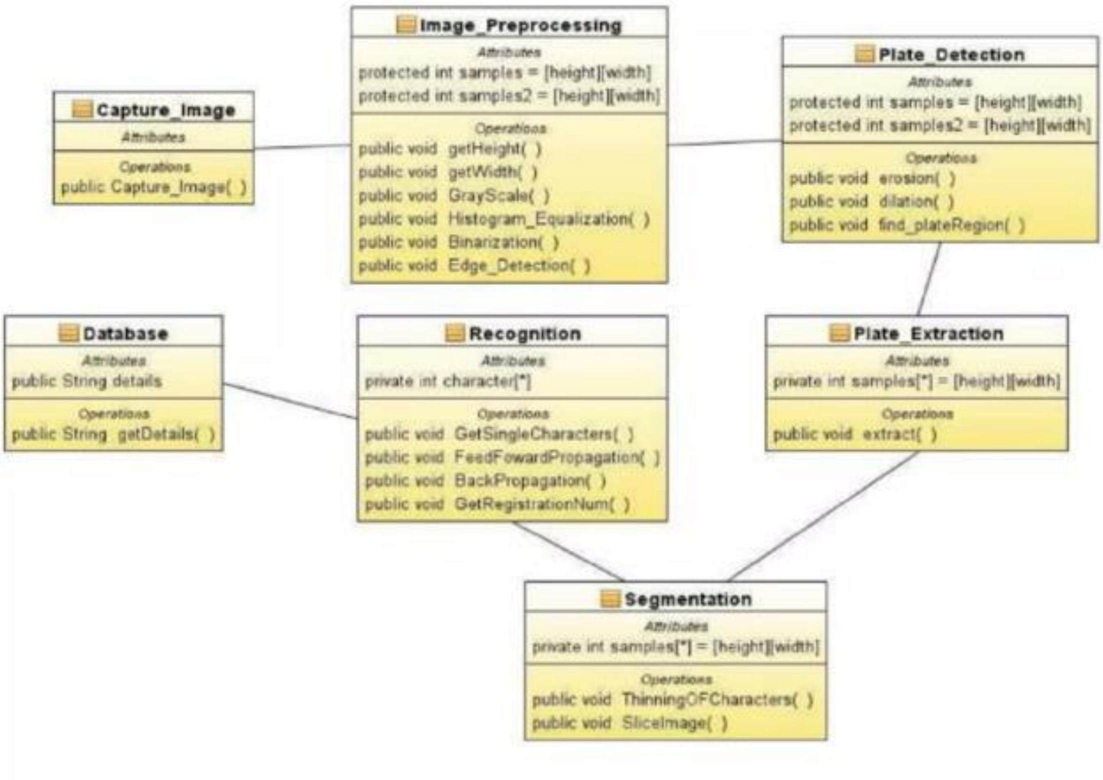


Fig 4.6 Class Diagram

EXPLANATION

The class diagram gives a clear picture of all the processes involved in the background in order to carry out the recognition process. It shows all the classes that happens in the background and as well gives a clear relationship on how they relates with one another to help recognize the characters in the plates at the end of the day. The class diagram contains of all the attributes involved in each class or method. It also gives a high clear idea towards the entire processing of the image, how the image is being processed to cater for recognizing the characters.

5. IMPLEMENTATION

CODE

```
!pip install pytesseract
!pip install easyocr
import os
import cv2
import numpy as np
import pandas as pd
import tensorflow as tf
import pytesseract as pt
import plotly.express as px
import matplotlib.pyplot as plt
import xml.etree.ElementTree as xet
from glob import glob
from skimage import io
from shutil import copy
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import TensorBoard
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import InceptionResNetV2
from tensorflow.keras.layers import Dense, Dropout, Flatten, Input
from tensorflow.keras.preprocessing.image import load_img, img_to_array
path = glob("images/*.xml")
labels_dict = dict(filepath=[],xmin=[],xmax=[],ymin=[],ymax=[])
for filename in path:
info = xet.parse(filename)
root = info.getroot()
member_object = root.find('object')
labels_info = member_object.find('bndbox')
xmin = int(labels_info.find('xmin').text)
xmax = int(labels_info.find('xmax').text)
ymin = int(labels_info.find('ymin').text)
ymax = int(labels_info.find('ymax').text)
labels_dict['filepath'].append(filename)
labels_dict['xmin'].append(xmin)
labels_dict['xmax'].append(xmax)
labels_dict['ymin'].append(ymin)
labels_dict['ymax'].append(ymax)
df = pd.DataFrame(labels_dict)
df.to_csv('labels.csv',index=False)
df.head()
```

VERIFY THE DATA

```
file_path = image_path[87] #path of our image N2.jpeg
img = cv2.imread(file_path) #read the image
# xmin-1804/ymin-1734/xmax-2493/ymax-1882
img = io.imread(file_path) #Read the image
fig = px.imshow(img)
fig.update_layout(width=600, height=500, margin=dict(l=10, r=10, b=10,
t=10),xaxis_title='Figure
8 - N2.jpeg with bounding box')
fig.add_shape(type='rect',x0=1804, x1=2493, y0=1734, y1=1882, xref='x',
yref='y',line_color='cyan')
```

DATA PROCESSING

READ DATA

```
#Targeting all our values in array selecting all columns
labels = df.iloc[:,1:].values
data = []
output = []
for ind in range(len(image_path)):
image = image_path[ind]
img_arr = cv2.imread(image)
h,w,d = img_arr.shape
# Prepprocesing
load_image = load_img(image,target_size=(224,224))
load_image_arr = img_to_array(load_image)
norm_load_image_arr = load_image_arr/255.0 # Normalization
# Normalization to labels
xmin,xmax,ymin,ymax = labels[ind]
nxmin,nxmax = xmin/w,xmax/w
nymin,nymax = ymin/h,ymax/h
label_norm = (nxmin,nxmax,nymin,nymax) # Normalized output
# Append
data.append(norm_load_image_arr)
output.append(label_norm)
```

SPLIT TRAIN AND TEST SET

```
# Convert data to array
X = np.array(data,dtype=np.float32)
y = np.array(output,dtype=np.float32)
# Split the data into training and testing set using sklearn.
x_train,x_test,y_train,y_test = train_test_split(X,y,train_size=0.8,random_state=0)
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

DEEP LEARNING FOR OBJECT DETECTION

INCEPTION-RESNET-V2 MODEL BUILDING

```
inception_resnet = InceptionResNetV2(weights="imagenet",include_top=False,
input_tensor=Input(shape=(224,224,3)))
# ---------------------
headmodel = inception_resnet.output
headmodel = Flatten()(headmodel)
headmodel = Dense(500,activation="relu")(headmodel)
headmodel = Dense(250,activation="relu")(headmodel)
headmodel = Dense(4,activation='sigmoid')(headmodel)
# ---------- model
model = Model(inputs=inception_resnet.input,outputs=headmodel)
# Complie model
model.compile(loss='mse',optimizer=tf.keras.optimizers.Adam(learning_rate=1e-4))
```

INCEPTION-RESNET-V2 TRAINING AND SAVE

```
tfb = TensorBoard('object_detection')
history = model.fit(x=x_train,y=y_train,batch_size=10,epochs=180,
validation_data=(x_test,y_test),callbacks=[tfb])
model.save('./object_detection.h5')
```

PIPELINE OBJECT DETECTION MODEL

MAKE PREDICTIONS

```
# Load model
model = tf.keras.models.load_model('./object_detection.h5')
print('Model loaded Sucessfully')
path = 'TEST\TEST1.png'
image = load_img(path) # PIL object
image = np.array(image,dtype=np.uint8) # 8 bit array (0,255)
image1 = load_img(path,target_size=(224,224))
image_arr_224 = img_to_array(image1)/255.0  # Convert into array and get the normalized
output
# Size of the orginal image
h,w,d = image.shape
print('Height of the image =',h)
print('Width of the image =',w)
fig = px.imshow(image)
fig.update_layout(width=700, height=500, margin=dict(l=10, r=10, b=10, t=10),
xaxis_title='Figure
13 - TEST Image')
image_arr_224.shape
image_arr_224.shapetest_arr = image_arr_224.reshape(1,224,224,3)
test_arr.shape
```

DE-NORMALIZE THE OUTPUT

```
# Make predictions
coords = model.predict(test_arr)
cords
# Denormalize the values
denorm = np.array([w,w,h,h])
coords = coords * denorm
cords
```

EXTRACT OUTPUT (TEXT)

```
reader = easyocr.Reader(['en'])
output = reader.readtext('/content/drive/MyDrive/t_i.jpg')
output
```

WEBSITE

app.py

```
from flask import Flask, render_template, request
import easyocr

app = Flask(__name__)

# Initialize EasyOCR reader
reader = easyocr.Reader(['en'])

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/detect')
def index():
    return render_template('index.html')

@app.route('/upload', methods=['POST'])
def upload():
    if 'image' not in request.files:
        return render_template('index.html', error='No file part')

    file = request.files['image']

    if file.filename == '':
        return render_template('index.html', error='No Selected File.')

    if file:
        image = file.read()
        extracted_text = extract_text(image)
```

```
        return render_template('index.html',
text=extracted_text,
filename=file.filename,image_data=image)

def extract_text(image):
    result = reader.readtext(image)
    extracted_text = '\n'.join([text[1] for text in result])
    return extracted_text

if __name__ == '__main__':
    app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Number Plate Prediction</title>
    <style>
        body {
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
            display: flex;
            flex-direction: column;
            align-items: center;
            background-color: #333;
        }

        .navigation-container {
            width: 100%;
            background-color:#333;
            padding: 10px;
            display: flex;
            justify-content: space-between;
            align-items: center;
            border-bottom: 3px solid rgb(200, 166, 166);
        }

        .navigation {
            margin-right: 20px;
        }

        .navigation-container h1 {
```

```
    color:white;
    margin-left:20px;
}

.nav-link {
    text-decoration: none;
    margin-left: 10px; /* Increase space between buttons */
    margin-right: 10px;
    padding: 8px 20px; /* Increased padding for the buttons */
    background-color: #555;
    border: none;
    border-radius: 20px; /* Curved edges */
    color: #fff;
    cursor: pointer;
    letter-spacing: 1px; /* Add spacing between letters */
    border-bottom: 2px solid white;
    margin-bottom:20px;
    font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.content {
    margin-top: 20px;
    padding: 20px;
    display: flex;
    flex-direction: column;
    align-items: center;
}

.upload-section {
    text-align: center;
    margin-bottom: 20px;
    color:white;
    padding-top:120px;
}

.file-input {
    display: none;

}

.file-label, .proceed-button {
    padding: 10px 20px;
    background-color: #555;
    color: white;
    border: #555;
```

```
        border-radius: 20px;
        cursor: pointer;
        border:15px;
        border-bottom:3px solid white;

    }

    .file-label:hover, .proceed-button:hover {
        background-color: red;
    }

    .proceed-button{
        font-size:medium;
        margin-left: 25px;
    }
    .nav-link:hover{
        background-color: red;
    }
    .result {
       text-align:center;
       padding-top:10px;
       color:white;
    }

    .result:hover{
        color:red;
    }

    .forecast-heading:hover{
        color:Black;
    }

    /* Your existing CSS styles end here */

    /* Additional CSS for image preview */
    #image-preview-container {
        margin-top: 20px;
    }
    #image-preview {
        max-width: 100%;
        max-height: 300px;
        margin-top: 10px;
    }
  </style>
</head>
```

```
<body>
    <div class="navigation-container">
        <h1 class="forecast-heading">Forecast of the ALPR</h1>
        <div class="navigation">
            <a href="/" class="nav-link">Home</a>
            <a href="/detect" class="nav-link">Detect</a>
        </div>
    </div>
    <div class="content">
        <div class="upload-section">
            <p><b>Upload your document. Let our system handle the rest upon clicking
Proceed.</b></p>
            <form action="/upload" method="post" enctype="multipart/form-data">
                <label class="file-label" for="file-input">Choose a file</label>
    <input type="file" name="image" id="file-input" class="file-input"
onchange="previewImage(this)">
                <span id="file-name"></span> <!-- Placeholder for displaying file name -->
                <button type="submit" class="proceed-button">Proceed</button>
            </form>
            <!-- Container to display the selected image -->
            <div id="image-preview-container">
                <img id="image-preview" src="#" alt="">
            </div>
        </div>

        {% if error %}
            <div class="result">
                <p>{{ error }}</p>
            </div>
        {% endif %}
        {% if text %}
            <div class="result">
                <h2>Number Plate : {{ text }}</h2>
            </div>
        {% endif %}
    </div>
    <script>
        // Function to update file name display and preview the selected image
        function previewImage(input) {
            var fileNameDisplay = document.getElementById("file-name");
            if (input.files.length > 0) {
                fileNameDisplay.innerText = '' + input.files[0].name;
                var reader = new FileReader();
                reader.onload = function(e) {
                    document.getElementById("image-preview").setAttribute("src", e.target.result);
```

```
            }
            reader.readAsDataURL(input.files[0]);
        } else {
            fileNameDisplay.innerText = "";
        }
    }
</script>
</body>
</html>
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Home Page</title>
<style>
  body, html {
    margin: 0;
    padding: 0;
    height: 100%;
    overflow: hidden;
    background-color: #333; /* Change background color as needed */
    color: #fff; /* Change text color as needed */
    font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  }
  #background {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-image: url('background_image.jpg'); /* Change 'background_image.jpg' to
the path of your background image */
    background-size: cover;
    background-position: center;
    animation: moveBackground 20s linear infinite;
  }
  @keyframes moveBackground {
    from { background-position: 0 0; }
    to { background-position: 100% 0; }
  }
  #content {
    position: absolute;
```

```
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    text-align: center;
    letter-spacing: 0.85px;
    font-family:'Times New Roman', Times, serif;
    margin-top:30px;
}
#image-container {
    display: flex;
    justify-content: center;
    align-items: center;
}
.image {
    width: 200px;
    height: 200px;
    margin: 10px;
    filter: brightness(130%); /* Adjust brightness as needed */
    border-radius: 15px; /* Add curved edges */
    border:3px solid black;
}
h1, p {
    margin: 10px 0;
    font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.navigation-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    position: absolute;
    top: 20px;
    left: 20px;
    right: 20px;
    border-bottom: 3px solid rgb(200, 166, 166);
}
.navigation button {
    margin-left: 10px; /* Increase space between buttons */
    margin-right: 10px;
    padding: 8px 20px; /* Increased padding for the buttons */
    background-color: #555;
    border: none;
    border-radius: 20px; /* Curved edges */
    color: #fff;
    cursor: pointer;
    letter-spacing: 1px; /* Add spacing between letters */
```

```
        border-bottom: 2px solid white;
        margin-bottom:20px;
        font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }
    .alpr-heading {
        margin-right: auto;
        color: white; /* Push ALPR heading to the left */
        margin-bottom:30px;
    }
    .alpr-heading:hover{
        color: black;
    }
    a {
        text-decoration: none;
        margin-left: 10px; /* Increase space between buttons */
        margin-right: 10px;
        padding: 8px 20px; /* Increased padding for the buttons */
        background-color: #555;
        border: none;
        border-radius: 20px; /* Curved edges */
        color: #fff;
        cursor: pointer;
        letter-spacing: 1px; /* Add spacing between letters */
        border-bottom: 2px solid white;
        margin-bottom:20px;
        font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }
    a:hover{
background-color: red;
    }
</style>
</head>
<body>
    <div id="background"></div>
    <div id="content">
        <h1>Welcome to our website!</h1>
        <p>We specialize in developing state-of-the-art Automatic License Plate Recognition
(ALPR) systems.</p>
        <br>
        <div id="image-container">
            <img src="https://www.universeoptics.com/wp-
content/uploads/Traffic_Cameras.jpg" class="image"> <!-- Change 'image1.jpg',
'image2.jpg', etc. to the paths of your images -->
            <img
src="https://www.researchgate.net/publication/355102486/figure/fig4/AS:10766404697866
```

```
26@1633702499803/The-effect-of-vehicle-license-plate-recognition-from-the-perspective-of-traffic.ppm" class="image">
            <img src="https://media.istockphoto.com/id/1336889543/video/video-surveillance-on-the-motorway-license-plate-reading-camera.jpg?s=640x640&k=20&c=_cwc83PmbvR67RKS5atCMvP9rl43xdhukB3eZBbbylg=" class="image">
            <img src="https://www.efkonindia.com/images/inner-header-products-rlvd.jpg" class="image">
            <img src="https://intertraff.com/wp-content/uploads/2023/02/VMMR.jpg" class="image">
            <img src="https://5.imimg.com/data5/RC/NQ/MY-29870930/traffic-camera-500x500.jpg" class="image">
        </div>
        <p>The Automatic License Plate Recognition (ALPR) system revolutionizes vehicle monitoring and management across various industries and applications. Its versatility and precision make it indispensable in law enforcement, enabling authorities to swiftly identify vehicles of interest, track stolen vehicles, and enforce traffic regulations effectively. In the realm of parking management, ALPR systems streamline operations by automating entry and exit processes, optimizing space utilization, and enhancing security. Moreover, ALPR technology plays a crucial role in toll collection, facilitating seamless transactions and minimizing congestion at toll booths. Beyond these core applications, ALPR systems find utility in diverse settings, including border control, vehicle tracking for logistics and fleet management, and even in smart cities initiatives for traffic monitoring and urban planning. With its ability to enhance security, efficiency, and convenience, the ALPR system stands as a cornerstone of modern transportation management solutions, driving innovation and effectiveness across a multitude of sectors.</p>
        <br>
    </div>
    <div class="navigation-container">
        <h1 class="alpr-heading">Automatic License Number Plate Recognition System</h1>
        <div class="navigation">
            <a href="/">Home</a>
            <a href="/predict">Predict</a>
        </div>
    </div>
</body>
</html>
```

6. RESULT ANALYSIS

The system evaluates the performance of four deep neural networks. Finding the best object detection method is the main objective of this study. i.e., a best object detection algorithm can accurately find out the license plate region so that the license plate number can be extracted successfully. The system automatically detects the license plate using three deep learning algorithms.

Fig 6.1 Number Plate Detection

The detected number plates are then processed for character segmentation. The segmented character are then recognized using OCR (Optical Character Recognition). The extracted character in a number plates are entered in a excel sheet with the date and time of the vehicle entry. The accuracy obtained by the system is shown below:

MODEL	ACCURACY
Car Detection	98%
Number Plate Recognition	96%
Optical Character Recognition	95%

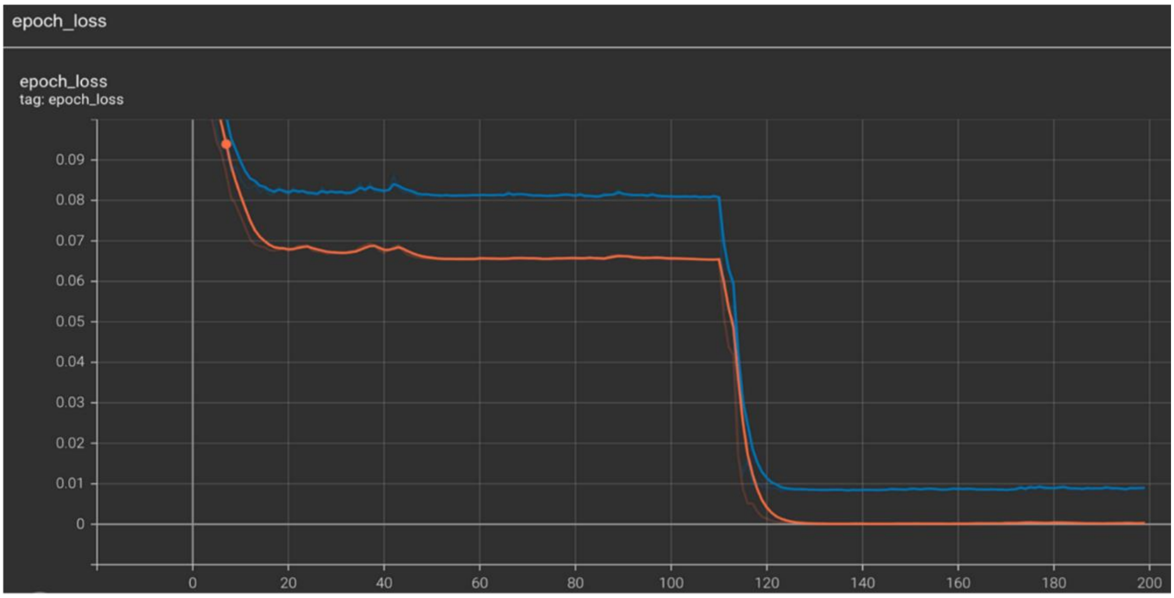


Fig 6.2 Graph Plots of the epochs

For the rating Criteria, the proposed system is evaluated by calculating the accuracy of the detection and the recognition which is defined as the number of correctly detected license plates divided by the number of correctly detected plus the number of incorrectly detected.

$$\text{Accuracy} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

The proposed method gives high accuracy for the three sequences and confirms that it can effectively detect the license plate and recognize its characters in different situations. Detection accuracy goes up to 98.4% and recognition accuracy goes up to 98.9% in some cases.

7. SCREENSHOTS

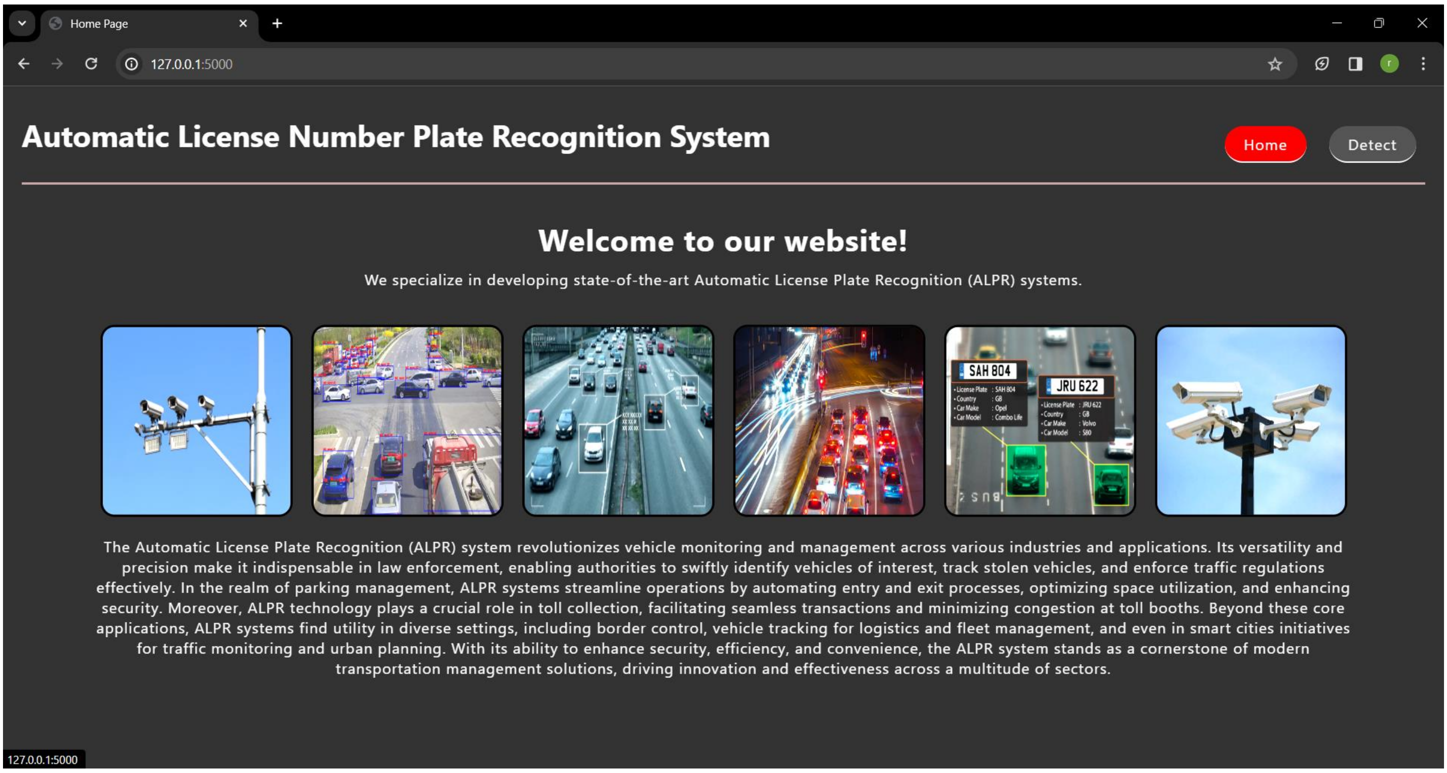


Fig 7.1 Home Screen

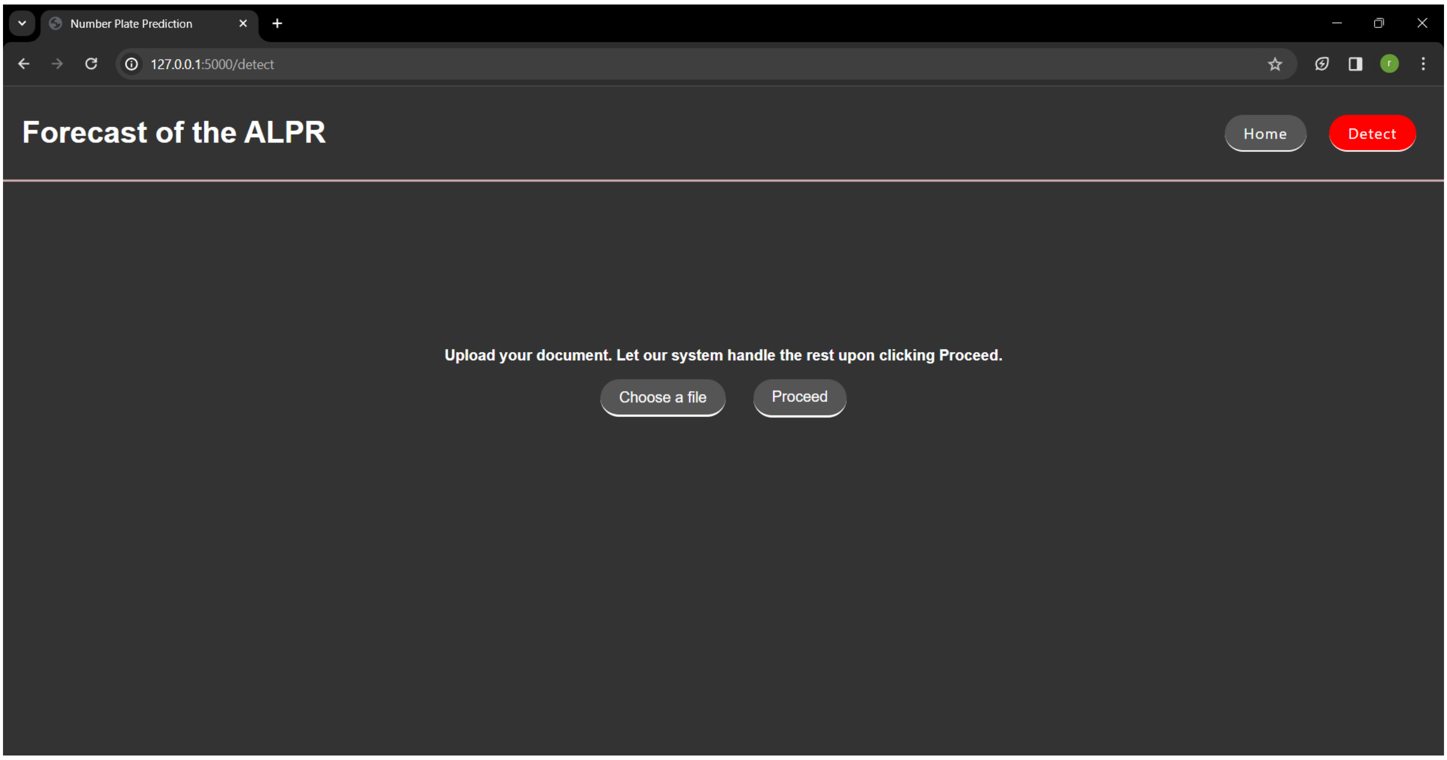


Fig 7.2 Detect Page

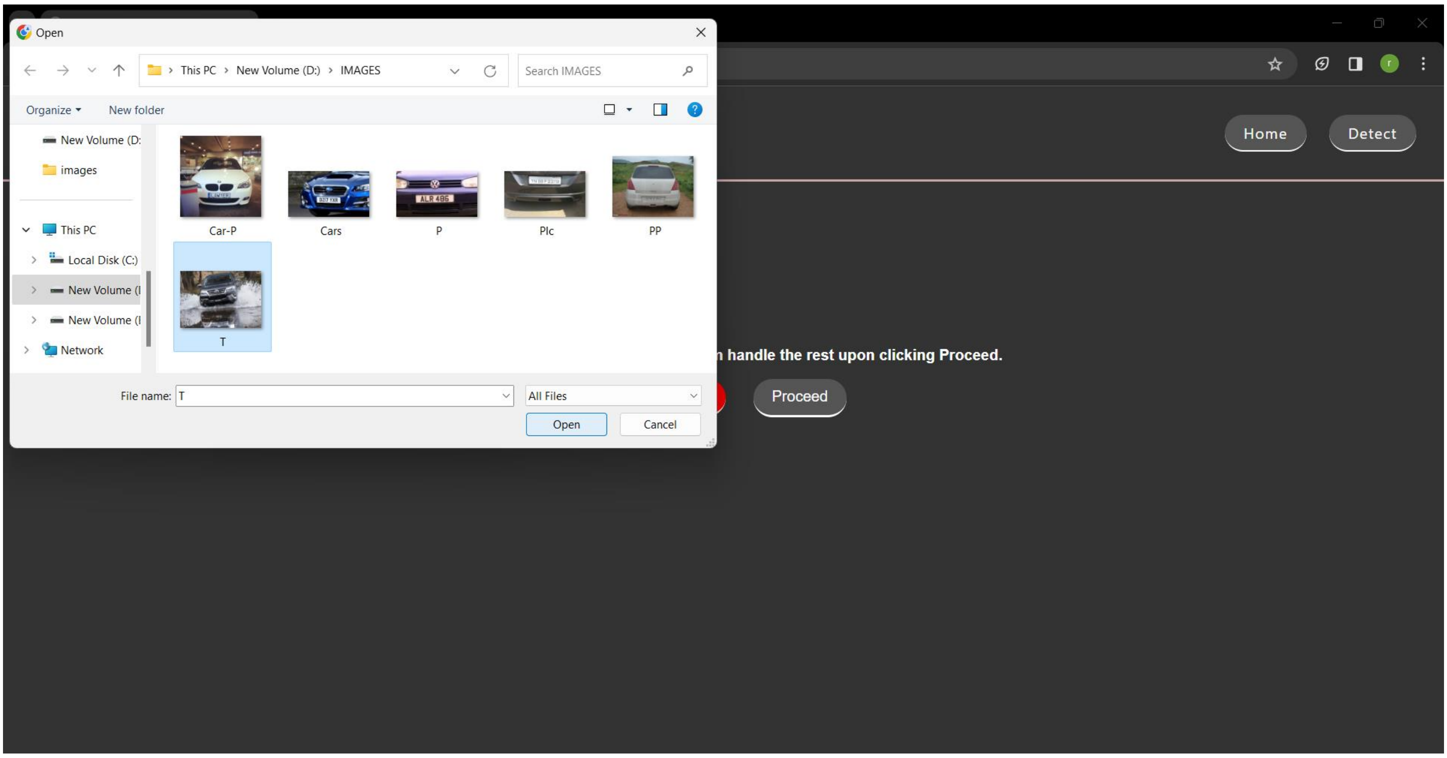


Fig 7.3 Selection of the Image to the System

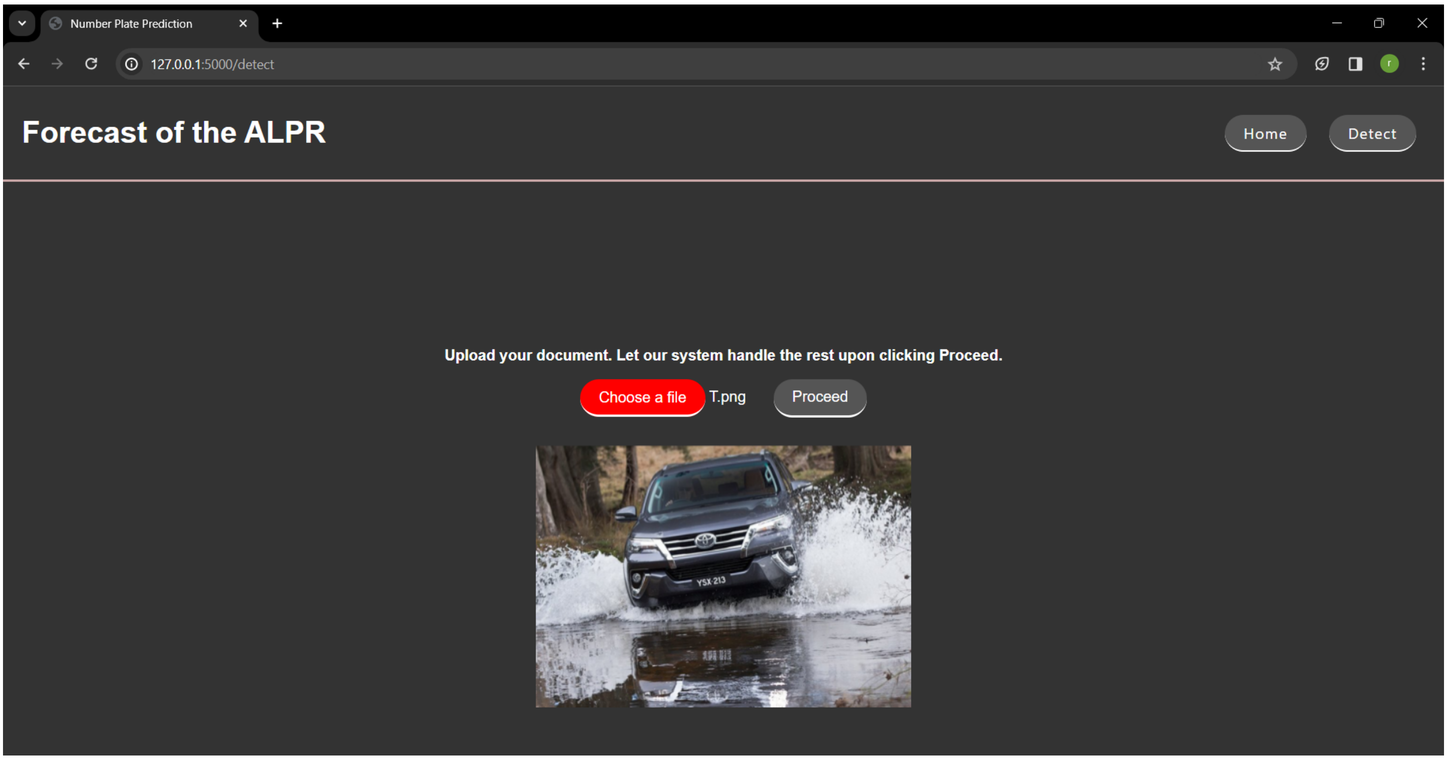


Fig 7.4 Display of the Selected Image

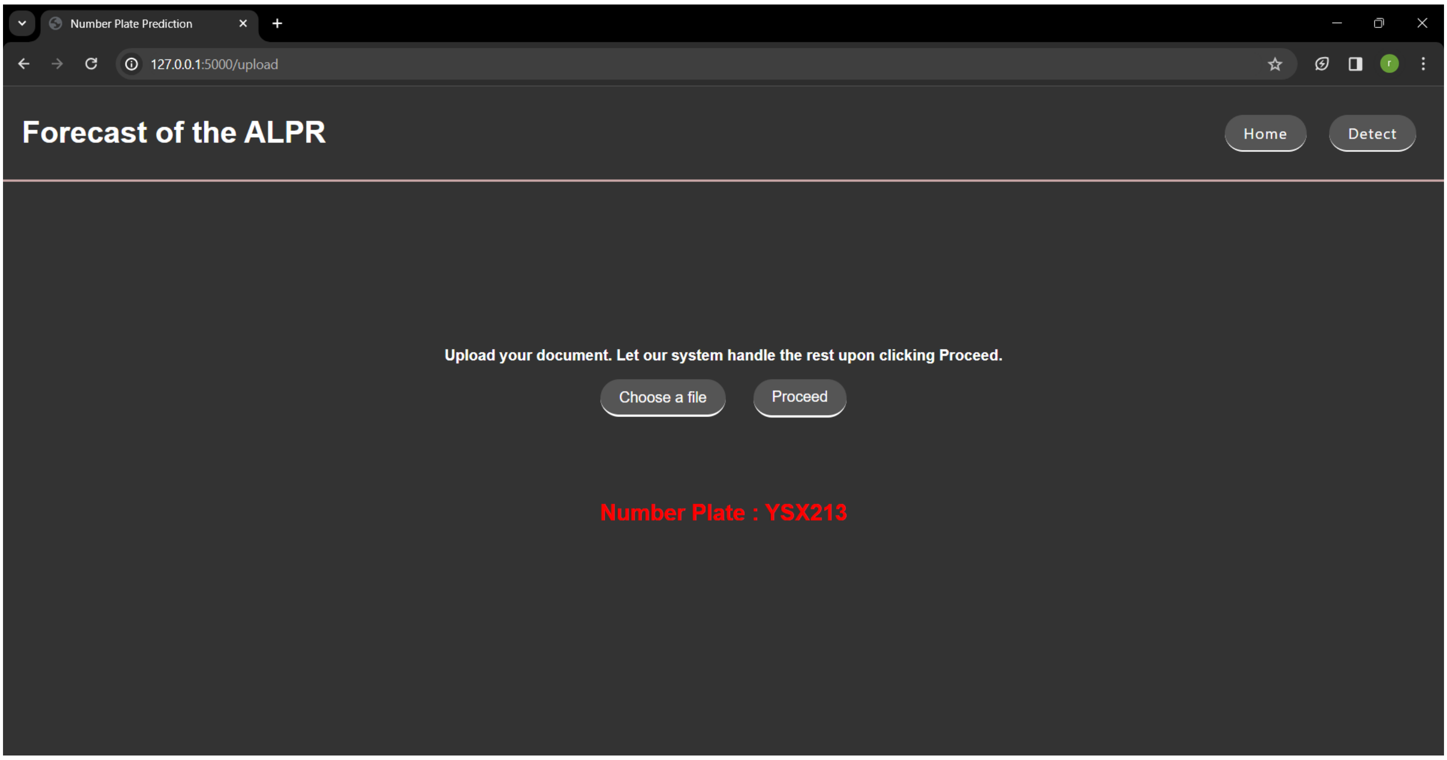


Fig 7.5 Output of the Image

8. CONCLUSION & FUTURE SCOPE

From review of various papers we conclude that there are different techniques available for recognition of car number plate such as Automatic license plate recognition, Novel method used for detects edge & fill holes less than 8 pixels only, categorize features in each stage, identifying & recognizing car license plate. The existing ALPR system has many deficiency such as inaccurate results. Furthermore, the character recognition has limitations such as number of characters which varies from region to region thus there is a great need of a universal algorithm for the same. The car number plate recognition system is capable of detecting an authorized car from a housing society, by recognizing the number plate of the car. It ensures an increase in security and management of the environment it is deployed in. It also decreases the overall cost and increases the efficiency. In further enhancement of the system a module can be added to automatically capture images of the number plates of the cars that have arrived at the gate. By including this module we can eliminate the need for loading the images manually in the application. A module can be added to determine the color of the vehicle. This color can be used to narrow down the search for a vehicle owner from the database, thus reducing the number of records to be scanned. The process of comparing two images can be further improved by using more advanced image comparing algorithms.

Therefore at this stage we use improved character segmentation method to reduce effort required for recognizing vehicle license number plate.

9. REFERENCES

[1] Zaarane A, Slimani I, Hamdoun A, Atouf I. Real-Time vehicle detection using crosscorrelation and 2D-DWT for feature extraction. J. Electr. Comput. Eng. 2019:1–9. https://doi.org/10.1155/2019/6375176. Jan. 2019.

[2] Slimani I, Zaarane A, Hamdoun A, Atouf I. Traffic surveillance system for vehicle detection using discrete wavelet transform. J Theor Appl Inf Technol 201896(17): 5905–17.

[3] AL Okaishi W, Zaarane A, Slimani I, Atouf I, Benrabh M. Vehicular queue length measurement based on edge detection and vehicle feature extraction. J Theor Appl Inf Technol 201997(5):1595603.

[4] Zaarane A, Slimani I, Hamdoun A, Atouf I. “Vehicle to vehicle distance measurement for selfdriving systems. In: 2019 6th international conference on control, decision and information technologies. CoDIT; 2019. https://doi.org/ 10.1109/CoDIT.2019.8820572. 2019 vehicles

[5] Zaarane A, Slimani I, Al Okaishi W, Atouf I, Hamdoun A. Distance measurement system for autonomous using stereo https://doi.org/10.1016/J.ARRAY.2020.100016.camera. Array Jan. 2020:100016.

[6] Shi X, Zhao W, Shen Y. Automatic license plate recognition system based on color image processing. Lect Notes Comput Sci 20053483(IV):1159–68. https://doi.org/ 10.1007/11424925_121.

[7] Dun J, Zhang S, Ye X, Zhang Y. Chinese license plate localization in multi-lane with complex background based on concomitant colors. IEEE Intell. Transp. Syst. Mag. Sep. 20157(3):51–61. https://doi.org/10.1109/MITS.2015.2412146. IEEE

[8] Ashtari AH, Nordin MJ, Fathy M. An Iranian license plate recognition system based on color features. Trans Intell doi.org/10.1109/TITS.2014.2304515. Transport Syst 201415(4):1690–705.

[9] Zheng D, Zhao Y, Wang J. An efficient method of license plate location. Pattern Recogn Lett Nov.
200526(15):2431–8. https://doi.org/10.1016/ j.patrec.2005.04.014.

[10] Deb K, Chae HU, Jo KH. Vehicle license plate detection method based on sliding concentric windows and histogram. J Comput 20094(8):771–7. https://doi.org/ 10.4304/jcp.4.8.771-777.

[11] Jia W, Zhang H, He X, Piccardi M. “Mean shift for accurate license plate localization,” in IEEE
Conference on Intelligent Transportation Systems. Proceedings, ITSC 20052005:566–71. https://doi.org/10.1109/ITSC.2005.1520110.

[12] Davis AM, Arunvinodh C, Arathy Menon NP. “Automatic license plate detection using vertical edge detection method. In: ICIIECS 2015 - 2015 IEEE international conference on

innovations in information, embedded and communication https://doi.org/10.1109/ICIIECS.2015.7193073. Systems; Aug. 2015.

[13] Rashedi E, Nezamabadi-pour H. A hierarchical algorithm for vehicle license plate localization. Multimed Tool Appl Jan. 201877(2):2771–90. https://doi.org/ 10.1007/s11042-017-4429-z.

[14] Khan MA, Sharif M, Javed MY, Akram T, Yasmin M, Saba T. License number plate recognition system using entropy-based features selection approach with SVM. IET Image Process Feb. 2018;12(2):200–9. https://doi.org/10.1049/iet-ipr.2017.0368.

[15] Hung KM, Hsieh CT. A real-time mobile vehicle license plate detection and recognition. Tamkang J Sci Eng 2010. https://doi.org/10.6180/jase.2010.13.4.09.

[16] Li H, Wang P, You M, Shen C. Reading car license plates using deep neural networks. Image Vis Comput 2018. https://doi.org/10.1016/j.imavis.2018.02.002.

[17] Daubechies I. The wavelet transform, time-frequency localization and signal analysis. IEEE Trans info Theor 1990. https://doi.org/10.1109/18.57199.

[18] Slimani I, Zaarane A, Hamdoun A. Convolution algorithm for implementing 2D discrete wavelet transform on the FPGA. In: Proceedings of IEEE/ACS international conference on computer systems and applications. AICCSA; 2016. https://doi.org/ 10.1109/AICCSA.2016.7945831.

[19] Silva LEV, Filho ACSS, Fazan VPS, Felipe JC, Murta LO. Two-dimensional sample entropy: assessing image texture through irregularity. Biomed. Phys. Eng. Express 2016. https://doi.org/10.1088/2057-1976/2/4/045002. for remote

[20] Nogueira K, Penatti OAB, dos Santos JA. Towards better exploiting convolutional neural networks sensing scene https://doi.org/10.1016/j.patcog.2016.07.001. Classification. Pattern Recognition; 2017.

[21] "The Caltech Database (Computational Vision at California Institute of Technology, Pasadena),."
http://www.vision.caltech.edu/archive.html.

[22] Hsu GS, Chen JC, Chung YZ. Application-oriented license plate recognition. IEEE Trans Veh Technol 2013. https://doi.org/10.1109/TVT.2012.2226218.

[23] Yuan Y, Zou W, Zhao Y, Wang X, Hu X, Komodakis N. A robust and efficient approach to license plate detection. IEEE Trans Image Process 2017. https:// doi.org/10.1109/TIP.2016.263190

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-184

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that **Kota Eswar Charan Rohith** , Narasaraopeta Engineering college has presented the paper title **Automatic License Number Plate Recognition System Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar

MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-184

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12^{th} & 13^{th} April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Inkollu Krishna Sai Ram, Narasaraopeta Engineering college has presented the paper title **Automatic License Number Plate Recognition System Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12^{th} and 13^{th} April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	**Chief-Convenor**	**Principal, Patron**
Dr.S.V.N.Srinivasu	**Dr.S.N.Tirumala Rao**	**Dr. M. Sreenivasa Kumar**

MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-184

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Jagadabi Yaswanth Abhishek Varma , Narasaraopeta Engineering college has presented the paper title **Automatic License Number Plate Recognition System Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Paper ID
NECICAIEA-2K24-184

Certificate of Presentation

This is to Certify that Thullibilli Prasanth Kumar, Narasaraopeta Engineering college has presented the paper title **Automatic License Number Plate Recognition System Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar

MHRD'S INNOVATION CELL (GOVERNMENT OF INDIA)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Paper ID
NECICAIEA-2K24-184

Certificate of Presentation

This is to Certify that Polavarapu Neelima, Narasaraopeta Engineering college has presented the paper title **Automatic License Number Plate Recognition System Using Deep Learning** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor	**Chief-Convenor**	**Principal, Patron**
Dr.S.V.N.Srinivasu	**Dr.S.N.Tirumala Rao**	**Dr. M. Sreenivasa Kumar**

Automatic License Number Plate Recognition Using Deep Learning

K.Eswar Charan Rohith
Student
CSE (Artificial Intelligence)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh
India
eswarkota9@gmail.com

T.Prasanth Kumar
Student
CSE (Artificial Intelligence)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh
India
prasanthkumar4888@gmail.com

I.Krishna Sai Ram
Student
CSE (Artificial Intelligence)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh
India
,inkollukrishnasai37@gmail.com

P.Neelima
Ass.Prof.NEC
CSE (Aritifical Intelligence)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh
India
pneelima888@gmail.com

J.Yeswanth Abhishek Varma
Student
CSE (Artificial Intelligence)
Narasaraopeta Engineering College
Narasaraopet
Andhra Pradesh
India
jagadabiabhivarma@gmail.com

Abstract— ***In this day and age, when there are more than 8 billion people on the planet and an equal amount of cars on the road, effective parking management has become essential. Vigilant gatekeeping is typically required in big residential areas because to the entry and egress of numerous vehicles, which can require significant labor and effort. In order to overcome these obstacles, the incorporation of Automatic License Plate Recognition (ANPR) is rather helpful. The approach that is being suggested entails creating an extensive database that holds car information for every member of a certain community. The Automatic Number Plate Recognition system, which has a camera placed strategically at the society's entrance gate, is smoothly linked to this database. The camera records a 10- to 12-second video as a vehicle approaches. This footage is then processed into images at a 24-frame-per-second (fps) rate utilizing sophisticated technologies like Matlab, Python, and several AI algorithms and search strategies. During this procedure, the image of the vehicle is carefully examined, which results in the number plate being extracted. Next, a cross-reference is made between this retrieved number and the previous database. When a match occurs, the gate opens automatically, allowing the car to enter the community.In addition, the system has the capability to record the times of entry and departure for every car, adding an extra degree of protection and streamlining monitoring. By limiting access to authorized vehicles exclusively, the implementation of an ANPR based system not only improves overall security but also streamlines the parking management process. Security staff now have a much lighter duty because they only haveto step in when a guest enters the society thanks to this innovation.***

KEY WORDS: Deep Learning, Convolutional Neural Network (CNN), Two-Dimensional Decomposition (2D-WD), Vertical Edges, Connected Components Analysis (CCA), Optical Character Recognition (OCR), Easy OCR.

I. INTRODUCTION

These days, the world's trend toward globalization demands constant technical development. Staying up with this trend means that technology needs to be updated frequently. The population of the world is growing at an accelerating rate, which causes a corresponding rise in the volume of vehicles on the roadways. A lot of work is required to park these cars in an efficient manner. The principal aim of the system's implementation is to provide an effective parking management and access control mechanism, thereby mitigating the obstacles related to parking. The first step of the system's operation is the input of a vehicle image, which is captured at four different levels .The input image is then preprocessed using a variety of tools, such as Matlab and a number of Python modules, libraries, and packages. The car's number is then recognized and extracted by the system. In order to determine access for the car, it lastly compares this extracted number with an already-existing database. If the number matches the database, admission to the specified society is made possible. On the other hand, access is refused if there is no match. Additionally useful in overseeing sizable events like parties and gatherings, this technique helps enforce social norms pertaining to discipline. It also offers thorough monitoring of vehicle entry and exit timings, providing information on how cars are moving within the designated region.

II. LITERATURE REVIEW

This section describes the first step of the procedure, in which a vehicle enters the society and the camera records a 12- to 15-second video. The video is then submitted to video scanning. Next, MATLAB is used to transform the video to 24 frames per second (fps). A crucial part of the idea is turning the license plate of the car into an image. To identify the vehicle's license plate, sophisticated image processing methods such as segmentation, identification, and localization are used. By using the Canny Edge Detection Algorithm, the number plate's edges may be distinguished, which eventually helps with identification. All vehicle number plates in India follow the same format, regardless of whether they are for commercial or noncommercial vehicles. In the first two letters of the General Form, the District Code comes after the State [1].

Code, followed by the addition of a unique 4digit numerical code, a random alphabet sequence, and a numerical series.

"BHARAT," a new television series from India, is designed to make car transfers across states easier. This series' format consists of the current year (YY), "BHARAT" (BH), a 4-digit code (XXXX), and alphabets that are chosen at random (XX) [2].

When the colors of the license plate and the background are similar, there could be a problem when taking the picture. Technologies for character segmentation and image processing are necessary for number plate recognition. It takes a high-resolution camera to get a good picture of the license plate. The video capture, conversion to frames, choosing a clear image, and plate region processing based on characteristics like aspect ratio and edge density are the four fundamental phases. Subsequent segmentation is subsequently executed to discern every digit and alphabet present on the number plate.

Despite the system's efficacy, certain drawbacks are identified. The fixed camera records a 12-15 second video, which is processed using Matlab to yield 240 frames per image. Specific techniques are applied to extract the license plate from these images. Several resolution methods, including Image Restoration and Contrast Enhancement, are employed to enhance the clarity of the retrieved images for effective number plate identification.

During the digitization process of images, various noise kinds, including poison noise, salt noise, and pepper noise, are encountered. To deal with these noises, several filtration methods are used, such as BM3D (Block Machine), Gaussian-guided, Minmax, Linear, Median, and Wiener. Experiments reveal that no single filter is superior to the others; however, BM3D is shown to be stable and all-encompassing. While linear filters are used to get rid of harmful noises, median filters are good at removing noises like salt and pepper. Adaptive fuzzy median filters work well in difficult situations to reduce noise from salt and pepper [3].

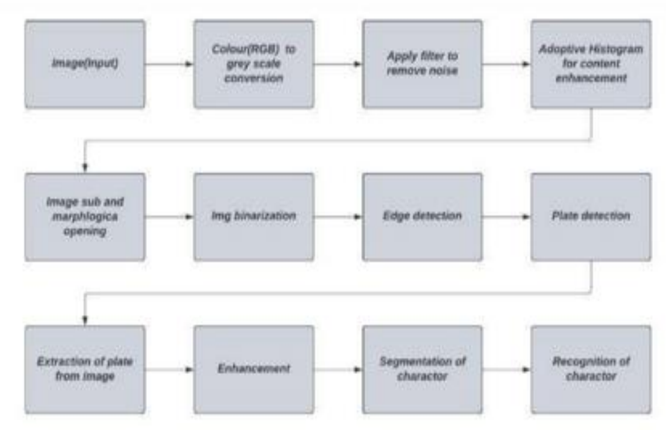
This study concentrates on identifying license plates in images that are well-lit and poorly-lit, although it advises against utilizing images that are noisy, fuzzy, or have low contrast. The procedure for recognizing license plates in photos with and without light is shown in the flowchart. Using Raspberry Pi and Matlab procedures, the paper

Plate edges	*Detect edge of number plate*
Character Analysis	*To find charactor*
Deskew	*To transform size*
OCR	*Number plate recognition*
Detection	*To detect number plate*
Binarization	*To convert grey scale to binary*
Post processing	*To process the number*
Character segmentation	*To segment number*

Fig 2.1

implements "ANPR" with an emphasis on parking premises security. Getting an image is the first step, which depends on having a steady and balanced image. When a car approaches, the sensor network's TSOP 1738 sensor detects it, prompting the Raspberry Pi camera to take pictures. Next are image processing chores like grayscale conversion and cropping the desired region. Dilation techniques are used to improve edges and lessen color disparities. The plate localization is which get over morphological processing's drawbacks. The execution of a neural network with feedforward backpropagation demonstrates a processing time of 1.3 seconds in a dynamic environment, and effectively accomplishes plate detection, localization, and character detection [4].

To capture license plate numbers at parking gates, Automatic Number Plate Recognition (ANPR) is used to



Block diagram of proposed system

Fig 2.2

monitor and regulate entry in both public and private organizations. The identification of stolen cars on the road is another application for this technique. At entrances and exits, cameras are positioned strategically, and the photos they take are strategically, and the photos they take are processed on computers and kept in a database for a long time. The system makes parking gates open and close automatically, and it needs highend, weatherproof, and dustproof cameras that can handle a range of weather conditions. It demonstrates its affordability by analyzing photos in any condition range and identifying cars on both the white- and black-lists, prohibiting unwanted access and guaranteeing security.

III.PROPOSED SYSTEM

Automatic License Plate Recognition (ALPR) is a system that uses image processing technology to identify and read license plate numbers on vehicles. To effectively recognize license plates, it is important to perform several preprocessing steps to ensure the images are in the correct format and are of high quality.

Hence, data preprocessing is an important step in ALPR to ensure accurate and efficient recognition of license plates. By applying the appropriate techniques for image acquisition, enhancement, plate localization, character segmentation, and character recognition, ALPR systems can achieve high accuracy rates and perform effectively in a variety of settings.

3.1.1 Dataset Description

The dataset was taken from the Kaggle repository which contain 273 images out of which 248 belongs to train category and 25 are test category images. All the images are of jpeg format.

3.1.2 Splitting the Set of data

The first step in splitting a dataset [5] is to prepare the data. This includes cleaning the data, removing any duplicates or outliers, and encoding categorical variables. The data should also be normalized or standardized if necessary.

Once the data has been prepared, the next step is to split the dataset into training, validation, and test sets.

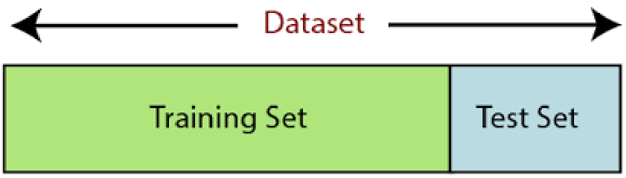


Fig 3.1

3.2 License Plate Detection and Recognition

The system technology uses a combination of image processing and computer vision algorithms to detect and recognize the characters on a license plate.

3.2.1 License Plate Detection

The first stage is to detect the license plate in an image or video stream. This is achieved using morphological operations to identify the region of interest (ROI). Once the ROI is identified, it can be cropped and passed on to the next stage of the LPDR process [6].

Fig 3.2

3.2.2 Two Dimensional Wavelet Decomposition

2D-WD is a technique used in signal processing and image analysis to decompose a two-dimensional signal or image into different frequency bands. The technique involves decomposing the signal into different scales and orientations using a series of filters, each of which extracts information at a particular frequency and direction.Scharr filters, to compute the gradient magnitude of the image in the vertical direction. The resulting gradient magnitude image can then be thresholded to identify regions with a high density of vertical edges [7].

To obtain a more robust extraction, additional processing steps can be performed. Connected component analysis can be used to group nearby edge segments into regions, which can then be filtered based on their size and shape.

d3.2.3 High Vertical Edge Density Extraction

The basic idea behind high vertical edge density extraction is to use a set of filters that are sensitive to vertical edges, such as the Sobel, Prewitt, or the image in the vertical direction. The resulting gradient magnitude image can then be threshold to identify regions with a high density of vertical edges.

To obtain a more robust extraction, additional processing steps can be performed. Connected component analysis can be used to group nearby edge segments into regions, which can then be filtered based on their size and shape.

3.2.4 Localization and Segmentation

License plate localization is the process of identifying and localizing license plates in an image or video frame. The goal is to accurately detect the location of the license plate within the image or video frame so that it can be processed for further tasks such as character recognition.

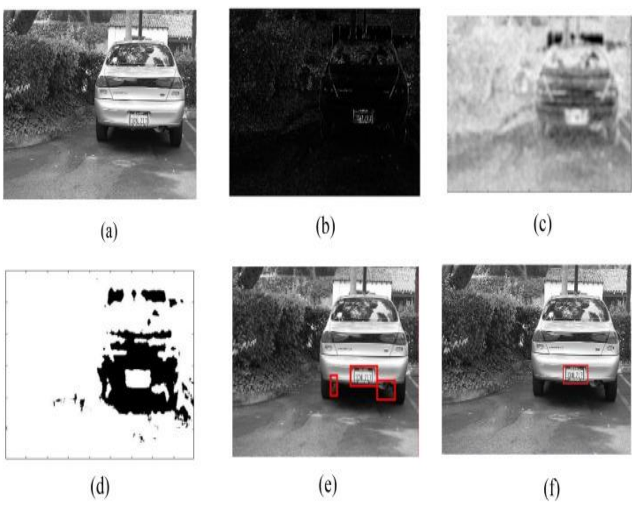


Fig 3.3

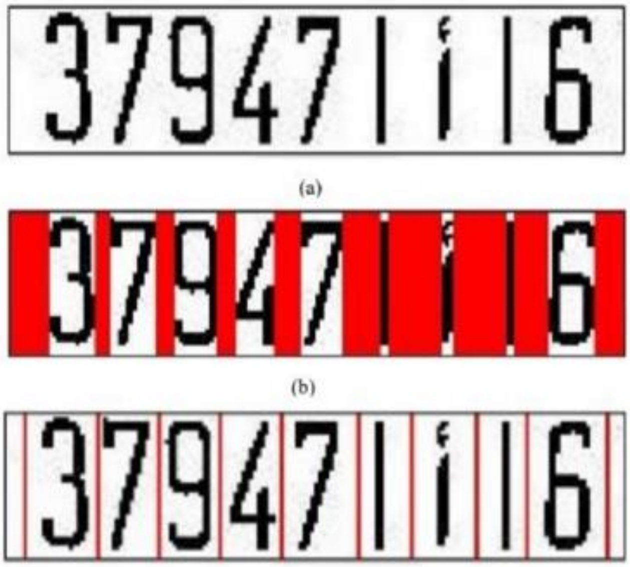
License plate segmentation is the process of separating the characters or digits of a license plate from the background in an image or video frame. The goal is to accurately segment the characters of the license plate so that they can be recognized and processed for further tasks such as character recognition.

Fig 3.4

IV.RESULT ANALYSIS

4.1 Result Analysis

The model analysis of this system involves assessing the underlying algorithms and techniques used to capture and recognize license plate numbers. The model is trained for 25 to 32 epochs with 10 to 15 batches and validated each epoch using loss and accuracy The system has shown to have high accuracy rates, ,fast processing speed, high throughput, and robust performance in various environmental conditions that can quickly and accurately read and process the license plate numbers in real-time.



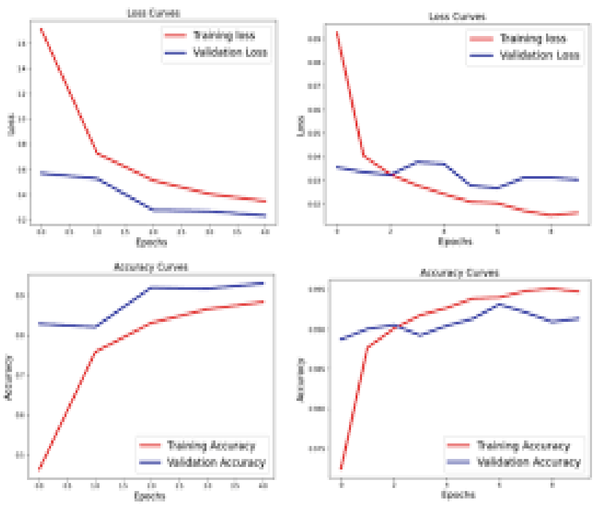


Fig 4.1

Furthermore, It has been improved in security and privacy features, such as data encryption, access control, and data retention policies, which help to protect the privacy of individuals whose license plates are being recorded and prevent unauthorized access to the data.

With the continued advancement of technology, it is likely that the system will become even more accurate and efficient, making them an even more valuable tool.

V. CONCLUSION

Every company in the modern world aspires to adopt the newest technologies, a need made even more pressing by the difficulties presented by the most recent worldwide epidemic. There is a constant need for innovation in the field of parking and traffic management systems. The increase in automobile traffic has made effective parking management essential. As a result, we have concentrated on investigating Automatic License Number Plate Detection using Deep Learning.

Gatekeepers must exert significant effort to manually handle the main gate in large housing societies, where thousands of vehicles constantly enter and exit. With the use of Automatic Number Plate Recognition (ANPR), which automates gate access by scanning and recognizing vehicle numbers, this procedure is made much simpler and less demanding on watchmen.

VI. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of LipschitzHankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand.Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science.

ALPR-Confrence-1.pdf

by A a

Automatic License Number Plate Recognition

Eswar Charan Rohith Kota[1], Krishna Sai Ram Inkollu[2], Yaswanth Abhishek Varma Jagadabi[3], Prasanth Kumar Thulluibilli[4], P Neelima[5]

[1,2,3,4] *Student, Department of CSE(AI), Narasaraopeta Engineering College, Narasaraopet, A.P, India.*

[5] *Assistant Professor, Deapartment of CSE(AI), Narasaraopeta Engineering College, Narasaraopet, A.P, India.*

eswarkota@gmail.com, inkollukrishnasai37@gmail.com , jagadabiabhivarma@gmail.com, prasanthkumar4888@gmail.com

pneelima888@gmail.com

Abstract— *In this day and age, when there are more than 8 billion people on the planet and an equal amount of cars on the road, effective parking management has become essential. Vigilant gatekeeping is typically required in big residential areas because to the entry and egress of numerous vehicles, which can require significant labor and effort. In order to overcome these obstacles, the incorporation of Automatic License Plate Recognition (ANPR) is rather helpful.*

The approach that is being suggested entails creating an extensive database that holds car information for every member of a certain community. The Automatic Number Plate Recognition system, which has a camera placed strategically at the society's entrance gate, is smoothly linked to this database. The camera records a 10- to 12-second video as a vehicle approaches. This footage is then processed into images at a 24-frame-per-second (fps) rate utilizing sophisticated technologies like Matlab, Python, and several AI algorithms and search strategies.

During this procedure, the image of the vehicle is carefully examined, which results in the number plate being extracted. Next, a cross-reference is made between this retrieved number and the previous database. When a match occurs, the gate opens automatically, allowing the car to enter the community. In addition, the system has the capability to record the times of entry and departure for every car, adding an extra degree of protection and streamlining monitoring.

By limiting access to authorized vehicles exclusively, the implementation of an ANPR based system not only improves overall security but also streamlines the parking management process. Security staff now have a much lighter duty because they only have to step in when a guest enters the society thanks to this innovation.

KEY WORDS: Deep Learning, Convolutional Neural Network (CNN), Two-Dimensional Decomposition (2D-WD), Vertical Edges, Connected Components Analysis (CCA), Optical Character Recognition (OCR), Easy OCR.

I. INTRODUCTION

These days, the world's trend toward globalization demands constant technical development. Staying up with this trend means that technology needs to be updated frequently. The population of the world is growing at an accelerating rate, which causes a corresponding rise in the volume of vehicles on the roadways. A lot of work is required to park these cars in an efficient manner. The principal aim of the system's implementation is to provide an effective parking management and access control mechanism, thereby mitigating the obstacles related to parking.

The first step of the system's operation is the input of a vehicle image, which is captured at four different levels .The input image is then preprocessed using a variety of tools, such as Matlab and a number of Python modules, libraries, and packages. The car's number is then recognized and extracted by the system. In order to determine access for the car, it lastly compares this extracted number with an already-existing database. If the number matches the database, admission to the specified society is made possible. On the other hand, access is refused if there is no match. Additionally useful in overseeing sizable events like parties and gatherings, this technique helps enforce social norms pertaining to discipline. It also offers thorough monitoring of vehicle entry and exit timings, providing information on how cars are moving within the designated region.

II. LITERATURE REVIEW

This section describes the first step of the procedure, in which a vehicle enters the society and the camera records a 12- to 15-second video. The video is then submitted to video scanning. Next, MATLAB is used to transform the video to 24 frames per second (fps). A crucial part of the idea is turning the license plate of the car into an image. To identify the vehicle's license plate, sophisticated image processing methods such as segmentation, identification, and localization are used. By using the Canny Edge Detection Algorithm, the number plate's edges may be distinguished, which eventually helps with identification. All vehicle number plates in India follow the same format, regardless of whether they are for commercial or noncommercial vehicles. In the first two letters of the General Form, the District Code comes after the State [1].

Code, followed by the addition of a unique 4digit numerical code, a random alphabet sequence, and a numerical series.

"BHARAT," a new television series from India, is designed to make car transfers across states easier. This series' format consists of the current year (YY), "BHARAT" (BH), a 4-digit code (XXXX), and alphabets that are chosen at random (XX) [2].

When the colors of the license plate and the background are similar, there could be a problem when taking the picture. Technologies for character segmentation and image processing are necessary for number plate recognition. It takes a high-resolution camera to get a good picture of the license plate. The video capture, conversion to frames, choosing a clear image, and plate region processing based on characteristics like aspect ratio and edge density are the four fundamental phases. Subsequent segmentation is subsequently executed to discern every digit and alphabet present on the number plate.

Despite the system's efficacy, certain drawbacks are identified. The fixed camera records a 12-15 second video, which is processed using Matlab to yield 240 frames per image. Specific techniques are applied to extract the license plate from these images. Several resolution methods, including Image Restoration and Contrast Enhancement, are employed to enhance the clarity of the retrieved images for effective number plate identification.

Plate edges	*Detect edge of number plate*
Character Analysis	*To find charactor*
Deskew	*To transform size*
OCR	*Number plate recognition*
Detection	*To detect number plate*
Binarization	*To convert grey scale to binary*
Post processing	*To process the number*
Character segmentation	*To segment number*

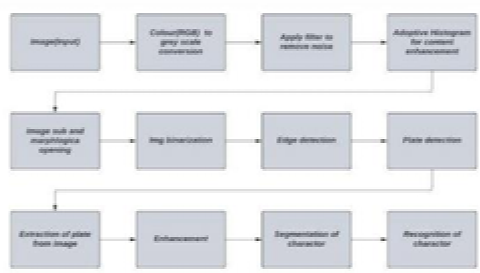
Fig 2.1

During the digitization process of images, various noise kinds, including poison noise, salt noise, and pepper noise, are encountered. To deal with these noises, several filtration methods are used, such as BM3D (Block Machine), Gaussian-guided, Minmax, Linear, Median, and Wiener. Experiments reveal that no single filter is superior to the others; however, BM3D is shown to be stable and all-encompassing. While linear filters are used to get rid of harmful noises, median filters are good at removing noises like salt and pepper. Adaptive fuzzy median filters work well in difficult situations to reduce noise from salt and pepper [3].

This study concentrates on identifying license plates in images that are well-lit and poorly-lit, although it advises against utilizing images that are noisy, fuzzy, or have low contrast. The procedure for recognizing license plates in photos with and without light is shown in the flowchart. Using Raspberry Pi and Matlab procedures, the paper implements "ANPR" with an emphasis on parking premises security. Getting an image is the first step, which depends on having a steady and balanced image. When a car approaches, the sensor network's TSOP 1738 sensor detects it, prompting the Raspberry Pi camera to take pictures. Next are image processing chores like grayscale conversion and cropping the desired region. Dilation techniques are used to improve edges and lessen color disparities. The plate localization is which get over morphological processing's drawbacks.

The execution of a neural network with feedforward backpropagation demonstrates a processing time of 1.3 seconds in a dynamic environment, and effectively accomplishes plate detection, localization, and character detection [4].

To capture license plate numbers at parking gates, Automatic Number Plate Recognition (ANPR) is used to monitor and regulate entry in both public and private organizations. The identification of stolen cars on the road is another application for this technique. At entrances and exits, cameras are positioned strategically, and the photos they take are strategically, and the photos they take are processed on computers and kept in a database for a long time. The system makes parking gates open and close automatically, and it needs highend, weatherproof, and dustproof cameras that can handle a range of weather conditions. It demonstrates its affordability by analyzing photos in any condition range and identifying cars on both the white- and black-lists, prohibiting unwanted access and guaranteeing security.



Block diagram of proposed system

Fig 2.2

III.PROPOSED SYSTEM

Automatic License Plate Recognition (ALPR) is a system that uses image processing technology to identify and read license plate numbers on vehicles. To effectively recognize license plates, it is important to perform several preprocessing steps to ensure the images are in the correct format and are of high quality.

Hence, data preprocessing is an important step in ALPR to ensure accurate and efficient recognition of license plates. By applying the appropriate techniques for image acquisition, enhancement, plate localization, character segmentation, and character recognition, ALPR systems can achieve high accuracy rates and perform effectively in a variety of settings.

3.1.1 Dataset Description

The dataset was taken from the Kaggle repository which contain 273 images out of which 248 belongs to train category and 25 are test category images. All the images are of jpeg format.

3.1.2 Splitting the Set of data

The first step in splitting a dataset [5] is to prepare the data. This includes cleaning the data, removing any duplicates or outliers, and encoding categorical variables. The data should also be normalized or standardized if necessary.

Once the data has been prepared, the next step is to split the dataset into training, validation, and test sets.

Fig 3.1

3.2 License Plate Detection and Recognition

The system technology uses a combination of image processing and computer vision algorithms to detect and recognize the characters on a license plate.

3.2.1 License Plate Detection

The first stage is to detect the license plate in an image or video stream. This is achieved using morphological operations to identify the region of interest (ROI). Once the ROI is identified, it can be cropped and passed on to the next stage of the LPDR process [6].

3.2.2 Two Dimensional Wavelet Decomposition

2D-WD is a technique used in signal processing and image analysis to decompose a two-dimensional signal or image into different frequency bands. The technique involves decomposing the signal into different scales and orientations using a series of filters, each of which extracts information at a particular frequency and direction.Scharr filters, to compute the gradient magnitude of the image in the vertical direction. The resulting gradient magnitude image can then be thresholded to identify regions with a high density of vertical edges [7].

To obtain a more robust extraction, additional processing steps can be performed. Connected component analysis can be used to group nearby edge segments into regions, which can then be filtered based on their size and shape.

3.2.2 High Vertical Edge Density Extraction

The basic idea behind high vertical edge density extraction is to use a set of filters that are sensitive to vertical edges, such as the Sobel, Prewitt, or the image in the vertical direction. The resulting gradient magnitude image can then be threshold to identify regions with a high density of vertical edges.

To obtain a more robust extraction, additional processing steps can be performed. Connected component analysis can be used to group nearby edge segments into regions, which can then be filtered based on their size and shape.

3.2.3 Localization and Segmentation

License plate localization is the process of identifying and localizing license plates in an image or video frame. The goal is to accurately detect the location of the license plate within the image or video frame so that it can be processed for further tasks such as character recognition.

License plate segmentation is the process of separating the characters or digits of a license plate from the background in an image or video frame. The goal is to accurately segment the characters of the license plate so that they can be recognized and processed for further tasks such as character recognition.

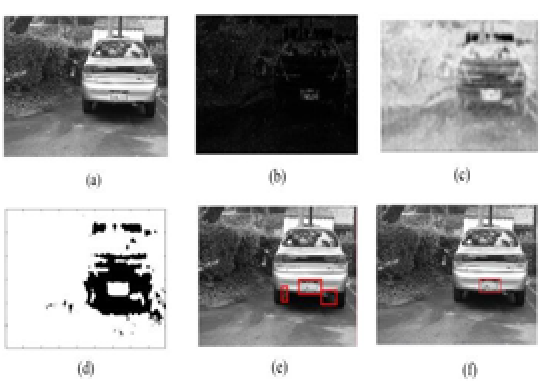


Fig 3.2

IV.RESULT ANALYSIS

4.1 Result Analysis

The model analysis of this system involves assessing the underlying algorithms and techniques used to capture and recognize license plate numbers. The model is trained for 25 to 32 epochs with 10 to 15 batches and validated each epoch using loss and accuracy

The system has shown to have high accuracy rates, ,fast processing speed, high throughput, and robust performance in various environmental conditions that can quickly and accurately read and process the license plate numbers in real-time.

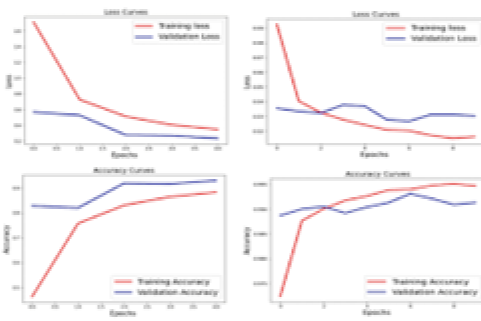


Fig 4.1

Furthermore, It has been improved in security and privacy features, such as data encryption, access control, and data retention policies, which help to protect the privacy of individuals whose license plates are being recorded and prevent unauthorized access to the data.

With the continued advancement of technology, it is likely that the system will become even more accurate and efficient, making them an even more valuable tool.

V. CONCLUSION

Every company in the modern world aspires to adopt the newest technologies, a need made even more pressing by the difficulties presented by the most recent worldwide epidemic. There is a constant need for innovation in the field of parking and traffic management systems. The increase in automobile traffic has made effective parking management essential. As a result, we have concentrated on investigating Automatic License Number Plate Detection using Deep Learning.

Gatekeepers must exert significant effort to manually handle the main gate in large housing societies, where thousands of vehicles constantly enter and exit. With the use of Automatic Number Plate Recognition (ANPR), which automates gate access by scanning and recognizing vehicle numbers, this procedure is made much simpler and less demanding on watchmen.

VII. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of LipschitzHankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand.Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 19

ALPR - Conference-1.pdfF

ORIGINALITY REPORT

9%	0%	7%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Omkar Shinde, Omkar Pagade, Yusuf Shaikh. "Automatic Number Plate Recognition", 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), 2023 Publication	3%
2	Manoj Kumar, Urmila Pilania, Vivaan Mittal. "Design and Development of an Intelligent Number Plate Identification System Utilizing Raspberry Pi Technology", 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), 2023 Publication	2%
3	Submitted to University of Bedfordshire Student Paper	1%
4	K. Krishna Kishore, B. Venkata Siva, Jinugu Babu Rao, N.R.M.R. Bhargava. "Tribological studies of Al-based metal matrix composite reinforced with Al-20Cu-10Mg ternary alloy using Taguchi technique", International	1%

Journal of Materials Engineering Innovation, 2018
Publication

5	Slimani Ibtissam, Zaarane Abdelmoghit, Al Okaishi Wahban, Atouf Issam, Hamdoun Abdellatif. "An Automated License Plate Detection and Recognition System Based On Wavelet Decomposition and CNN", Array, 2020 Publication	1%
6	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
7	Submitted to Gitam University Student Paper	<1%
8	Ninni Singh, Vinit Kumar Gunjan, Moustafa M. Nasralla. "A Parametrized Comparative Analysis of Performance between Proposed Adaptive and Personalized Tutoring System “Seis Tutor” with Existing Online Tutoring System", IEEE Access, 2022 Publication	<1%
9	github.com Internet Source	<1%

Exclude quotes On

Exclude matches Off

Exclude bibliography Off