ST. MARY'S UNIVERSITY

Ethiopian License Plate Detection and Recognition Using Deep Learning

A Thesis presented

by

Mikiyase Tafese Mesekelu

to

The Faculty of Informatics

Of

St.Marry's University

In Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Computer Science

February 29, 2024
Addis Ababa, Ethiopia

ACCEPTANCE

Ethiopian License Plate Detection and Recognition Using Deep LearningBy

Mikiyas Tafese Mesekelu

Accepted by the Faculty of Informatics, St. Marry's University, In Partial Fulfillment of the Requirements for the Degree of Master of Science

Thesis Examination Committee:

____

Internal Examiner

(Alembante Mulu Ph.d, Signature and Date)

____

External Examiner

(Minale Ashagrie Ph.d, Signature and Date)

____

Dean Faculty of Informatics

(Full Name, Signature and Date)

(Date of Defense)

February 2024

DECLARATION

Ethiopian License Plate Detection and Recognition Using Deep Learning

I. **The undersigned, declare that thesis work is my original work ,has not been presented for a degree in this or any other universities, and all sources of material used for the thesis work have been duly acknowledged.**

Mikiyas Tafese Mesekelu

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Million Meshesha (PHD)

million

Signature

Addis Ababa

Ethiopia

{Exact Date of Defense}

{February 2024}

Acknowledgment

First, I would like to thank the almighty God for his divine guidance and constant support throughout this study.

Next this research work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their invaluable assistance in time of need. First and foremost, I would like to express my gratitude to my advisor Dr. Million Meshesha for offering me the invaluable advice, guidance, monitoring and support during each step of my work. Beside, my thanks goes to Bedru Yimam for their advice and support. There are also individual that I am unable to list but have directly or indirectly helping me in completion of my thesis that I want to express my heart-felt thanks. My deepest gratitude does to my family for their unfailing support, patience and prayers

Table of Contents

List of Figure

List of Tables

Abstract

The study on License Plate Recognition (LPR) systems for Ethiopian to efficiently detecting and recognizing license plates from images. Our study aims to investigate and improve the performance of LPR systems by addressing key challenges, such as variations in license plate appearance, occlusions, and low image quality. The study uses a deep learning approach to design and develop an efficient LPR system.

Three steps make up a typical LPR system: segmenting characters, detecting license plates, and recognizing characters. In our method, we localized the license plates while reducing the false positives by licensing detection. This study looks into the automatic localization of license plates using the state-of-the-art Mask R CNN object detector. Then, we Apply preprocessing techniques such as resizing, normalization, and image enhancement (e.g., contrast adjustment, noise removal) to enhance the quality of the license plate region. Finally, we feed the segmented character plate number to the CNN based character recognition model.

In the study we collected 1190 plate images were fed into the system to assess the method's effectiveness, and the approach produced a 96. % accuracy rate for license plate recognition. An accuracy of 88 %was attained when all the recognized license plates were assessed to see how well the segmentation approach worked.

Keywords: Ethiopian License Plates; Digital Image Processing; License Plate Detection; License Plate Recognition; Deep Learning

Chapter one

Introduction

1.1 Background

A license plate is a small metal or plastic plate attached to the front and/or back of a vehicle that displays the registration number assigned to the vehicle by the state's Department of Motor Vehicles (DMV) [1]. This unique combination of letters and numbers acts as a type of identification for the vehicle and allows law enforcement officials to easily track down stolen vehicles or identify vehicles involved in crimes. The main usage of a license plate is to provide an easy way to identify a vehicle and trace its ownership [2][3]. License plates can be used for toll collection, parking enforcement, and automated traffic monitoring systems [4].From country to country worldwide, license plates vary in shape, size, and design based on vehicle type and region [5]. For instance, rectangular plates with alphanumeric characters are common in the US, while specialized plates exist for specialty vehicles, veterans, and causes. In Europe, plates are smaller and uniform, featuring alphabetic or numeric codes to indicate country of origin. Additional variations include plates for commercial vehicles, temporary plates, and government vehicle plates License plates in Ethiopia consist of a combination of letters (English and Amharic) and numbers, with a maximum of seven characters [6]. The first two characters represent the region where the vehicle is registered, followed by four digits and a final character. The final character indicates the type of vehicle, such as a private car or commercial vehicle. Some examples of Ethiopian license plates are shown in figure 1.1 below.

Figure 1. 1: samples of Ethiopian License Plate styles

In our country, license plates are issued by the Federal Transport Authority and are mandatory for all motor vehicles [7]. Proper display and registration of license plates are enforced by law enforcement agencies. The Ethiopian Road Transport Authority (ERTA) uses a variety of methods to detect license plates [6] [7][8]. These methods include visual inspection by law enforcement officers, and automated license plate recognition (ALPR) systems. However, it has its own limitations: the first method has been encountered in time-consuming, tedious, accuracy, and other human made errors. The second one which is ALPR has been faces in unable to detect and recognize unusual design and symbols of plate image, and only detect and recognize visible plate images [9].

To solve the problems of existing methods, researcher's shift to use a License Plate Recognition (LPR)[10][2][11]. License Plate Recognition is an advanced machine vision technology used to identify vehicles by their number plates without direct human intervention[12]. This development of Intelligent Transportation System provides numerous benefits to various industries, including reduced traffic incidents, improved parking management, enhanced toll collection, and increased surveillance [13]. Recent years have seen a huge surge in interest in the difficult scientific field of license plate recognition (LPR) [14]. This is because different locations have different conditions (such as light, color, dirt, shadows, a sense of character, language, and so on) and different kinds of license plates.

Since there are more and more automobiles on the road, it is becoming more and more difficult for humans to recognize license plates. The identification and detection of license plate sub-systems are typically combined into one vehicle license recognition system [1]. License plate recognition seeks to translate the characters on the plate into digital information, whereas license plate detection seeks to locate the car and its license plate.

Even though a lot of research has been done on the detection and recognition of license plates, many of the systems have limitations and only work well in certain settings [15]. Lighting, the existence of noise, blur, and distortion, tilting the camera when taking pictures, and other basic limitations are some examples. Days are just one of the many font form variations; each symbol has a varied size, color, and variation depending on the kind of vehicle, the state, and the nation.

Convolutional Neural Networks (CNN)-based deep learning techniques have demonstrated encouraging progress in license plate recognition in recent years [16]. CNN-based techniques recognize license plates using region-based or sliding window techniques. Some of the most often used methods for detecting and recognizing license plates include the Multi-scale Sliding Window CNN (MS-CNN) and Region-based CNNs like Faster R-CNN and SSD [17].[18][19]

As a result, research and development of a car license plate recognition system for Ethiopian license plates are required. In this work, we employ a deep learning methodology to improve the efficiency and precision of license plate recognition and detection.

1.2 Statement of the problem

With the substantial increase in the number of vehicles in our country, vehicle and parking problems have begun to negatively affect the comfort, health, and safety of residents [8][20]. Nowadays, traffic police officers rely on a manual process to identify vehicle license plates, which is slow, inefficient, and has a high likelihood of errors [7][8]. This can create safety risks on the roads and may not be adequate for identifying stolen or criminal vehicles, as officers cannot check every vehicle manually. In addition to this, recognizing vehicle license plate manually is time-consuming for law enforcement body and drivers. This leads to traffic congestion with the number of vehicles on the roads, coupled with an increase in theft and crime rates. Furthermore, detecting a stolen or criminal vehicle poses a significant challenge for police officers, and inaccurately recognizing[20].

Vehicle management technology have therefore grown in significance. The ability to identify and recognize license plates (LPs) is essential to vehicle management. Ethiopia has produced research works despite the difficulties in detecting and recognizing license plates [6][21][22][23]. A combination of Gabor Filter and Connected Combination Analysis (CCA) approaches was reported in a study in [21][22][23][6] to detect license plate images and OCR for recognitions. Nevertheless, these techniques are noise-sensitive and susceptible to variations in illumination, occlusion, and picture orientation. Furthermore, OCR is impacted by language (it might not be able to identify invisible characters), font styles and types, layout, and structure [24]. In order to tackle this issue, we employ a cutting-edge deep.

The main research questions that are investigated and answered in this study are the following:

1. What are the unique characteristics of Ethiopian license plates that must be considered in the development of a recognition system?

2. How to design a Mask R-CNN deep learning model that able to detect and recognize Ethiopian license plate images?

3. To what extent the proposed model detect and recognize Ethiopian license plate?

1.3 Objective of the study

1.3.1 General objective

The primary objective of this study is to detect and recognize Ethiopian license plates using Mask R-CNN deep learning algorithms.

1.3.2 Specific objectives

In order to achieve the general objective, the following specific objectives are identified.

- Collect a dataset of Ethiopian license plate images
- Apply image preprocessing techniques on collected license plate images
- Investigate detection techniques optimized for license plate
- Design a Mask R-CNN deep learning architecture to detect and recognize Ethiopian license plates.
- Train the proposed model using the preprocessed license plate images
- Evaluate the performance of the proposed system in detecting and recognizing license plates.

1.4 Scope of the Study

The main focus of this study is designing and developing a deep learning algorithm that able to detect and recognize Ethiopian License plate. Typically, a vehicle license identification system consists of two sub-systems: license plate recognition, which identifies the characters on the plate, and license plate detection, which locates the vehicle and its license plate. This study attempts to apply both license plate detection and recognition. However, this study is not included the recommendation of policy at the time of recognition.

1.5 Methodology of the study

Ethiopian license plate recognition is the process of automatically recognizing and reading license plate numbers on vehicles within Ethiopia. This technology utilizes image processing algorithms to extract the characters from the license plate image and convert them into a machine-readable format. License plate detection systems can be used for a variety of purposes, including traffic management, parking enforcement, and law enforcement. The use of this technology can significantly improve efficiency and accuracy in these areas, allowing for more effective management and enforcement of traffic laws. Towards achieving the aim of this study, the following step-by-step procedure followed.

1.5.1 Research Design

This study follows experimental research. To conduct extensive experiment, the study undertakes

Data collection and preparation, implementation and evaluation. Hereunder tasks performed, methods applied and tools used for each tasks are presented.

1.5.2 Data collection and preparation

The data of Ethiopian license plates is collected using web scraping techniques. We extract license plates images from online sources and check them for quality, correctness, and consistency. We also check for different backgrounds and lighting conditions. To enhance the quality and prepare the data for training the deep learning model, different data preprocessing techniques are applied, such as resizing, normalization, and noise reduction. Also data augmentation is used to increase the variability of the dataset by rotation, cropping and flipping methods.

1.5.3 Model Design

For constructing the model required for license plate detection and recognition, a deep learning architecture is used to recognize Ethiopian license plates. We evaluate different CNN models such as YOLOv3 (You-Only- Look-Once), Faster R-CNN, and SSD (Single-Shot Detector) to select the best performing license plate detection model.

1.5.4 Development Tools

The artifactual answer is developed in this part. In this activity, the architecture and intended functionality of the artifact are determined before the actual item or model is created. To create the CNN model, Keras is utilized, with TensorFlow as the backend. The CNN algorithm is coded in Python, and Visual Studio Code is used as an editor. A robust library that runs on top of TensorFlow is called Keras. For interacting with TensorFlow and managing computational graphs, it features a built-in graph data structure. Sequential and functional models are the two primary model types that Keras supports. The sequential approach is intended for straightforward systems in which layer stacking is to be done in a linear method. Multi-output models and other more general models with a varied layer structure are supported by functional models [33].

TensorFlow, in general terms, is a software framework for numerical competitions based on dataflow graphs [25]. Nodes in this graph stand for operations (such addition or division), and edges show the tensor data that is moving through the system. In deep learning, tensors are the standard data representation method. Tensors are really just multidimensional arrays, which are two-dimensional tables or matrices extended to high-dimensional data. Tensors, or three-dimensional arrays, are used to represent RGB images. Each pixel in an RGB image has three values that indicate its red, green, and blue components.

1.5.5 Evaluation

To find out how well the developed system supports a problem solution, it is appraised. Testing datasets were put into the created model in order to assess the system in a logical manner. The effectiveness of the suggested solution or model has been assessed using a variety of performance measures. Among these, f1-score, recall, accuracy, and precision are frequently used to gauge how well suggested solutions function.

Accuracy: is the proportion of true positives (include both true positives and true negatives) against the whole population. Accuracy may mislead the quality of the model if the class is not balanced.

$$\text{Accuracy} = (TP + TN) / (P + N)$$

Precision: is the proportion of true positives against the whole positives. Mathematically, it is expressed as:

$$\text{Precision} = TP / P$$

Recall or sensitivity: is the proportion of true positives against the whole true or correct data. It quantifies how well the model avoids false negatives [26]. It is also known as true positive rate or hit rate.

$$\text{Recall} = TP / (TP + FN)$$

F1-score: is the weighted average the precision and recall. The relative contribution of precision and recall to the F1-score are equal.

$$\text{F1-score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

1.6 Significance of the study

There are several significant implications of a study on Ethiopian license plate recognition using deep learning, including:

1. Enhanced law enforcement: By enabling automatic vehicle and owner identification, the study can enhance law enforcement. This can lead to an increase in general safety and security by enhancing the ability of law enforcement to follow and monitor criminal activity, stolen vehicles, and traffic infractions.
2. Improved traffic management: By providing real-time monitoring of traffic flow, congestion, and volume, accurate license plate identification helps enhance traffic management. This may make transportation planning and traffic control more effective.
3. Enhanced productivity and efficiency: Toll collecting, parking regulation, border control, and other activities associated with vehicle tracking can be automated with license plate recognition.
4. Better security and surveillance: Real-time monitoring of automobiles and their license plates allows deep learning-based license plate identification to improve security and surveillance.

1.7 Organization of the study

There are five chapters in this thesis. The issue statement, purpose, scope, significance, and methodology of the study are introduced in Chapter 1. A quick explanation of license plates is included in chapter two. We gave a summary of the techniques and strategies applied to license plate recognition. Thorough examination of the many works on license plate identification that have been presented. A thorough explanation of the suggested system is covered in Chapter 3. A detailed description of each component that makes up the suggested system is provided, along with its respective responsibilities. A detailed description of the experimental evaluation of the proposed model is provided in Chapter Four. There is a detailed description of both the suggested model's implementation and the dataset that was used. Chapter Five provides a summary of the main conclusions drawn from this study.

Chapter two

Literature review

2.1 Overview of License plate

License plates are a necessary component of vehicle registration and identification. They serve as a unique identifier for each vehicle on the road and facilitate the enforcement of traffic laws such as registration requirements, parking regulations, and toll collection [4]. License plates typically consist of a combination of letters and numbers, and can contain various symbols and color schemes depending on the country or jurisdiction. For example, in Ethiopia, license plates consist of three letters and four numbers arranged in a specific order.

The design, format, and color scheme of license plates can vary widely depending on the rules and laws of the issuing jurisdiction. Some countries may choose to include information such as the vehicle's registration expiry date or the name of the country on the plate, while others may use different colored plates for different classes of vehicles.

License plates play an important role in law enforcement and security, allowing authorities to track stolen vehicles, identify vehicles used in crimes, and monitor traffic flow [16]. They also help to enforce parking regulations, facilitate toll collection, and ensure that vehicles are properly registered and insured. Technological advancements have enabled the development of automatic license plate recognition systems, which use cameras and software to detect and recognize license plates in real-time. These systems have automated many tasks that were previously done manually and can help to improve efficiency and accuracy in law enforcement and traffic management.

License plates are important for several reasons [26]. Firstly, they help to identify and differentiate between vehicles on the road, making it easier for law enforcement officers to keep track of each vehicle. License plates also provide information about the vehicle owner, which is crucial in ensuring public safety. Secondly, license plates play a critical role in vehicle registration and ensuring that only registered vehicles are allowed on the road. This helps to ensure that only safe and roadworthy vehicles are allowed to operate on the public roads.

Thirdly, license plates are also crucial in enforcing traffic laws, including speed limits, parking regulations, and red light violations. This helps to reduce the number of accidents and promote safety on the roads. Additionally, license plates are effective in tracking stolen vehicles, identifying vehicles involved in accidents, and responding quickly in emergency situations, enabling first responders to arrive more quickly. Overall, license plates are an essential aspect of vehicle identification, registration, and public safety.

2.2 License plate recognition system

For recognizing the license plate, the process can be divided into two blocks. In the first blocks, the relevant position of license plate is located in the testing image. Then, extract the characters from the license plate and input them into the second block. There are more detailed steps involved in each block. In the following part, we present the application step by step (see figure 2.1).

The image taken from the scene may experience some complexities depending upon the type of camera used, its resolution, lightening/illumination aids, the mounting position, capability complex scenes, and other environmental constraints [1]. The variety of license plate styles, colors, fonts, sizes, and physical attributes could influence how accurately they are recognized. The procedure known as "number plate extraction" occurs when a vehicle is identified in the scene or image and the system applies plate localization functions to extract the license plate from the vehicle image. Before the recognition process, the retrieved number plate's characters are split.

An algorithm called character segmentation finds the alpha numeric characters on a license plate. The optical character recognition (OCR) procedures are then used to convert the segmented characters into an alpha numeric text entry. Algorithms like template matching and neural network classifiers are utilized for character recognition. The efficiency of each step determines how well an Automatic Number-Plate Recognition (ANPR) system performs. The performance-rate, also known as the success-rate, is a metric that is used to quantify the entire process. It is calculated by dividing the total number of input photos collected by the number of number plates that are successfully recognized. The three stages of the recognition process—number plate extraction, segmentation, and character recognition—are all included in the performance rate.

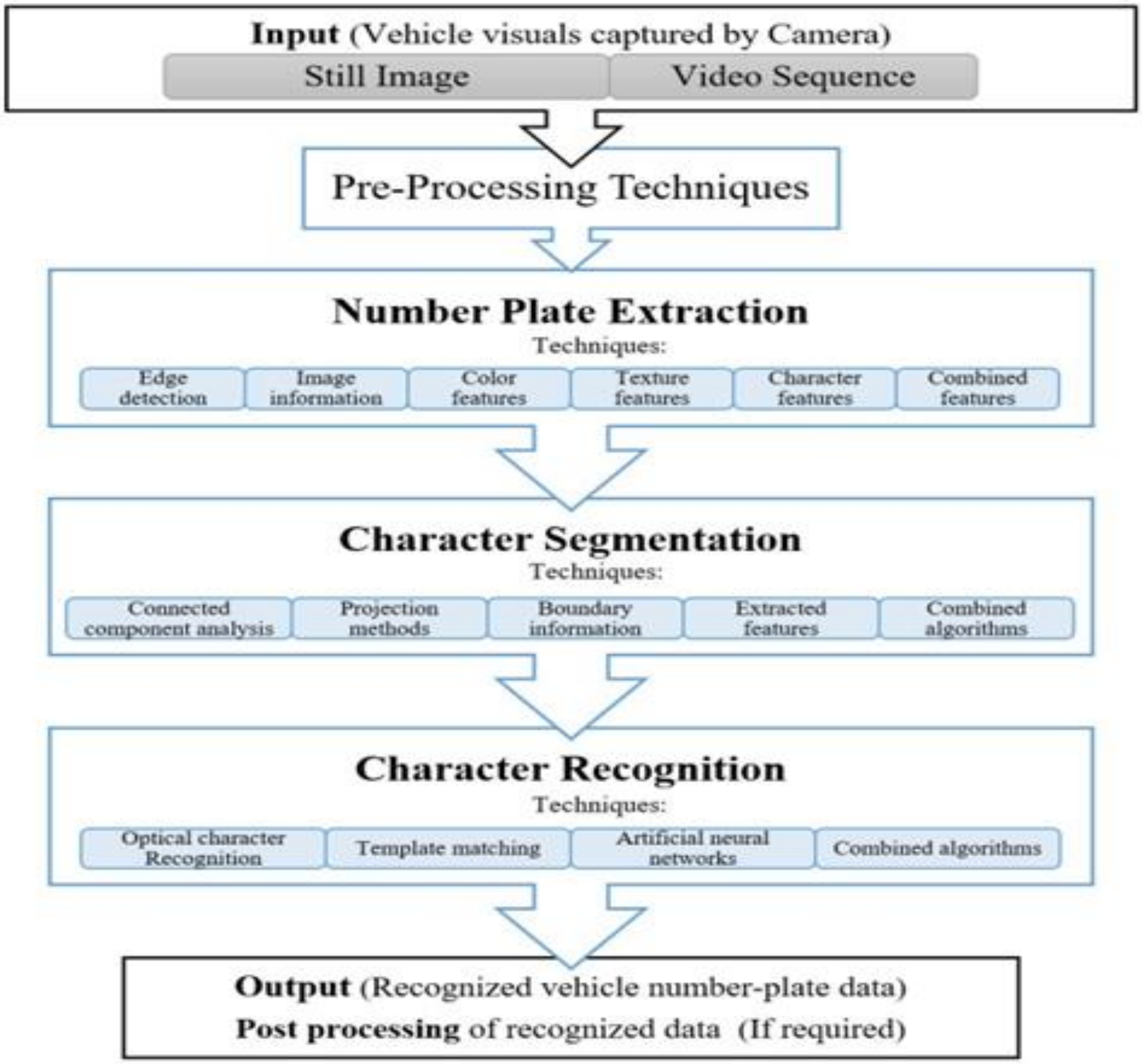


Figure 2. 1: General process of License plate recognition system

2.3 Steps in license plates recognition system

The identification of license plates occurs in two main stages: An essential piece of equipment found in parking lots, toll booths, security checkpoints, and other surveillance systems is license plate detection [11][25]. The technology can be used to automatically identify automobiles, follow their movement, and guarantee compliance with regulations. It reads license plate characters using cameras and software. It is crucial to remember that license plate detection systems have their limitations and might not be completely accurate because of things like dim lighting, obstructions from plates, and differences in font and color. Therefore, in order to guarantee the accuracy of the detections produced by the license plate recognition system, human verification and monitoring are still required. There are two common steps in the number plate recognition system.

1. Localizing the plate region from the entire image is the first step in detecting and extracting it. It incorporates a number of ideas from image processing, including segmentation, color conversion, noise filtering, and component analysis. A current area of research aims at achieving real-time performance in plate localization algorithms by using smaller computer resources to produce the fastest algorithm [16].

2. Character recognition in the image: Following the extraction of the plate region, each character on the plate has to be extracted and identified. This stage's primary processing involves locating and identifying characters on the plate, such as numbers and alphabets.

2.4 License plate detection

The process of identifying and reading license plates on automobiles using technology is known as license plate detection. An observer can accomplish this manually or by using automated license plate recognition (ALPR) technology. License plate numbers are captured, photos are analyzed, and the alphanumeric characters on the plates are identified using ALPR systems' cameras, sensors, and software. There are numerous significant applications for license plate detection in security, traffic control, and law enforcement. For instance, it can assist law enforcement with enforcing parking and traffic laws, tracking the whereabouts of suspects, and identifying stolen or wanted automobiles. Additionally, it can be used to track and monitor traffic flow in real-time, which can lessen traffic jams and increase the effectiveness of transportation as a whole.

More precise and effective license plate detecting systems have been created as a result of recent developments in artificial intelligence (AI) and machine learning. Real-time license plate recognition and analysis is possible with deep learning algorithms, which enables quick and precise vehicle identification while driving. The topic of autonomous vehicles is one of the most promising applications of license plate detection. For self-driving cars to operate safely on public roads, sophisticated sensor and perception systems are needed, and license plate recognition can be a useful tool in this process. Autonomous vehicles can have a better understanding of their environment and make more intelligent decisions about when to change lanes, accelerate, or slow down by recognizing and analyzing license plate numbers.

Ethiopian license plate detection is the process of identifying and reading Ethiopian license plates using technology for license plate recognition. The device takes pictures of license plates of cars going through parking lots, toll booths, surveillance sites, and other monitoring systems using cameras and sophisticated software. Identifying license plates in Ethiopia is particularly difficult because different parts of the country have varied font and color schemes. Furthermore, environmental elements like dim illumination and impediments might have an impact on how accurate license plate detection devices are.

License plate detection technologies have the potential to enhance security, traffic control, and law enforcement in Ethiopia despite these obstacles. Authorities can more quickly identify and track suspect or wanted vehicles, manage traffic flow, and enforce parking laws by automating vehicle identification.

As noted by scholars [27][11][28], there are three main methods of license plate detection :

1. Optical Character Recognition (OCR)-based detection: This technique analyzes and captures license plate photos in real time using cameras and OCR software. The characters on the license plate are recognized by the program, which then converts them into legible language. This technique works well in a range of lighting circumstances and is precise.
2. Template Matching-based detection: Using pre-defined license plate templates, this system compares the license plates recorded by cameras with the real ones. To find out if there is a match, the algorithm compares the retrieved features of the captured plate with the template. Although it might not be as precise, this method can be faster than OCR-based detection.
3. Deep Learning-based detection: Complicated artificial neural networks are used in this technology to identify and detect license plates on vehicles. Ethiopian license plates come in a variety of fonts, colors, and characters that can be recognized by the deep learning model. Although it may need more data and processing power than other approaches, this one may be more accurate.

2.5 Digital Image Processing

Giving computers the ability to analyze and comprehend visual input from the environment around them is the subject of the study area known as computer vision[29][27]. In order to extract information from digital images or video streams, digital image processing techniques are frequently used. More precisely, image processing is the editing and improvement of digital images. It is a subset of computer vision. Developing models and algorithms that allow machines to detect and interpret visual data in the same way that humans do is the main objective of computer vision. This includes tracking, image segmentation, object recognition, scene reconstruction, and other tasks.

In computer vision, digital image processing is frequently one of the most important and initial phases. This is so that machines can have a hard time deriving meaningful information from raw visual data, which frequently contains noise, distortions, and other aberrations. Image processing allows us to extract features for recognition, increase image quality by removing noise, and segment images into areas of interest. The term "digital image processing" describes the alteration or manipulation of digital images through the use of different computer tools, algorithms, and techniques. A variety of tools, including digital cameras, scanners, medical imaging equipment, and satellites, can be used to take these pictures. Digital image processing involves various operations such as image enhancement, image restoration, image compression, image segmentation, feature extraction, pattern recognition, and object detection.

It is essential to several areas, including robotics, machine vision, remote sensing, and medical imaging. Our everyday perception and utilization of images has been completely changed by digital image processing. The practice of manipulating digital images obtained from different sources using computer algorithms and techniques is known as digital image processing. It is a subfield of computer science that deals with creating algorithms for digital image processing, analysis, and transformation. Digital cameras, scanners, medical imaging devices, and satellites can all be used to take these pictures. Since its conception, digital image processing has undergone tremendous development and is now a crucial component of contemporary technology. It has uses in many different industries, including robotics, engineering, and medical. It can be applied to enhance visual. There are five major steps of digital image processing. Details of each of them are discussed as follows:

2.5.1. Image Acquisition

The process of processing digital images begins with image acquisition. Digital images are obtained throughout the image acquisition process from a variety of sources, including digital cameras, scanners, and satellite photographs [2]. The accuracy of the ensuing processing processes may be impacted by the image's quality at this point. As a result, it's critical that photos be taken in high definition using the right parameters for the intended application. For instance, in medical imaging, diagnosis and treatment planning may depend significantly on the quality of the obtained image. Consequently, in order to guarantee the highest possible image quality, medical images are frequently obtained through the use of specialist tools and processes. Satellite photos are used in remote sensing to collect information about the surface of the world. The photos are taken with certain sensors.

Image acquisition for license plate detection entails taking pictures of cars from a variety of sources, including security cameras, traffic cameras, and mobile devices with cameras. The license plate must be readable and clearly apparent in the photos, which must be taken in this manner. The accuracy of license plate detection is highly dependent on the quality of the obtained photos. It is crucial to use cameras that have the right quality and settings as a result. For instance, more detailed images can be produced by cameras with higher resolutions, which facilitates the identification of license plate numbers.

2.5.2. Image Preprocessing

Cleaning and preparing a picture for additional analysis or alteration is known as image preprocessing, and it is a crucial stage in the digital image processing process. Enhancing the image's quality by eliminating undesired noise, artifacts, or distortions is the main objective of image preprocessing, which also aims to prepare the image for further tasks like segmentation, feature extraction, or object recognition [3].

Depending on the unique needs of the image and the intended result, a variety of approaches are employed in image preprocessing. One popular method is noise reduction, which is taking out undesirable signals or random fluctuations that might deteriorate the quality of the image. Techniques like wavelet denoising, Gaussian smoothing, and median filtering can be used to accomplish this. Contrast enhancement is one of the key components of image preprocessing. This entails modifying the image's brightness and contrast to enhance its visual appeal and draw attention to key details. To increase contrast and enhance overall image quality, apply techniques like gamma correction, adaptive contrast stretching, or histogram equalization [4].

Image preprocessing is essential for enhancing the precision and dependability of the license plate detection process. Taking a picture of the scenario with the cars is the first stage in the license plate detection process. The presence of many kinds of visual noise and distortions in this photograph may make it challenging to precisely identify license plates. Thus, to clean and ready the image for detection, image preprocessing techniques including thresholding, filtering, and morphological procedures are applied.

Filters: Image edges are improved and noise is eliminated by the use of filters. This can be accomplished by using a variety of filters, including the Sobel, Gaussian, and median filters. Using a threshold value, the image can be turned into a binary image in which the pixels are either black or white. This serves to accentuate the license plates and simplify the picture.

Morphological operations are used to fill in gaps and eliminate small items from the binary image, including erosion, dilation, opening, and closing. This makes it more likely that the license plates will be recognized as a single, complete object.

Following the completion of preprocessing, license plate detection techniques can be used to find the license plates inside the picture. Several applications, such as parking lot management, law enforcement, and traffic management, depend on the accurate and dependable detection of license plates. In order to increase the process's overall accuracy and efficiency, picture preprocessing is therefore essential to license plate detection.

2.5.3. Segmentation

Image segmentation is also a crucial step in license plate detection, which involves dividing the preprocessed image into multiple segments, or regions, based on their unique properties [5]. This technique helps to identify and isolate the license plate region in the image, which can then be extracted and processed further. In license plate segmentation, various algorithms such as region growing, edge detection and template matching can be used to segment the image. Edge detection algorithms such as Sobel, Canny, and Laplacian filters can be used to detect edges and edges of the license plate, which can be used as a cue to segment the plate .

Region growing algorithms can be used to group together the pixels belonging to the plate, by looking for regions with high gradients or homogeneity. The region growing algorithm can be guided by the edges detected earlier. Template matching is another approach, which involves comparing the preprocessed image with a predefined template of the license plate to find the best match. This method requires careful selection of the template, as it may contain variations due to different plate sizes, fonts, and backgrounds.

Other image segmentation methods include the following:

1. Watershed algorithm: This method is based on controlling the flooding of watersheds in the image to segment it into regions. It is effective in segmenting images with complex or overlapping objects.
2. K-means clustering: This method groups the pixels in the image into K clusters based on their color, texture, or other features. This method is useful for segmenting images with a clear separation of different objects or regions.
3. Graph-based segmentation: This method uses graph theory to segment the image into regions based on a weighted graph of the pixels. The nodes of the graph represent the pixels and the edges represent the relationship between the pixels.
4. Mean-shift algorithm: This method is based on clustering similar pixels together in the image, by iteratively shifting the color value of each pixel towards the mean color of the surrounding pixels. It is useful for segmenting images with a large number of objects or regions.
5. Gaussian mixture model: This method models the image as a mixture of Gaussian distributions and segments it by clustering the pixels based on their likelihood of belonging to each Gaussian. It is useful for segmenting images with complex textures or color distributions.
6. Active contours (or snake) algorithm: This method uses an energy function that defines the shape and location of the contour that separates the object from the background. It is useful for segmenting images with a clear boundary between the object and the background.

Once the license plate region is segmented, it can be extracted and processed further for recognition tasks such as character segmentation and recognition. Accurate segmentation of the license plate is critical for successful recognition, which has important implications for various applications such as traffic analysis, surveillance, and security.

2.5.4. Feature Extraction

In license plate recognition, feature extraction is the process of identifying and extracting relevant features from the segmented license plate image. The extracted features are then used to classify the license plate and extract the textual content. The features extracted from the license plate image usually include shape, color, texture, and size information. Some of the common features used in license plate recognition are the following:

Histogram of oriented gradients (HOG): This feature describes the distribution of gradients in the license plate image. It is useful in detecting the shapes and edges of the characters. Histogram of oriented gradients (HOG) is a popular feature extraction method that is used to detect and describe local features in an image. The method works by first dividing the image into small cells and calculating a histogram of gradient orientations within each cell. The histograms of neighboring cells are then concatenated to form a feature vector that describes the local structure of the image. This feature vector can be used to train a classifier for object recognition or detection. The HOG method is robust to variations in illumination and contrast, and can detect objects of different sizes and orientations. The method has been successfully used in various computer vision applications, such as pedestrian detection, face recognition, and vehicle tracking. However, the method may not perform well in images with complex backgrounds or occlusions, where the local structure may be affected by the background or other objects. Overall, HOG is a powerful feature extraction method that can be used to detect and describe local features in images. The method has been widely used in computer vision and machine learning, and has led to significant advances in object recognition and detection.

Color features: Color features are another important type of feature used in image processing. Color is a key visual cue that humans use to distinguish between objects and to understand the scene. Color features can be used to segment or classify objects based on their color, and to extract meaningful information from images. Some common color features used in image processing include color histograms, color moments, and color coherence vectors. A color histogram is a representation of the color distribution in an image, where the image is divided into bins based on the color channel values. Color moments are statistical measures of the color distribution in an image, such as the mean, standard deviation, and skewness of the color channels. Color coherence vectors measure the spatial coherence of color regions in an image, and can be used to segment objects based on their color.

However, color features can be affected by changes in illumination, shadows, and reflections, which can lead to false detections or incorrect classifications. To address this issue, various color normalization and correction methods have been developed to improve the robustness of the color features, such as gray-world assumption, histogram equalization, and color constancy techniques.

Morphological features: Such features include the size, shape and orientation of the license plate. Morphological features are useful in distinguishing between license plates of different sizes and orientations. Morphological features are another type of feature used in image processing, particularly in license plate detection. Morphology is the branch of mathematics that deals with the study of the structure and shape of objects. Morphological operators are used to extract features based on the shape and structure of objects in an image.

Morphological characteristics like edge detection, dilatation, erosion, opening, and closing are frequently employed in license plate recognition. The process of determining the borders between several regions in an image is called edge detection. Objects in an image can be made larger or smaller using dilations and erosions, respectively. An image can be made noise-free or have gaps filled in with opening and shutting. The license plate area can be separated from the rest of the picture by segmenting it using morphological features. For instance, the margins of the license plate area can be improved and minor objects removed using dilations and erosions. While closure can be used to close gaps or join, opening can be used to eliminate noise or tiny spaces inside the license plate area.

Texture features: Texture features are another important type of feature used in image processing. Texture refers to the pattern or arrangement of the visual elements in an image, such as the roughness or smoothness of a surface, or the regularity or randomness of a pattern. Texture features can be used to segment or classify objects based on their texture, and to extract meaningful information from images. Some common texture features used in image processing include gray-level co-occurrence matrix (GLCM), gray-level run length matrix (GLRLM), and local binary patterns (LBP).However, texture features can be affected by changes in scale and orientation, which can lead to false detections or incorrect classifications. To address this issue, various scale and rotation-invariant texture features have been developed to improve the robustness of the texture features, such as multi-scale LBP and rotation invariant GLCM.

2.5.5. Image understanding

Image understanding is the steps where a computer system able to identify and recognize objects, patterns, and shapes within an image [5]. This is achieved through the use of advanced algorithms that analyze and extract features from the image. These features include color, texture, shape, and size, among others.

Once the features have been extracted, the system can use machine learning to categorize and classify them into different groups and identify the patterns that exist within the image. This is particularly useful in applications such as facial recognition, object detection, and medical imaging, where the accurate identification and recognition of patterns can be critical in making informed decisions and accurate diagnoses.

Overall, Image understanding plays a vital role in helping computers to interpret and understand images, enabling them to perform complex tasks that would otherwise be impossible for human beings to do manually. These days deep learning is widely used for image understanding and recognition.

2.6. Deep learning convolutional neural network

In deep learning, a convolutional neural network is a type of deep neural network commonly used in image detection and recognition tasks [26]. The algorithm consists of multiple layers of neurons that process and extract features from the input image and then classify it into a specific class. The first layer of a CNN is usually a convolutional layer, which applies a set of filters to the input image to extract features that may be important for classification. The filter weights are learned during training using backpropagation. After the features are extracted, the result is passed to a pooling layer, which down samples the features by taking the maximum or average value over a certain region (e.g. 2x2) of the features. Multiple convolutional and pooling layers can be stacked to further extract and refine features from the input image. The final layer of a CNN is typically a fully connected layer, which takes the output of the previous layers and maps it to the predicted class probabilities.

During training, the algorithm adjusts the weights in each layer to minimize the error between the predicted output and the actual output (i.e. the labels). This is done using back propagation and an optimization algorithm such as stochastic gradient descent (SGD). CNNs have shown great success in image-related tasks such as object detection, facial recognition, and even digital art style transfer. They can also be used for other types of data such as audio and text. CNNs are also computationally efficient due to their ability to share weights across spatial regions of the input, allowing them to handle large images in a reasonable amount of time.

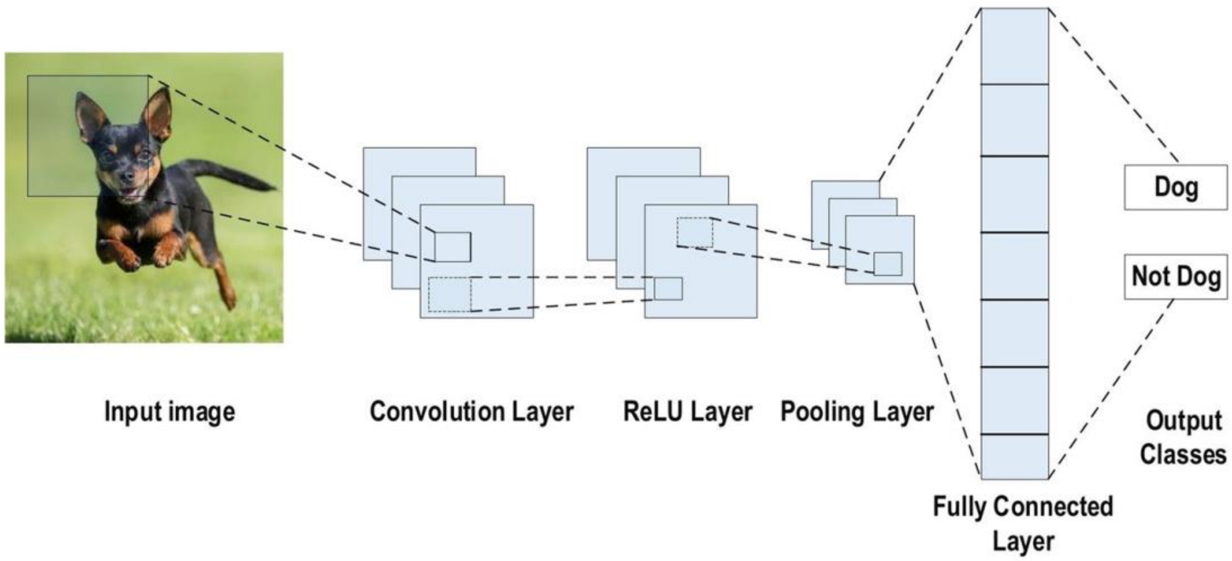


Figure 2. 2: An example of CNN architecture for image classification.

The CNN architecture consists of a number of layers (or so-called multi-building blocks). Each layer in the CNN architecture, including its function, is described in detail below .

Convolutional Layer: The convolutional layer is the most important part of the CNN design. It is comprised of an assortment of convolutional filters, also referred to as kernels. These filters are used to convolve the input image, which is expressed as N-dimensional metrics, to produce the feature map that is output.

• Definition of a kernel: The kernel is represented by a grid of discrete numbers or values. We refer to each value as the kernel weight. At the start of CNN training, random numbers are assigned to serve as the kernel's weights. Furthermore, there exist multiple techniques for initializing the weights. The kernel then learns to extract important features by adjusting these weights at each training epoch.

Convolutional Operation: In order to extract characteristics that are pertinent for classification, this layer applies a set of filters on the input image. The basic unit of a convolutional neural network (CNN) is called a convolutional layer. It is the layer that creates a set of output features, or feature maps, by applying the convolution process to the input data or image. A collection of filters, or kernels, in the convolutional layer convolve over an input image to extract pertinent characteristics including edges, corners, forms, patterns, and textures. Every filter is slid over the input image, and the dot product between the filter and the input's local region is calculated at every point. This procedure is carried out once more for every filter position, producing a feature. The goal of training stochastic gradient descent (SGD) and back propagation is to minimize the error between the expected and actual outputs, which is how the filters in the convolutional layer are learned. The model is then more resilient and efficient in classifying new input data because to the learnt filters, which turn into feature detectors for particular patterns or shapes in the input data.

In conclusion, the convolutional layer is essential to the CNN because it creates a collection of output feature maps by removing pertinent features from the input data or image. The network becomes increasingly efficient at tasks like segmentation, object detection, and image classification thanks to the learned filters.

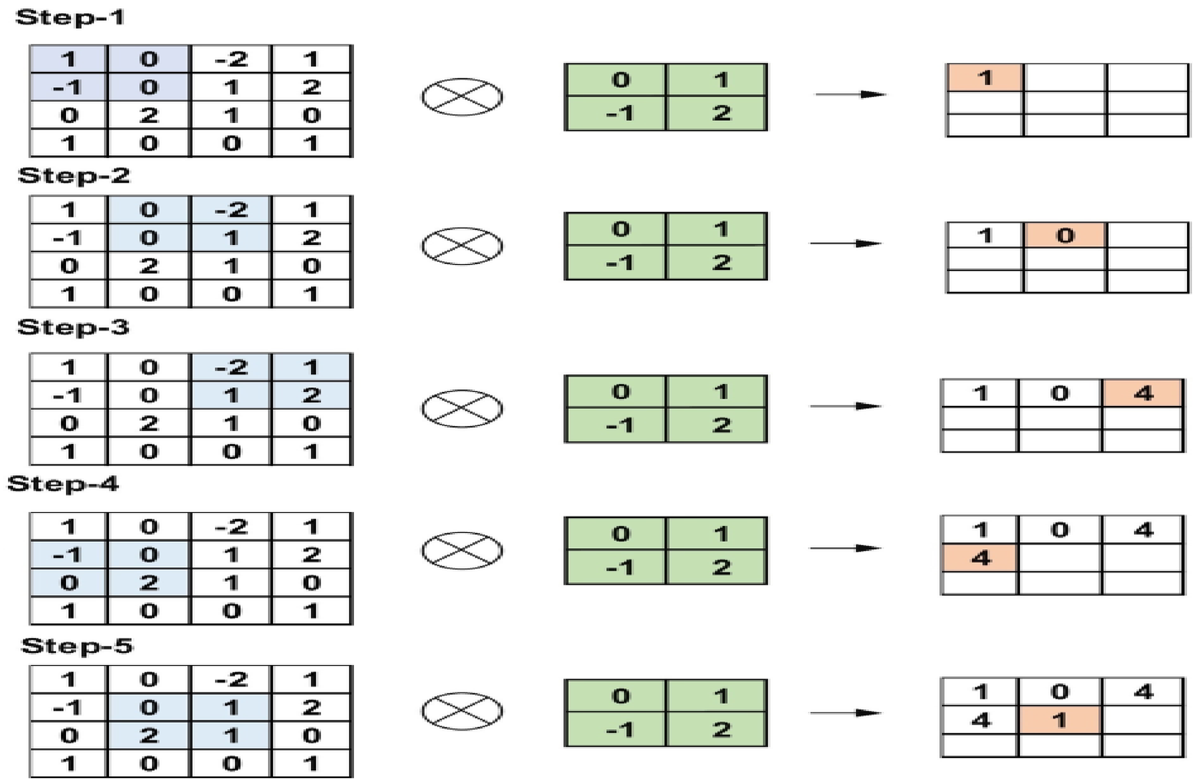
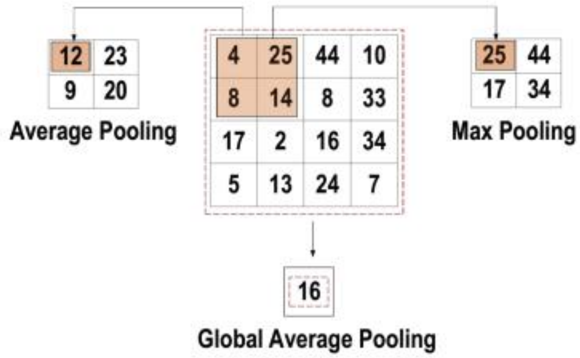


Figure 2. 3: primary calculations executed at each step of convolutional layer

- **Pooling layer:** The features that the convolutional layers extracted are down-sampled by this layer. It aids in lowering the network's computational complexity. A convolutional neural network (CNN)'s pooling layer is a layer that keeps crucial information while shrinking the input data's spatial size (height and width). By lowering the number of parameters, the pooling layer helps the network compute more quickly while simultaneously strengthening the output layer's resistance to slight changes in the input. The max pooling layer is the most popular kind of pooling layer. The way it operates is by taking the maximum value across each region after breaking the input data up into smaller sections. A 2x2 max pooling layer, for instance, would divide.

Another type of pooling layer is the average pooling layer, which takes the average value over each region instead of the maximum value. The pooling layer is usually inserted after the convolutional layer(s) in the CNN architecture. By doing this, the pooling layer helps to extract the most important features from the input data, while at the same time reducing the spatial dimensionality of the input, allowing for faster computation in the subsequent layers of the CNN.

The size of the pooling layer and the stride of the sliding window used to compute the pooling operation are also important hyper parameters that affect the performance of the network.



Activation Function (non-linearity The fundamental job of all kinds of activation functions in all kinds of neural networks is to map the input to the output. Convolutional neural networks (CNNs) and other deep neural networks depend heavily on the activation function. It is used to add non-linearity to a layer's output, assisting the network in learning intricate and nonlinear correlations between inputs and outputs.

A set of activation values, or feature maps, are obtained by applying the activation function element-by-element to the convolution operation's output in a convolutional layer. CNNs have a number of frequently utilized activation functions, such as:

- ReLU (Rectified Linear Unit): This is the most widely used activation function. ReLU sets any negative activation values to zero, and any positive values are unchanged. This function is simple and computationally efficient, making it ideal for deep networks.

- Sigmoid: The sigmoid function maps any input value to a value between 0 and 1, providing a smooth activation gradient. This function was widely used in earlier neural networks but has largely been replaced by ReLU.
- Tanh: The hyperbolic tangent function is similar to sigmoid but outputs a value between -1 and 1. It is also used less frequently than ReLU.

- Softmax: This activation function is used for multi-class classification problems and produces a probability distribution over the output classes.

The choice of activation function depends on the specific task and data, and experimentation with different functions is often necessary to determine the best one. The activation function plays a crucial role in the learning capacity and performance of a CNN, as it is responsible for the network's ability to model complex nonlinear relationships and prevent overftting.

Fully Connected Layer: This layer is typically found at the very end of the CNN architecture. This layer uses the so-called Fully coupled (FC) method, in which every neuron is coupled to every other neuron in the layer above. The output from the earlier levels is mapped to the anticipated class probabilities in this layer. In a neural network, a fully connected layer is one in which every neuron is connected to every other neuron in the layer before it. The fully connected layer of a convolutional neural network (CNN) translates the output from the preceding convolutional and pooling layers to the anticipated class probabilities. The number of neurons in the fully connected layer is typically defined based on the number of classes in the dataset being used. For example, if the dataset has 10 classes, the fully connected layer might have 10 neurons, with each neuron representing the probability of the input belonging to a certain class.

Regularization to CNN

Overftting is the main problem for CNN models in terms of achieving well-behaved generalization. When a model performs exceptionally well on training data but fails miserably on test or unseen data, it is said to be over fitted. The opposite of an overfitted model is an underfitted model, which happens when the model does not learn enough from the training data. If the model performs well on both training and testing data, it is considered just-fitted. These three types are illustrated in Figure 2.5. Various intuitive concepts are used to help the regularization to avoid over-fitting such as dropout and augmentation.

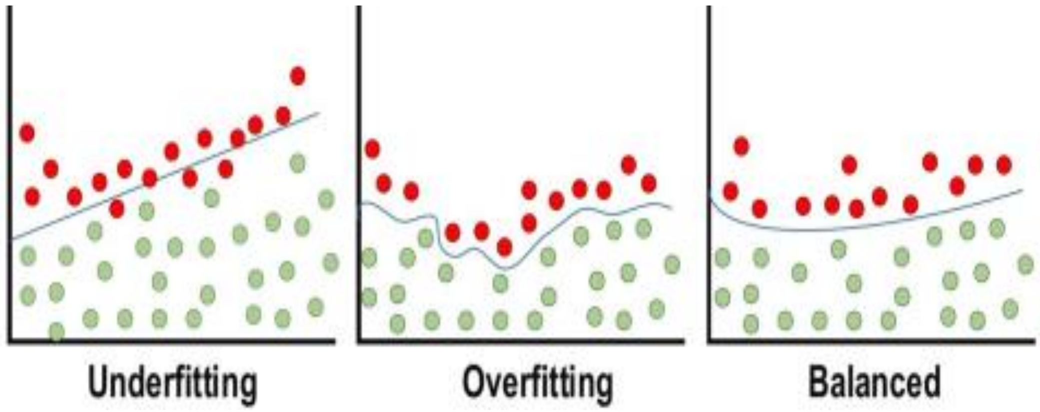


Figure2. 4:over-fitting and under-fitting case

Dropout: This is a generalization method that is frequently used. Neurons are dropped at random during every training epoch. By doing this, the model is forced to learn several independent features and the feature selection power is dispersed evenly over the entire group of neurons. The dropped neuron will not participate in either forward or backward propagation throughout the training phase. On the other hand, during testing, the full-scale network is used to make predictions.

Data Augmentation: One of the simplest ways to prevent over-fitting is to train the model on a sizable amount of data. Data augmentation is utilized to do this. The training dataset is artificially expanded using a number of methods. The techniques for data augmentation are covered in more detail in the following section.

Batch Normalization: The output activations' performance is guaranteed by this technique. This performance has a Gaussian distribution with units. To normalize the output at each layer, subtract the mean and divide the result by the standard deviation. This can be differentiated from and integrated with other networks, even if it is possible to think of it as a pre-processing operation at each network layer.

CNN architecture

A number of CNN architectures have been introduced in the past ten years [21, 26]. A key component in raising various apps' performance is their model architecture. From 1989 till the present, numerous changes have been made to CNN's architecture. These adjustments include of regularization, parameter optimization, structural reformulation, etc. On the other hand, it should be highlighted that the creation of new blocks and the rearranging of the processing units were primarily responsible for the significant improvement in CNN performance. The most innovative advancements in CNN topologies, specifically, concerned the utilization of network depth. This section reviews the most widely used CNN architectures, starting in 2012 with the AlexNet model and concluding in 2020 with the High Resolution (HR) model. Examining the characteristics of these structures (such input.

AlexNet

LeNet's debut marked the beginning of the history of deep CNNs [89]. The handwritten digit recognition tasks that the CNNs were limited to at the time were unscalable to all image classes. AlexNet is a well-respected deep CNN architecture [30], having produced ground-breaking achievements in image identification and classification. After introducing AlexNet for the first time, Girshick [30] enhanced CNN's capacity for learning by deepening it and applying a number of parameter optimization techniques. The fundamental layout of the AlexNet architecture is shown in Figure 15. Hardware constraints at the time hampered the deep CNN's capacity to learn. In order to get over these hardware constraints, AlexNet was trained in parallel on two GPUs (NVIDIA GTX 580).

Faster R-CNN is a well-liked object detection model based on deep learning that was first presented in a research study in 2015[30]. It is an enhanced variant of the original R-CNN (Region-based Convolutional Neural Network) method that recognizes objects in images by utilizing the ideas of feature extraction and region proposal. A deep convolutional neural network (CNN) is the foundation of Faster R-CNN. After extracting features from an input image, the CNN utilizes a region proposal network (RPN) to identify possible object areas. The RPN finds the most likely areas in an image where an object is probably present. Accurate object positions and sizes are subsequently obtained by refining these regions. Finally, the items inside the improved zones are categorized using a classification network.

2.6 Related Works

There have been initiatives aimed at recognizing Ethiopian license plates. An automatic license plate recognition system for Ethiopian license plates was proposed by Puarungroj and Boonsirisumpun [1]. The authors employed a correlation-based template matching technique for character recognition, Gabor filters for plate detection, and connected component analysis for character segmentation. Plate detection rate was 88.9%, character segmentation accuracy was

83.9% among identified plates, and character recognition rate was 84.7% among correctly segmented characters, according to the study's findings. Additionally, they have a 63.1% total plate recognition rate.

For Ethiopian automobiles, Shraddha and Ghadage [2] created a car plate recognition system. The authors used a template matching technique for character identification with connected component analysis (CCA) for plate region extraction. The image is first sent into the OCR once the plate region has been removed. Matlab is used to model the system, while real photos are used to test performance. Twenty-three test photos were used to assess the proposed system. The accuracy of the author's experiment was 52.63%. Additionally, a different test was conducted on the OCR system and plate region localization. The results of the individual tests indicated that the OCR had an accuracy of 82.11% and the plate region localization an accuracy of 82.6%.

Sanap and Narote [3], proposed a deep learning based Ethiopian Car’s License plate detection and recognition. The authors were used OpenCV- python to build their model. The author’s segmentation model achieved 86.66%. The segmented characters are going to be given to the classification or recognition model which was developed using Convolutional Neural Network. The CNN model classifies each character image to its corresponding class.

In the study [38] presents a framework for a vehicle license plate detection system that combines morphological operations and deep learning. The proposed method achieves high precision, recall, and overall accuracy, with a detection rate of 97.90% using morphological operations and 100% recall and 98.65% overall accuracy with the addition of deep learning. The evaluation of the proposed method is based on a single dataset called "YellowLP" consisting of 1050 images. The generalizability of the framework to other datasets or real-world scenarios is not extensively explored.

The authors of the paper [39] provided an end-to-end technique for effectively identifying and detecting license plates in real-time applications while taking into account various environmental factors like light and weather, artificial and natural noise, and variations in illumination. The suggested approach increases speed and accuracy in locating the plate by using a single-shot detector-based deep learning model to detect automobiles in input photos and video frames. A convolutional network-based architecture is suggested for locating the plate, and long short-term memory (LSTM) and a deep convolutional network are used to identify characteristics associated with the plate.

The authors of the study [40] stated that learning-based algorithms for automatic license plate identification are predicated on training and test data that are aligned. Nevertheless, in harsh environmental circumstances or in performance and excellent detection ability. YOLOV3 is used for license plate detection, utilizing its multi-scale detection capabilities and fine-grained features in high and low layers .CRNN is employed for license plate recognition, leveraging its excellent detection ability to accurately recognize license plate characters.

In the paper[44] the authors proposed a Character-Region Awareness For Text Detection (CRAFT) is used to detect the poly region of the license plate's word line(s) and effectively detect the text line of both one-line and multi-line license plates . The extracted features are then fed to the Bi-LSTM architecture with Connectionist Temporal Classification to predict the output text in each input region.

The authors of the study [45] designed and integrated a system for detecting and recognizing license plates using the YOLOv7, STN, and LPRNet models. The suggested approach can instantly and precisely identify Chinese license plates. The study's outcome, which has an average accuracy of 96.1% in complicated contexts, demonstrates good robustness in such settings.

The paper in the study [46] presented a unique method for detecting license plates that increases the precision and effectiveness of license plate detection by extending the Sobel mask. The efficacy of the suggested technique in identifying automobiles is demonstrated by its high accuracy of 98% in detecting license plates.

The pervious attempted work on detection and recognition mainly focus on constrained environments and used traditional based approaches. This results to improper detection and recognition of LP. Therefore, we need to develop a deep CNN model for detection and recognition of Ethiopian LP. Our research considered unconstrained environment and adopt state–of-the-art model called Mask R-CNN. This model has a framework to do state-of-art instance segmentation and generates high-quality segmentation mask. This results to the proposed work have good speed, accuracy, and easy to implement.

Chapter Three

Methods

3.1. Overview

In this chapter, detailed description of the proposed system for detection and recognition of license plate is discussed. It requires passing via a series of steps starting from preprocessing of images, segmentation of the ROI, feature extraction and learning to license plate detection and recognition. Recognition mainly encompasses two major phases; these are training phase, and testing phase. In section 3.2 general description about the proposed system architecture is presented. In the next sections, each task, such as preprocessing, detection, segmentation and recognition are described thoroughly.

3.2. Proposed System Architecture

The identification of license plate regions-of-interest (ROIs) through segmentation of the extracted regions is the first of four basic steps in the number plate recognition process. The identification of the license plate's characters comes last. The Deep Fully Convolutional Network (DFCN), Region Proposal Network, ROI pooling, Fully Connected (FC) networks, Bounding Box Regressor, and Classifier make up the LP detection and segmentation module, as illustrated in figure 3.1 below. We use the CNN model in the recognition module to identify the character on the license plate. Figure 3.1 shows all of the steps with thorough descriptions. In a later subsection, the methods employed at each stage is also covered.

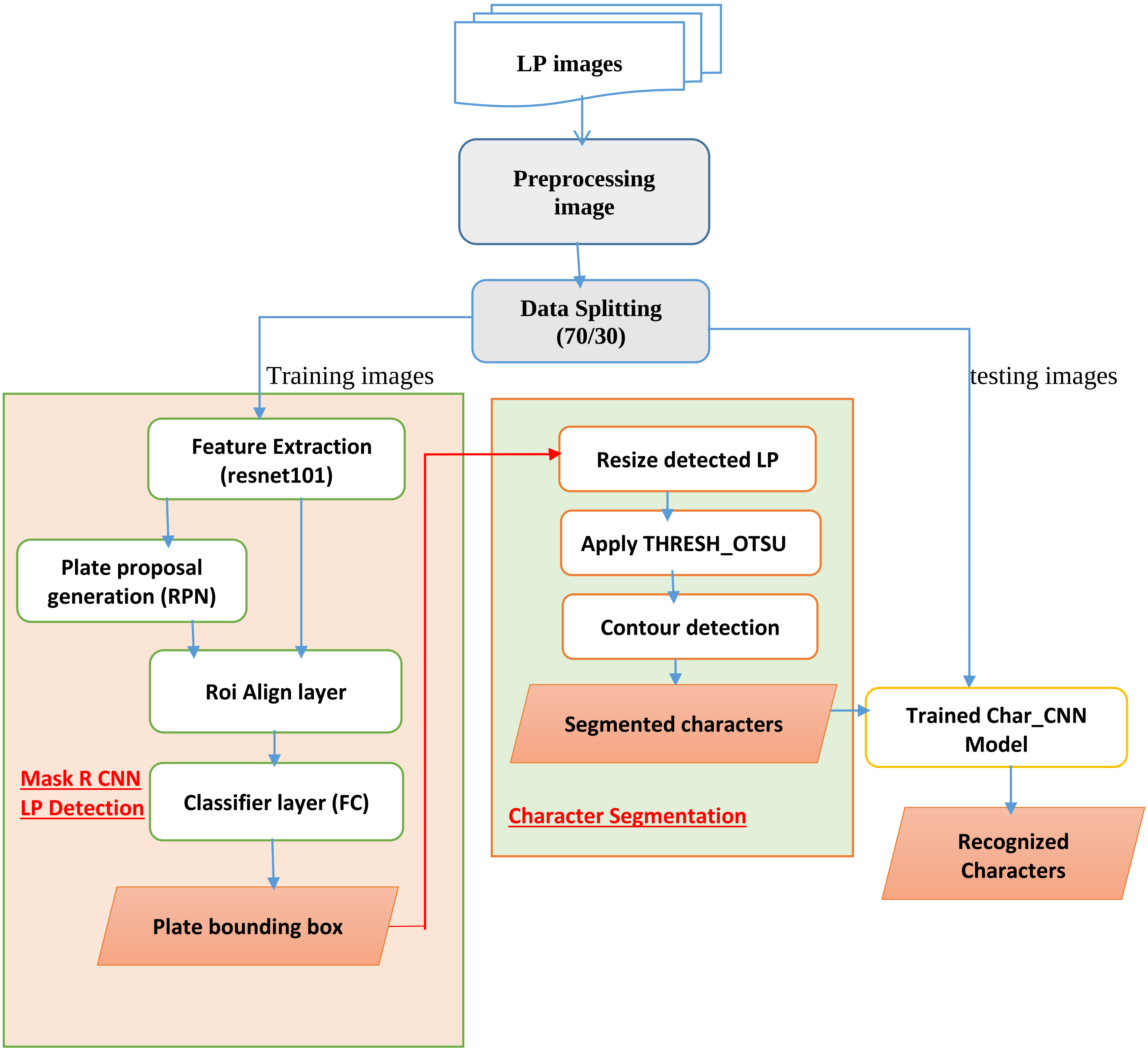
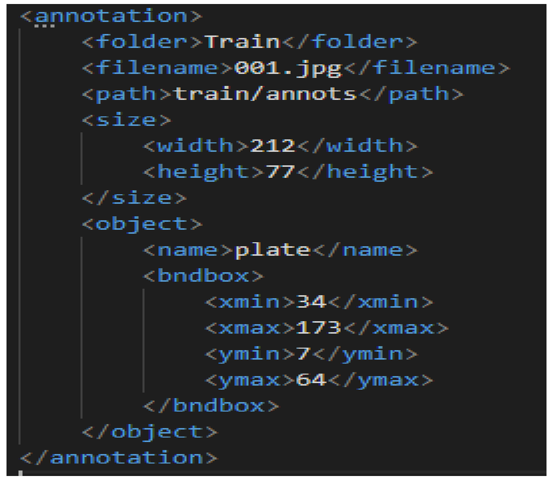


Figure 3. 1: Proposed System Architecture

3.2.1 Image pre-processing

The process of transforming unprocessed data so that a machine learning model can use it is known as data pre-processing [29][32]. Preprocessing enhances the data for subsequent processing. Picture scaling and normalization are examples of pre-processing processes. Resizing the photos (to 224 × 224) and converting them into NumPy arrays that Keras can use are part of the preprocessing step. The most recent models accept an image with dimensions of (224 by 224) as input. Therefore, while comparing our network to the most advanced models, it is practical to choose an image size that is comparable to earlier models. Image scaling and NumPy array conversion are done with OpenCV [35]. After each convolution layer, the input layer should be divisible by two several times. This makes it possible for the.




```xml
<annotation>
    <folder>Train</folder>
    <filename>001.jpg</filename>
    <path>train/annots</path>
    <size>
        <width>212</width>
        <height>77</height>
    </size>
    <object>
        <name>plate</name>
        <bndbox>
            <xmin>34</xmin>
            <xmax>173</xmax>
            <ymin>7</ymin>
            <ymax>64</ymax>
        </bndbox>
    </object>
</annotation>
```


Figure 3.2: Sample annotated plate image using VGG (source: from collected dataset)

After the data annotation, the software generates an XML (Extensible Markup Language) file containing the bounding box information for the license plates, as shown in Figure 3.2. The tool gives the actual starting coordinates (xmin, ymin) and the ending coordinates (xmax, ymax) for each bounding box. In the research, 70-30% train-test split was used to divide the datasets into training and test sets.

3.2.2 License Plate (LP) detection module

Object detection involves drawing a bounding box license plate number in an image [33].License plate detection modules can use different approaches and algorithms depending on the specific requirements and constraints of the application[34]. These modules can be trained using machine learning techniques to improve their accuracy and robustness in detecting license plates under various conditions such as different lighting conditions, angles, or blurriness

We proposed Mask R CNN-based architecture to detect the regions that can contain plate number. The propose model have the capable of handling unconstrained environments, such as varying illuminations conditions, colored background plates, and variations in plate, and font sizes and variations in plate, and font sizes.

To adapt Mask R-CNN for license plate detection, the model is trained on a dataset containing images with labeled license plates. During training, the model learns to detect license plates by iteratively refining the RPN and ROI pooling layers based on ground truth annotations.

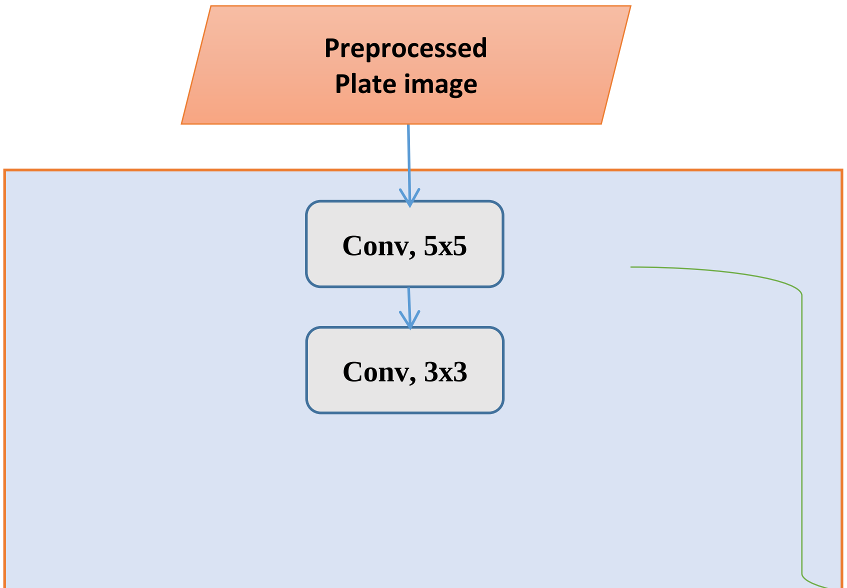
During inference, the trained Mask R-CNN model takes an input image and generates region proposals using the RPN. These proposals are then passed through the ROI pooling layer for classification and segmentation. The model predicts object bounding boxes, class labels (including a specific class for license plates), and instance masks that delineate the boundaries of the license plates

3.2.2.1 Feature Extraction Model

In plate detection module, the feature extractor model is responsible for extracting features from the preprocessed plate image and generating a high-level representation of the image that can be used for object detection. The input image is passed through several convolutional layers, where each layer extracts increasingly abstract and complex features. The deeper layers in the CNN usually capture high-level semantic information, while the earlier layers capture low-level details and spatial information.

The input image is initially passed through a series of convolutional layers. These layers typically have small filter sizes and high stride values, which help to reduce the spatial dimensions of the image while increasing the number of channels or feature maps. As the input image progresses through the feature extractor network, the convolutional layers become increasingly deeper and more complex. These layers are responsible for capturing more abstract and higher-level visual features. They have larger filter sizes and lower stride values, which help to preserve spatial information and capture more detailed features.

In addition to the convolutional layers, we apply pooling layers and normalization layers (e.g., Batch Normalization). Pooling layers downsample the feature maps, reducing their spatial dimensions while preserving their important features. Normalization layers help to normalize the features and facilitate convergence during training.



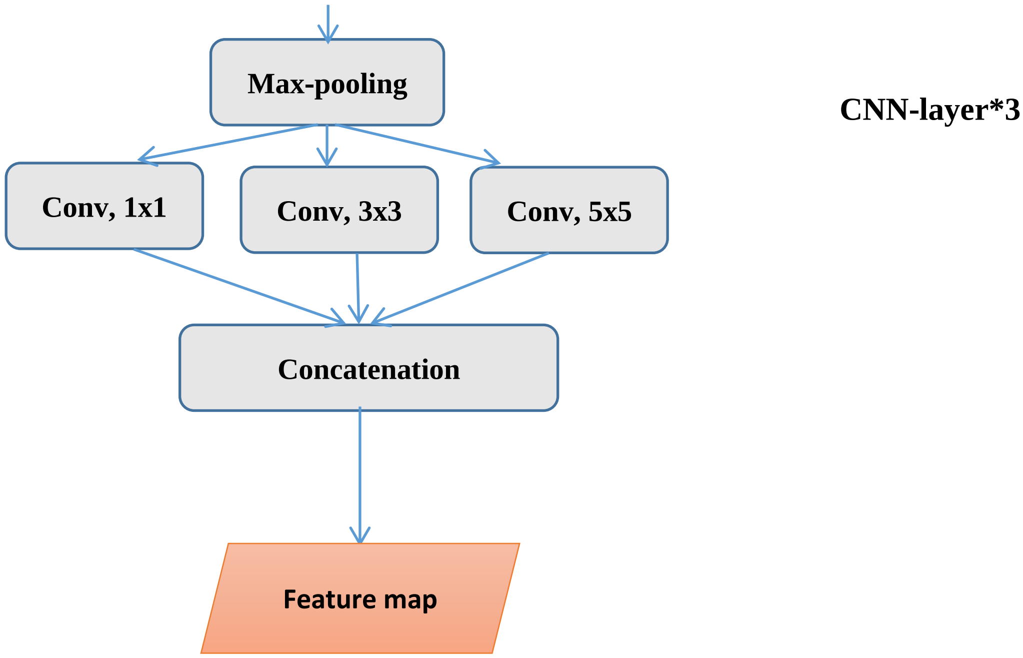


Figure 3. 3: Deep Convolutional Neural Network Architecture for Feature Extraction network

In our context, we use 12 convolution layers in training phase as feature extractor. The input to the first convolution layer is 224 x 224 x 3 image. The convolution operation requires four parameters. The first parameter is the number of filters that are used to control the depth of the output volume. In our model, we use 32, 64, 128, 256, and 512 filters. The numbers of filters we have applied are increased as we go down to the fully connected layers and the Softmax classifier. Second parameter is the receptive field size, which determines the size of each filter (kernel) and is nearly always square. We use 5 x 5, 3 x 3, and 1 x 1 filter size at a single layer. The third parameter is stride size, which determines the number of pixels skipped (horizontally and vertically) each time we make convolution operation. We have used stride size of two (2, 2) and one (1, 1). When stride size is two, image dimension is reduced by half vertically and horizontally. The fourth (last) parameter is the amount of zero-padding, which is used to control the size of the output. We have used "same" padding, which means the size of the output is equal to the size of the input if the stride size is one.

In the convolution module, we use 1 x 1, 3 x 3, and 5 x 5 filter size at the same layer. We select the filter size systematically based on the characteristic features on LP images. It has been select based on width and the length of the LP images.

The feature map is subsequently fed into other components of the model, such as the Region Proposal Network (RPN) which generates region proposals and extracts features for object detection and segmentation.

3.2.2.2 Plate proposal generation: Region Proposal Network (RPN)

In this study, the RPN takes the feature map as input and outputs a set of object proposals, which are candidate regions of interest (RoIs) that may contain objects. To feed the feature map into the RPN, the following steps are typically performed: the first operation is apply a sliding window over the feature map: The feature map is divided into a set of fixed-size regions called anchors, which are used as reference boxes for the object proposals. The size and aspect ratio of the anchors are chosen to match the characteristics of the objects that need to be detected.

For every anchor box, the RPN generates two scores: 1) the probability of the anchor box containing an object, and 2) the regression values for adjusting the anchor box to the true box. To obtain these two scores, the anchor box is classified as "foreground" or "background" based on their intersection-over-union (IoU) overlap with a ground-truth box. If the IoU is greater than a predefined threshold (usually 0.5), the anchor box is labeled as "foreground", otherwise, it is labeled as "background". For the foreground anchor boxes, the regression values are calculated by minimizing the difference between the anchor box and the ground-truth box. The regression values adjust the anchor box to the size and position of the true object. After the scores and regression values are obtained for every anchor box, the RPN selects a set of high-scoring anchor boxes as region proposals and passes them to the next stage of the algorithm, RoI (Region of Interest) pooling

Then, it generate a set of candidate proposals: For each anchor, the RPN applies a set of convolutional filters to the feature map, which produces two outputs: objectness scores and bounding box offsets. The objectness scores indicate whether the anchor contains an object or not, while the bounding box offsets adjust the size and location of the anchor to better fit the object. In the third step, the network filters and refines candidate proposals: The candidate proposals are filtered based on their objectness scores and the top-k proposals are selected for further processing. The proposals are then refined by using the bounding box offsets from the RPN to adjust their size and position.

Finally, we implement Non-maximum suppression (NMS); during object detection, the RPN network may propose multiple bounding boxes that overlap with the same object. NMS is a technique that eliminates redundant bounding boxes and retains only the one with the highest objectness score. This reduces the number of overlapping objects and improves detection accuracy. The refined proposals are further filtered by applying NMS, which removes redundant proposals that overlap significantly with each other. NMS is implemented in case the Intersection over Union (IoU) overlap is 0.5**.**

Table 3. 1: Algorithm for Builds the computation graph of Region Proposal Network

Algorithm for Builds the computation graph of Region Proposal Network.		
Inputs	feature map	backbone features [batch, height, width, depth]
	anchors_per_location	number of anchors per pixel in the feature map
	anchor stride	Controls the density of anchors. Typically 1 (anchors for every pixel in the feature map), or 2 (every other pixel)
Process	Generate anchor	Create a set of anchor boxes at different scales and aspect ratios over the feature map. - Each anchor box is defined by its center coordinates (cx, cy) and its width (w) and height (h).
	Apply Convolutional Layers	Pass the feature maps through a series of convolutional layers with ReLU activation. - These layers aim to capture spatial context and relationships within the feature maps
	Regression head	Apply a convolutional layer to predict the bounding box adjustments (delta_x, delta_y, delta_w, delta_h) for each anchor box. - The output of this layer is a set of bounding box regression offsets for each anchor.
Returns	rpn_bbox	batch, H, W, (dy, dx, log(dh), log(dw))] Deltas to be applied to anchors

RoI Align layer

In this study, we use RoI align layer which takes input from feature extractor network and RPN. We use RoI align layer for keep the size of the feature map the same so that the same sub-network can be used to predict class, mask and regress bounding box, focus on translation variance - the location of the object matters, and to avoid quantization that causes misalignment

.

For each region proposal, the RoI Align layer applies a pooling operation to the feature maps. Unlike the standard max-pooling, which applies a fixed-size filter to a fixed-size region of the feature map, the RoI Align layer introduces sub-pixel precision by allowing fractional offsets inside the cells of the pooling grid. This means that the output of each region proposal pool has a fixed spatial extent that corresponds to a user-defined resolution (7x7), but the pooling operation is performed on a finer-grained grid to allow for sub-pixel alignment.

To extract features from the sub-pixel-aligned regions, we applies bilinear interpolation to the feature map values inside the pooling cells. This interpolation scheme computes weighted averages of the feature map values at the four corners of each cell, based on the fractional offsets of the pooling grid relative to the feature map grid. The output of the RoI Align layer is a fixed-size tensor for each region proposal, with dimensions that match the resolution of the RoI pooling grid. This tensor contains the features that correspond to the region proposal, and can be used for further object detection or segmentation tasks.

The output of the final fully-connected layer is given as input to the prediction of bounding box of the plate number that is used to perform character segmentation. In each of the images above, you can see that we have clearly found the license plate in the image and drawn a green bounding box surrounding it.

Figure 3. 4: sample detected License plate by our model

3.2.3 Character Segmentation

The character segmentation process typically involves segmenting character from the image to enhance the LP region and removing any noise or unwanted objects. Thus, for each object detected, the model generates a corresponding segmentation mask that indicates which pixels in the image belong to the object.

To segment characters in a license plate, we can follow these steps:

1. Preprocess the detected License Plate Region: Extract the region of the license plate from the original image. We Apply preprocessing techniques such as resizing, normalization, and image enhancement (e.g., contrast adjustment, noise removal) to enhance the quality of the license plate region.

2. Thresholding: To create a binary image from the license plate region, use the Thresholding approach. This procedure aids in separating the background from the characters. In addition to extracting the license plate characters, we also excise a ton of uninteresting "stuff," such any bolts anchoring the license plate to the vehicle, branding logos on the plate, and embellishments on the license plate frames. We do this by applying simple thresholding. These three factors have the potential to interfere with our character segmentation method. To differentiate the license plate characters from the background, we employ adaptive thresholding, which applies thresholding to each local 28 × 28 pixel region of the picture.

Figure 3. 5: steps of character segmentation output by our model

After an image was represented in binary, we would often employ. To isolate individual characters from any related regions, we use morphological processes like erosion and dilation. These actions aid in character boundary refinement, noise reduction, and filling in spaces between characters. Determine the separated characters' outlines. In the binary image, contours stand in for the borders of the connected regions. Remove contours that don't match the characters by applying filters to certain parameters like size, location, and aspect ratio. By removing erroneous positives (such as noise and artifacts), this stage helps to save only the candidate character outlines.

For each character, crop a bounding box around its contour to extract it separately.

Figure 3. 6: Separately cropped Plate character by model

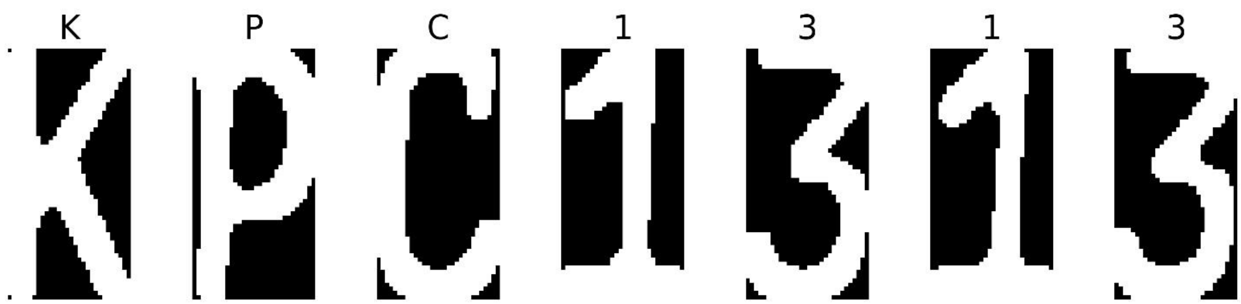
3.2.4 Character Recognition

After character segmentation, we focus on character recognition in this part. An essential step in the Automatic License Plate Recognition and Detection system is character recognition, which is the process of extracting the characters from the detected license-plate region.

To develop character recognition model, we acquire a large dataset of labeled character images. This dataset should include examples of all the characters you want to recognize. Ensure that the images are properly labeled with their corresponding characters. Then, before training, it's important to preprocess the dataset. We apply resizing the images to a consistent size, normalizing pixel values, and augmenting the data by applying transformations like rotation, scaling, or blurring. Finally, we input the processed character images into CNN model.

A character recognition model typically consists of several layers, each with a specific function in extracting features and classifying characters. Here is an overview of the layers commonly used in a character recognition model: We also apply Batch Normalization or Dropout layers used to improve the performance or generalization of the model. Here under figure 3.7 presents sampled character recognition CNN model.

Finally, the model predicts the recognized character as following way.



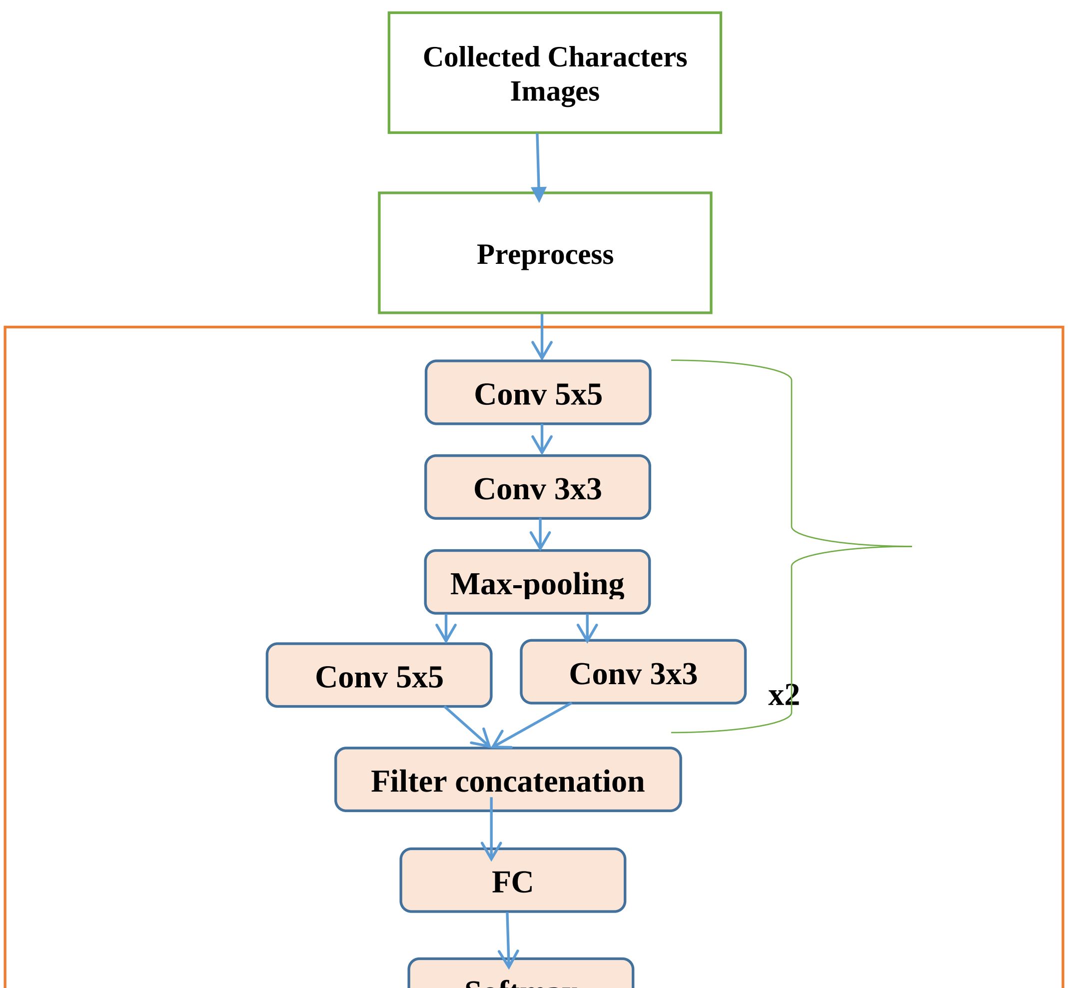


Figure 3. 7: Character recognition CNN model

3.3 . Evaluation methods

The designed ALPR system's accuracy was tested using the standard Accuracy metric [14].

Accuracy: The ratio of correctly predicted class to total instances is known as accuracy. When each class in the datasets has an equal number of instances, this method of determining accuracy is most well-known. Displays the equation to determine the basic accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

Precision: The ratio of accurately anticipated positive observations to all predicted positive observations is known as precision. The confusion matrix is used to calculate the precision.

$$Precision = \frac{TP}{TP + FP}$$

Recall: A recall is a ratio between the correctly predicted positive observations to the observations in the actual class. The recall from the confusion matrix is calculated using Equation 32

$$Recall = \frac{TP}{TP + FN}$$

Where:

TP – number of true positives (number plate is presented and detected);

TN – true negatives (number plate isn't presented and isn't detected, not applicable to the problem, equals zero);

FP – false positives (number plate wasn't presented, but was detected);

FN – false negatives (number plate was presented but not detected).

Chapter four

Experimental result and discussion

4.1 Introduction

This chapter provides a detailed description of the experimental evaluation of the proposed model for license plate detection and recognition. The proposed model or architecture is approved for realization based on experimental examination. There is a detailed description of both the model's implementation and the dataset that was used. Furthermore, the test results are shown and contrasted with the most recent models.

4.2 Dataset description

First, for the study, about 1190 images of license plates were gathered from all throughout the country. Prior to annotating the datasets, every vehicle image needs to have a standard extension, such "JPG." The original photos were split into 224x224 pixel sub-images while maintaining the same pixel density in an attempt to lower the total number of training images that needed to be obtained. A random selection of the acquired images with different lighting levels, occlusions, and overlapping license plates was made from this cache in order to train the neural network. .

After the data annotation, the software generates an XML (Extensible Markup Language) file containing the bounding box information for the vehicle and license plates, as shown in Figure 5.1. The tool gives the actual starting coordinates (xmin, ymin) and the ending coordinates (xmax, ymax) for each bounding box. In the research, 70-30% train-test split was used to divide the datasets into training and test sets.

Table 4. 1: Dataset Description

Dataset Description			
No.	**Total collected dataset**	**Training images**	**Testing images**
1.	1,190(plate images)	833	357
2.	18,423 (character image)	12,600	5823

Figure 4. 1: Sample Annotated plate Image Using VGG

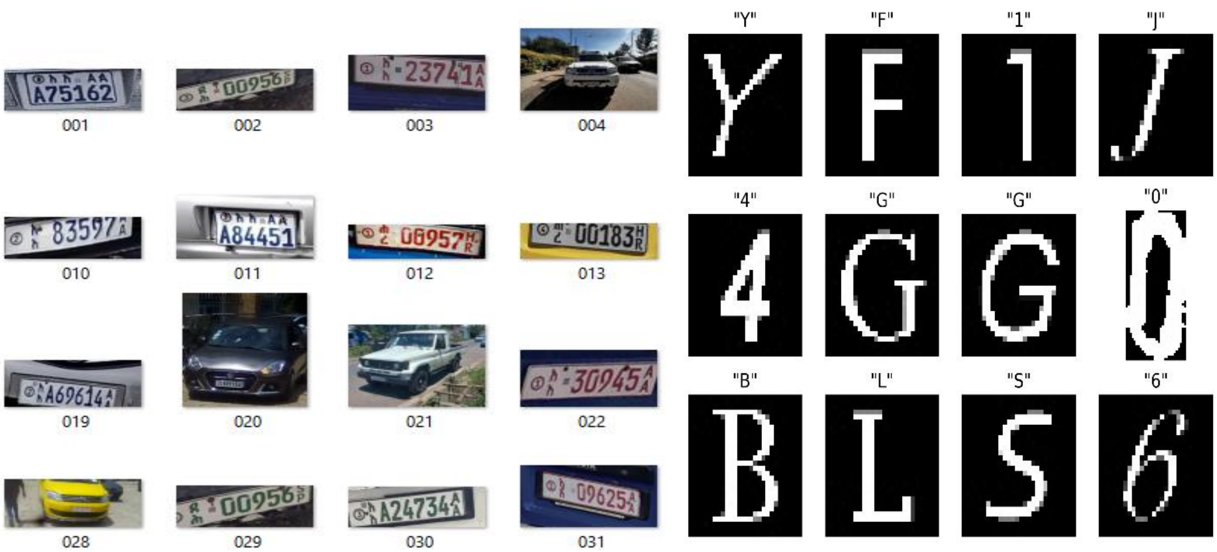


Figure 4. 2: Sample collected plate images and characters

4.3 Implementation

Experiments are done based on the prototype developed with Keras (TensorFlow as a backend) on Intel Core ™ i5-6200 CPU, and 4 GB of RAM. The model is trained for 10 epochs, a batch size of 32, and a starting or initial learning rate of 0.001 (1e-3). The data is partitioned into training and testing dataset such that 70 percent of the data is assigned for training the model and 30 percent of the data is allotted for testing. Allocating 2/3rd of the dataset [32] for training is close to optimal for reasonable sized datasets (greater than 100 images)

4.4 License plate detection

Since every other stage in the LPR system depends only on the accuracy of the license plate detection system is regarded as one of the important components. Keeping this in mind, the process of detecting license plates from the image was carried out initially in order to eliminate false positives that were caused by items that shared similar features with license plates. The task of automatically detecting license plates within the car image was completed by using Mask R CNN. 1190 vehicle images were gathered for the tests and training. The model was trained for 10 iterations in the experiments, after training was finished, the model was assessed to ensure the test's accuracy.

Figure 4. 3: Experimental results for detection of License plate

4.5 Character segmentation

The retrieved license from the detection step was used as an input to separate the characters for recognition during the character segmentation phase. The license plate was improved using a few preparation techniques, including grayscale conversion and thresholding, before the characters were segmented. Next, using connected component analysis, the characters were separated from the license plate. The performance and outcomes of the character segmentation approach have been covered in this section.

Performance evaluation of segmentation

To evaluate the character segmentation, we have used 50 license plates that were recovered from the detection stage were used as the input for the character segmentation phase evaluation. The retrieved license plates were first preprocessed before the characters were segmented. Only 44 license plates were correctly separated out of 50 retrieved license plates from the detecting process. The character segmentation algorithm's overall accuracy was 88%. As seen in Figure 5.4, the inaccuracy is the result of faded license plate characters, low resolution license plate images, and some characters that have fused together. A few characters were connected to the edge of the license plate.

Figure 4. 4: incomplete segmentation by our model

4.6 Character recognition model evaluation

The final phase of license plate detection and recognition system is character recognition. The accuracy of detection of license plates and the subsequent segmentation of individual characters are prerequisites for the accurate recognition of license plate characters.

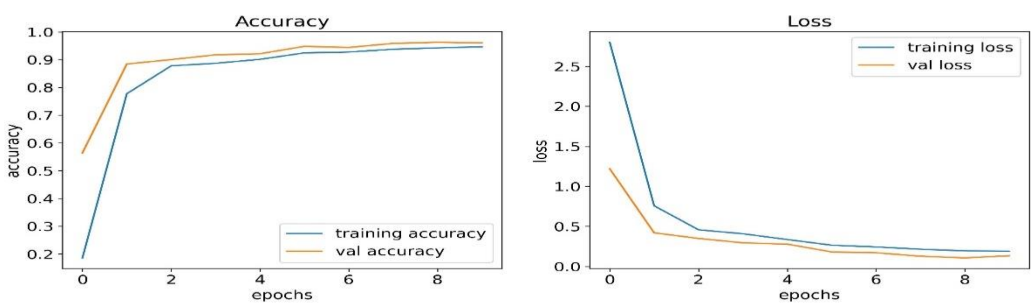


Figure 4. 5: Training and validation accuracy

The character recognition algorithm's overall accuracy was 95% and 96% training and validation accuracy respectively. Parallel, the training and validation loss decrease from 2.5 up to 0.01. As seen in Figure 4.5. Ultimately, the accuracy of a character recognition model can be improved by carefully curating the dataset, applying appropriate data preprocessing techniques, experimenting with different model architectures, and fine-tuning hyper parameters to achieve the best possible performance.

Chapter five

Conclusion and recommendation

5.1 Conclusion

The definition of issue statements and a thorough background of the study were covered in the first chapter. The study's primary goals were to create an LPR system and a feature extraction technique appropriate for the circumstances of our country plate. Three steps make up a typical LPR system: segmenting characters, detecting license plates, and recognizing characters. In our method, we localized the license plates while reducing the false positives by licensing detection. A thorough review of the literature was done in the second chapter. This study looks into the automatic localization of license plates using the state-of-the-art Mask R CNN object detector.in the study we collected 1190 plate images were fed into the system to assess the method's effectiveness, and the approach produced a 96.% accuracy rate for license plate recognition. An accuracy of 88 %was attained when all the recognized license plates were assessed to see how well the segmentation approach worked.

5.2 Limitation and future work

The system performs satisfactorily in every LPR stage, but because it was created under particular presumptions, it has some limitations. The following are the limitations:

1. Because license plate records are absent, data was collected using a smartphone.
2. The model was trained on less datasets.
3. The classifier only employed 50 characters.
4. To develop a classifier for character recognition, we manually designed the features especially size of each character images.
5. The characters written in different fonts are not being recognized by the system with ease.

Therefore, we want to give a suggestion for future work as follows:

1. To achieve greater precision in the future, the datasets must be gathered with a high-resolution camera.
2. To make the model more robust in identifying license plates and vehicles, the training dataset should be expanded.
3. To make it easier to identify a car registered with new characters, the character classifier must be designed using all 26 alphabets 10 numerals, and 36*7 Amharic characters.

Reference

[1] W. Puarungroj and N. Boonsirisumpun, "Thai License Plate Recognition Based on Deep Learning," *Procedia Comput. Sci.*, vol. 135, pp. 214–221, 2018.

[2] M. Shraddha, S. Ghadage, M. Sagar, and R. Khedkar, "Automatic Number Plate Recognition System Using Deep Learning Techniques," *Int. Res. J. Eng. Technol.*, no. 200101068, 2020.

[3] P. R. Sanap and S. P. Narote, "License Plate Recognition system-survey," *AIP Conf. Proc.*, vol. 1324, pp. 255–260, 2010.

[4] Z. Selmi, M. Ben Halima, U. Pal, and M. A. Alimi, "DELP-DAR system for license plate detection and recognition," *Pattern Recognit. Lett.*, vol. 129, pp. 213–223, 2020.

[5] M. A. Jawale, P. iam, A. B. Pawar, and N. Marriwala, "Implementation of number plate detection system for vehicle registration using IOT and recognition using CNN," *Meas. Sensors*, vol. 27, no. May, p. 100761, 2023.

[6] D. A. Ramasamy* and M. J. Wondwosen, "Deep Learning Based Ethiopian Car's License Plate Detection and Recognition," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 5730–5737, 2020.

[7] Addis Ababa City Roads and Transport Bureau, "Ethiopia Transportation System Improvement Project," no. February, pp. 1–148, 2016.

[8] T. Bireda, *Intelligent transport system in Ethiopia: Status and the way forward*, vol. 244. Springer International Publishing, 2018.

[9] A. Mhatre and P. Sharma, "INTELLIGENT SYSTEMS AND APPLICATIONS IN Deep Learning Approach for Vehicle Number Plate Recognition System with Image Enhancement Technique," vol. 11, pp. 251–262, 2023.

[10] J. Wang, X. Liu, A. Liu, and J. Xiao, “A deep learning-based method for vehicle licenseplate recognition in natural scene,” *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–7, 2019.

[11] A. Bhujbal and D. Mane, “A survey on deep learning approaches for vehicle and number plate detection,” *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 1378–1383, 2019.

[12] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, “Automated license plate recognition: A survey on methods and techniques,” *IEEE Access*, vol. 9, pp. 11203–11225, 2021.

[13] V. Patidar and S. Nadkarni, “Vehicle Number Plate Detection and Recognition,” vol. 3, no. 2, pp. 790–794, 2017.

[14] R. V Patel, A. Prof, and S. J. Swarndeep, “Literature Survey on Vehicle Recognition for Augmented Reality Using Android Smart Phones,” pp. 1552–1555, 2020.

[15] “EQ-LPR : EFFICIENT QUALITY-AWARE LICENSE PLATE RECOGNITION Cong Zhang , Qi Wang ∗ , Xuelong Li School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),” no. 2.

[16] A. Patel, M. Krunal, and R. Patel, “An Introduction to License Plate Detection System,” *J. Eng. Res. Appl. www.ijera.com*, vol. 4, no. 1, pp. 216–222, 2014.

[17] A. Ammar, A. Koubaa, W. Boulila, B. Benjdira, and Y. Alhabashi, “A Multi-Stage Deep-Learning-Based Vehicle and License Plate Recognition System with Real-Time Edge Inference,” *Sensors*, vol. 23, no. 4, 2023.

[18] M. A. Anoop, S. Poonkuntran, and P. Karthikeyan, “Deep learning-based Indian vehicle number plate detection and recognition,” *Object Detect. with Deep Learn. Model. Princ. Appl.*, vol. 6, no. 5, pp. 117–135, 2022.

[19] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, “Automatic number plate recognition using deep learning,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1084, no. 1, p. 012027, 2021.

[20] T. H. E. Law and A. Ababa, “Road Traffic Law Enforcement in Addis Ababa : the Law and Practice,” no. July, 2008.

[21] H. Zuber, “Design and Implementation of Car Plate Recognition System For Ethiopian Plates Addis Ababa University School of Graduate Studies Addis Ababa Institute of Technology Design and Implementation of Car Plate Recognition System for Ethiopian Car Plates,” *Addis Ababa Univ. Institutional Repos.*, pp. 1–62, 2011.

[22] H. Demma and D. M. Kaur, “Automatic Ethiopian Vehicle Number Plate Detection System using MATLAB,” *Int. J. Eng. Res. Technol.*, vol. 8, no. 06, pp. 716–719, 2019.

[23] S. Nigussie and Y. Assabie, “Automatic recognition of Ethiopian license plates,” *IEEE AFRICON Conf.*, vol. 2015-Novem, 2015.

[24] Y. Shambharkar, S. Salagrama, K. Sharma, O. Mishra, and D. Parashar, “An Automatic Framework for Number Plate Detection using OCR and Deep Learning Approach,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, pp. 8–14, 2023.

[25] Z. Xu *et al.*, “Towards end-to-end license plate detection and recognition: A large dataset and baseline,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11217 LNCS, pp. 261–277, 2018.

[26] N. A. Alam, M. Ahsan, M. A. Based, and J. Haider, “Intelligent System for Vehicles Number Plate Detection and Recognition Using Convolutional Neural Networks,” *Technologies*, vol. 9, no. 1, pp. 1–18, 2021.

[27] L. J. Fellow and J. I. N. Zhao, “A Survey on the New Generation of Deep Learning in Image Processing,” *IEEE Access*, vol. PP, p. 1, 2019.

[28] Lubna, N. Mufti, and S. A. A. Shah, “Automatic Number Plate Recognition : A Detailed Survey of,” *Sensors*, vol. 21, no. 9, p. 3028, 2021.

[29] L. Alzubaidi *et al.*, *Review of deep learning : concepts , CNN architectures , challenges , applications , future directions*. Springer International Publishing, 2021.

[30] R. Girshick, “Fast R-CNN,” pp. 1440–1448.

[31] G. lekhana, and R.srikantoswamy, professor, “Real Time Licence Plate Recognition System”, IJATERS, Vol. 2, p. 8, July, 2012.

[32] V. Gnanaprakash et al , “Authomatic number plate recognition using deep learning” , IOP Conf. series: Material Science and Enginnering, 2021.

[33] Chen Chunyu,Cheng Baozhi,Chen xin,Wand Fucheng,Zhang Chen, “Application of image processing to the vehicle license plate recognition”, Published by Atlantis press,paris,France,2013.

[34] Sarbjit Kaur, “An Automatic Number Plate Recognition System under image processing”,Modern Education and Computer Science,p. 14- 25, 2016.

[35] Zied Selmi, Mohamed Ben Halima, Adel M. Alimi, “Deep Learning System for Automatic Licence Plate Detection and Recignition ”, International Conference on Document Analysis and Recognition,P. 1138-1138, 2017.

[36] Bakshi, Akshay & Gulhane, Sudhanshu & Sawant, Tanish & Sambhe, Vijay & Udmale, Sandeep. (2023). ALPR - An Intelligent Approach Towards Detection and Recognition of License Plates in Uncontrolled Environments. 10.1007/978-3-031-24848-1_18.

[37] Parul Agarwal, Kritika Chopra, Mohd Kashif, Vinita Kumari,Implementing ALPR for detection of traffic violations: a step towards sustainability,Procedia Computer Science,Volume 132, 2018

[38] N. Hezil, A. Amrouche and Y. Bentrcia, "Vehicle license plate detection using morphological operations and deep learning," *2022 International Conference of Advanced Technology in Electronic and Electrical Engineering (ICATEEE)*, M'sila, Algeria, 2022, pp. 1-6, doi: 10.1109/ICATEEE57445.2022.10093752

[39] Pirgazi, Jamshid & Pourhashem Kallehbasti, Mohammad Mehdi & Ghanbari Sorkhi, Ali. (2022). An End-to-End Deep Learning Approach for Plate Recognition in Intelligent Transportation Systems. Wireless Communications and Mobile Computing. 2022. 1-13. 10.1155/2022/3364921.

[40] F. Schirrmacher, B. Lorch, A. Maier and C. Riess, "Benchmarking Probabilistic Deep Learning Methods for License Plate Recognition," in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 9, pp. 9203-9216, Sept. 2023, doi: 10.1109/TITS.2023.3278533

[41] Zhao, Kemeng & Peng, Liangrui & Ding, Ning & Yao, Gang & Tang, Pei & Wang, Shengjin. (2023). Deep Representation Learning for License Plate Recognition in Low Quality Video Images. 10.1007/978-3-031-47966-3_16.

[42] Patil, Atharvraj & Ingale, Jayesh & Upadhye, Aditya & Jaiswal, Sakshi & Kakade, Suhas & Bhatt, Abhishek. (2023). Deep Learning-based approach for Indian License Plate Recognition using Optical Character Recognition. 1-5. 10.1109/GCON58516.2023.10183391.

[43] X. Pan, S. Li, R. Li and N. Sun, "A Hybrid Deep Learning Algorithm for the License Plate Detection and Recognition in Vehicle-to-Vehicle Communications," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 23447-23458, Dec. 2022, doi: 10.1109/TITS.2022.3213018

[44] Huynh, Anh & Tan, Le & Huynh, Kha-Tu. (2022). License Plate Detection and Recognition using CRAFT and LSTM. 10.1007/978-3-031-29104-3_31.

[45] Pan, Shenghu & Liu, Jian & Chen, Dekun. (2023). Research on License Plate Detection and Recognition System based on YOLOv7 and LPRNet. Academic Journal of Science and Technology. 4. 62-68. 10.54097/ajst.v4i2.3971.

[46] Chalnewad, Surekha & Manjaramkar, Arati. (2023). Detection and Classification of License Plate by Neural Network Classifier. 531-535. 10.1109/IDCIoT56793.2023.10053459.