License Plate Recognition System

Deepali Zutshi, Poojitha Katta, Yasaman Emami and Praveen Kuruvangi Parameshwara

San José State University

DATA-270: Data Analytics Process

Dr. Eduardo Chan

March 5, 2022

1. Introduction

1.1 Project Background and Execute Summary

(Automatic number-plate recognition, 2022) The British Police Scientific Development Branch first invented License Plate Recognition Technology in 1976. (Introduction to License Plate Recognition Software, n.d.) The prototypes that were developed before this were low accuracy readings and functioned only under relative laboratory conditions and it was not suitable for real world applications. The technology was not good enough to cope up with the speed and performance. Systems were restricted to CPU and memory limits. The video footage was captured and later it was used for analysis. The License plate number was identified after processing the video.

Later, moderate improvements were made and created the working models to satisfy the requirements and implemented in Wokingham, England. The working models were not efficient and accuracy was below 60%. Several decades later, with the evolution of technology and highly effective systems the prior limitations incurred from vehicle speed, light fluctuation, angular skew, character segmentation and recognition have been solved with today's technology resulting in the Automatic License Plate Recognition (ALPR) system.

Every year more number of vehicles are landing on the road and several hundreds or thousands of license plates should be scanned to detect any crime happening at any place. The repetition of crimes or issues were happening such as bad license plates, smuggling, unpaid parking tickets, use of the vehicle for abduction, expired registration, unauthorized vehicle access, toll collection and car theft. Police officers need to pay attention both on driving and also on the vehicle to view the license plate numbers. More manual work was needed to enter the details in the computer or files. The action on the crime or cases were taking much longer time as there were a lot of dependencies on recognizing and passing the information for the next course of action. This gave rise to a need for a system that would detect license plates and extract the number from the image automatically.

An ALPR system makes use of a camera to click an image of the vehicle, applies OCR technique to get a string of alphanumeric characters that form the license plate and fetch the vehicle details from the database or “hot-list”. An OCR system works by scanning the image into light and dark areas. The areas that are dark are associated to be the text whereas the areas that are light are usually thought to be the background. Once these areas are mapped, the system then tries to identify numbers and alphabets in the image. OCR makes use of either feature recognition or pattern to identify the detected numbers and alphabets.

The pattern recognition approach works by learning from examples. The algorithm is trained to recognize text in different fonts and styles, and then compares the input image to recognize the various alphanumeric characters in the image. Feature recognition, on the other hand, works by applying a set of rules to the characters to correctly identify them. For example, the letter ‘B’ may be stored as a single vertical line with two semi-circles attached to it and the letter ‘P’ may be stored as a single vertical line with just one semi-circle attached to it.

A camera with high resolution and shutter speed is required to compensate for the velocity of the vehicle. It is expected that the camera shutter speed of 1/1000 sec is beneficial for LPR. Eclipse cameras will be helpful to reduce the headlight glares and also the light reflected from the vehicle. Infrared Night Vision Cameras are very effective in the area of low lights or total darkness because they see in the infrared spectrum. In some of the conditions, an infrared camera requires additional infrared illumination if the distance of the vehicle is more than 15 to 30 ft. The patterns are recognized from the image captured from the camera.

We have datasets with different format JPEG, PNG and XML. We are using Python-Tesseract, an OCR tool, and the YOLO model makes use of Darknet, an open source neural network framework.

1.2 Project Requirements

1.2.1 Functional Requirements

This section specifies what our application must perform. And we test it to see whether it is performing what it was designed and intended to do. In the initial stage, the application must accept a photo. It must inspect the image to see whether there is a car in it. If there is no car or the angle of the automobile is incorrect, it must notify the user of the incorrect input. The lighting in the image must be examined to determine whether the color of the automobile can be retrieved. The car's dimensions must be measured. The angle between the camera and a specified place in the vehicle must be calculated (which different angles might outcome to different dimensions). The license plate from the picture must be extracted. The characters of the plate must be returned as text output to the user. For many applications, functional testing is the most significant type of testing. Improving performance before that point is pointless.

1.2.2 AI-powered Requirements

Building a model using machine learning techniques necessitates a massive quantity of data in order to train a credible model. Because we will be employing supervised learning in this study, we will divide the acquired data from several datasets and devote a large portion of it to a training model while the remainder will be used for testing and validating our model. We will shuffle data from many datasets so that if the images in one dataset are similar, the model will be able to be generalized. In this study, version control tools would be used to simply track the process of constructing and comparing models. Following the construction of each model, we assess accuracies based on overall accuracy, and to be more detailed, we measure confusion matrix, specificity, and sensitivity. Using methods like McNemar's test, would allow us to compare metrics from multiple matrices. For the purposes of this study, we are implementing YOLO-v3, a deep learning approach for extracting and detecting an automobile in a picture, as well as plate identification. Tesseract, which is an Optical Character Recognition (OCR) would be examined for the purpose of text extraction from plate .

1.2.3 Data Requirements

Static data from multiple sources was used in this study, including: Labeled License Plate Dataset from Kaggle published in 2020, License Plate Annotated Image Dataset from Robotflow published in 2021, Car License Plate Detection from Kaggle published in 2020, Vietnamese Plate Dataset from Github published in 2020, Indian License Plate Dataset from Kaggle published in 2020, Cars Dataset published in 2013, Vehicle Make and Model Recognition Dataset (VMMR) published on Github published in 2013.

All of these databases gathered license plates with various lighting, automobile kinds, and angels.

The picture files as input are then transformed to matrices for further calculation; all of these datasets would collect large amounts of data, making them suitable for training models.

1.3 Project Deliverables

The following deliverables have been identified and would be implemented during this project. Each deliverable has its own metrics such as description, timeline, requirements, etc.

1.3.1 Project Plan

Description. A detailed description of the stages of execution of the project following the CRISP-DM approach which would involve the following steps:

- Business Understanding,
- Data Understanding,
- Data Preparation,
- Modeling,
- Evaluation,
- Development

Time (in weeks). 3

Requirements. The creation of a project plan would require a list of project objectives and their own requirements, a detailed schedule, and definition of the roles and responsibilities of each team member.

1.3.2 Project Report

Description. The project report would encapsulate the goals and objectives of the project as well as help keep track of the progress of the project. It would describe the landmarks along with the issues and risks associated with it, a schedule for the various stages and how the resources would be allocated to each of those stages.

Time (in weeks). 10

Requirements. An effective project report would require clearly defined stages, goals, a list of risks and issues, cost breakdown summary, a realistic project timeline, and a resource allocation plan.

1.3.3 Work Breakdown Structure (WBS)

Description. A Work Breakdown Structure or WBS is a tool for the successful implementation of any multi-stage project by breaking down large goals into smaller tasks to effectively manage a large-scale project.

Time (in weeks). 1

Requirements. A good WBS must have a list of all the tasks involved in the project, the subtasks associated with each of them and the corresponding requirements for all the subtasks.

1.3.4 Prototype

Description. The project requires the implementation of prototypes, namely, the creation of an Optical Character Recognition system or OCR and the You Only Look Once or YOLO model.

Time (in weeks). 6

Requirements. The implementation of an OCR system would require the use of Python-Tesseract, an OCR tool, and the YOLO model makes use of Darknet, an open source neural network framework.

1.3.5 Test Cases

Description. Test cases define the various inputs under different conditions to test whether the system achieves the desired results and determine where it fails to help overcome them.

Time (in weeks). 5

Requirements. For this particular model, we would require images of cars with their license plates under different conditions. For example, images of cars in motion, in dimly lit conditions, blurred images, images of cars with customized plates, etc.

1.3.6 Final Model

Description. The final model would include an Optical Character Recognition system implemented along with the You-Only-Look-Once model that would take as input an image of a vehicle with its license plate, find the position of the plate in the image, segment the different alphanumeric characters, recognize them and give as output the plate number of the vehicle.

Time (in weeks). 10

Requirements. The model would require Python, YOLO model with darknet, Python-Tesseract, datasets with images of cars in different conditions and an IDE for the implementation.

1.4 Technology and Solution Survey

The main objective of License Plate recognition is to rapidly detect the location of the license plate from fast moving vehicles and to recognise the characters. This activity is subjected to many taxing conditions in the real world. Such as illuminating changes due to environmental conditions, blurry images due to motion, night time contrast issues, occlusion effects etc., should be handled to produce optimal results.

The process of license plate detection is broadly classified into Multi-stage and Single-stage License plate recognition systems.

1.4.1 Multi-Stage Recognition System

Most of the current ALPR solutions use the multi-stage approach. This system is divided into two stages, i.e. License Plate Detection and License Plate Recognition. This first stage uses traditional computer vision based feature extraction techniques. It focuses on features such as color, shape, size, texture, symmetry etc. The second stage is further divided into three parts namely Image processing, Character Segmentation and Character Identification. Certain systems skip the Character Segmentation step as it makes use of segmentation-free algorithms. The identification step uses traditional deep learning algorithms like Neural Networks and Fuzzy Classifiers to identify the characters of the license plate. The model's efficiency is mostly determined by this step as it is most likely to be subjected to errors in the recognition process.

1.4.2 Single-Stage Recognition System

The single stage ALPR system is a compact attempt to avoid various stages and computes the results in one go using fewer parameters to produce high efficiency. Such systems use a Single Deep Neural Network algorithm that makes use of the correlation between object detection and recognition in a single pass. This is done by using a convolution neural network model - VGG16. This is a feature extractor that is modified to use two convolutional layers and two rectangular filters. Once the features are obtained, they are concatenated to classify if they are license plates or not. This is called the Regional Proposal Network (RPN), as it focuses only one the number plate region of an image.

1.4.3 License Plate Detection

This section emphasizes on the detection techniques (as it talks about features and applicable scenarios). R. Zunino and S. Rovetta (2000) stated that the license plate by definition is “a rectangular area of a vehicle with a high density of horizontal and vertical edges” and this forms the base to define features for license plate detection by many algorithms.

Edge-Based. This technique takes into consideration the fact that a license plate is unique to the vehicle’s body considering the color as well, with known aspect ratios that are mostly the same universally (algorithms are designed to handle the nuances in different countries as well), and the boundaries of the license plate which are known as edges. Luo et al. (2009) have used a Sobel filter that extracts the edges of the license plate. They found out that detecting both horizontal and vertical edges provides 99.45% accuracy than using only horizontal edge detection. Duan et al. (2005) came up with an approach using Hough transformation that detects accurately upto an inclination of 30° but this method is not suitable for blurry images.

Color-Based. This technique uses the Hue, Lightness, and Saturation (HLS) color model developing upon the traditional Red, Green and Blue (RGB) color model. Algorithms such as Genetic Algorithm to identify the license plate color, Gaussian Weighted Histogram makes use of the weighted intersection of colors, Mean Shift Algorithm captures only the region of interest, were traditionally used. Wang et al. (2008) modified the HLS model to use HSV (Hue, Saturation and Value) model to address the real-time scenarios such as environmental changes , illumination changes etc. This determines the probability of detecting accurately using the position of the values in the Fuzzy Set.

Texture-Based. This method is useful to handle the color difference in the license plate between the plate’s background and the characters. Transforming the images to grayscale to extract the characters of the image as most part of the plate’s background can be omitted. Zunino et al. (2000) has implemented this using Vector Quantization (VQ). To address the irregularities in the image, they suggested the Sliding Concentric Windows (SCW) technique that segments the image based on texture. Gabor filters will further analyze texture in various directions to extract the characters accurately . Hsieh et al. (2005) suggested Wavelet Transform method which identifies a horizontal line around the feature that is used as a reference line dividing High-Low frequencies (HL) in the image for detection.

Character-Based. Matas and Zimmermann (2005) suggested a Neural Network (NN) based classifier that recognises and classifies a linear spatial region. Further, the contrast of the plate is increased to differentiate the plate and the characters. Cho et al. (2011) identified the inter-character distance to extract characters and achieved a detection rate of 99.5%.

Statistical Classifiers. To address the illumination challenges Haar-like features are used to feed into cascaded classifiers using Adaptive Boosting (AdaBoost). Support Vector Machines (SVM) can be used to classify the characters based on texture. This can be further improved by combining SVM with a Continuously Adaptive Mean-Shift Algorithm (CAMShift) to create a bounding box around the character values.

Deep Learning Techniques. Most of the traditional methods are replaced by computer vision algorithms in deep learning. Selmi et al. (2017) uses a Convolutional Neural Network (CNN) classifier to detect the image parts as license plate or non-license plate region. Further they coupled CNN with Non-Maximum Suppression (NMS) to locate plate regions exactly.

You-look-only-once (YOLO) is a very recent approach that is gaining popularity. A variant uses two separate CNNs or YOLO-2, one to locate the vehicle and the other to locate the license plate. Gee-Sern et al. (2017) proposed a model combining the YOLO and YOLO-2 to address the environment exhibitions faced in detecting the plate's location just by modifying a few bounding box parameters. Xie et al. (2018) have taken YOLO to another extent by adding multiple directions to it and called it Multi-Directional YOLO (MD-YOLO). This has multiple layers with filters considering other aspects like length, width, center, angles etc. to perform accurately in occlusal situations.

1.4.4 License Plate Recognition

This section focuses on the second-stage of the multistage license plate recognition system including the Pre-Processing of images, Character Segmentation and Character Recognition completely.

Pre-Processing. This step is to clear the extracted license plate image from noise, rotation etc, and prepare it for character segmentation. It mostly involves binarization of the plate to identify the characters in it. Grayscale transformation of images along with increasing the contrast levels will aid in character extraction. This can be done by setting up a threshold value for the image using Niblack's Binarization method.

Character Segmentation. Nukano et al. (2004) used Pixel Connectivity where pixels of the same weight are connected and labeled together. Another method called Projection Profiles are used to identify the start and end of a character. Generally the vertical profiles are used to locate the start and then the horizontal profiles are used to extract the character. Convolutional Neural Network can be used with bounding boxes around each character for extraction. But the CNN approach can be costly as it takes more time and requires more resources.

Character Recognition. Re-scaling needs to be done on the segmented characters, only then it can be used as inputs to classifiers to identify the character. Pattern-matching to measure the similarity of the extracted characters can be done by techniques such as Hamming distance, Mahalobian distance, Hausdorff distance, Jaccard value and Correlation.

Feature extraction techniques by calculating the eigenvectors coupled with machine learning classifiers such as SVM or Hidden Markov Models (HMM) can be used to recognise and classify the characters.

Probabilistic Neural Networks can be used to assess the probability of a character being classified accurately. Object detection techniques like YOLO can be used to match the template and the features extracted to identify but will be expensive. We need Optical Character Recognition (OCR) tools like Amazon Textract, Google Vision Cloud, Tesseract etc. to convert the images so that the data could be extracted in a form that the machine understands. These methods can be used for characters from multiple languages as well. This makes the model capable of handling universal scenarios.

1.4.5 Comparison of Existing Techniques

The existing ALPR techniques mostly are applied in multi-stage license plate recognition as it is known to produce more accurate results due to the depth of focus at each stage it can offer over single-stage approach as it can be taxing. The first stage is Detection and uses color models like RGB, HLS, HSV. The HSV is better among them as it divides the images into 13 categories that have been tested on Chinese license plates data. RGB is very basic and can be used if the images are in high resolution. Another study used Haar-like features with statistical classifiers in addition to considering features like color, shape, texture, that improved the performance of recognition and is feasible due to low load on the model. Mostly half of the traditional models use edge-based which are now considering deep learning models.

The next stage is Segmentation of the characters on the extracted number plates. Mostly binarization is used to segregate the background and characters as a pre-processing step. One powerful technique to do this is Niblack's Binarization method used 90% of the time. Pixel Connectivity just classifies based on weights but it is difficult to segregate one character from the other. Projection Profiles works best because it overcomes the drawback of Pixel Connectivity because it identifies and segregates the character based on its vertical and horizontal profiles. Another technique is using YOLO that generates bounding boxes and segregates the characters accurately. CNN approach gives 99.98% accuracy but will cost more and can weigh down the model.

The last stage is Recognition, the traditional approach uses calculation of distance between each segment and then labels a character. This method handles situations like rotated image well but cannot classify license plates having characters in different languages that have two segments as one character, thereby providing only 70% accuracy. The issue mentioned can be handled by transforming features to eigen vectors and then using classification techniques like SVM to classify characters of different languages accurately. The deep learning method uses NN and YOLO by extraction features and classifying at multiple levels due which provides high accuracy upto 99%.

Based on researchers done on object detection methods specifically for car detection YOLO seems to outperform other approaches and the reason is that it doesn't use a sliding window unlike other methods of object detection and it investigates the whole image at once. Fast R-CNN, for example, has mistakes in background identification from an image because it cannot perceive the entire context; yet, YOLO mistakes about half of the time as Fast R-CNN for background spots. Furthermore, YOLO can readily extend the model, allowing it to perform considerably better in new domains or with unusual input data.

Tesseract recognizes characters using a two-pass approach, which makes it a powerful solution for situations when the letters are difficult to read owing to weather circumstances such as rainfall or even the degree of illumination and angle of the picture or bad quality photographs.

Thus, this paper aims to use OCR methods like tesseract and YOLO for detection and recognition respectively due to its demonstrated accuracy and aim to reduce the cost factor by selecting the necessary features only without compromising on the accuracy.

1.5 Literature Survey of Existing Research

There has been a lot of research in the problem of license plate recognition and various approaches have been taken by researchers to come up with innovative ways to go about solving it. This section discusses some of the techniques used and their results.

Jamtsho et al. (2021) use the You-Only-Look-Once (YOLO) model to extract license plates of bikes whose riders do not have a helmet on in real time. The proposed system works by taking as input a frame containing the image of the rider, then it is determined whether the rider has a helmet on or not, if the rider does not have a helmet on the license plate of the bike is isolated, the characters are extracted, else the system simply moves on to the next frame and repeats the process. The input data is first annotated with a bounding box on three objects, namely, the rider, the helmet and the plate. For each of the bounding boxes, coordinates are generated and normalized between zero and one to fit the YOLO model. The model consists of 19 convolutional layers, five max pooling layers and a softmax function for object classification. The model also makes use of the centroid of the bounding boxes to track the object as it moves in the frame. The model is then trained to identify each person, the helmet , and the license plates at different epochs. The average mean precision obtained for all the three classes is 97.9% with an average training loss of 0.0829.

Silva et al.(2021) aim to overcome the drawbacks of the Automatic License Plate Recognition (ALPR) system and make use of the YOLOv2 to successfully obtain plate numbers from frontal images of vehicles where the image is considerably distorted due to an oblique view of the vehicle. The authors use the YOLOv2 as is, without making any changes to it, and treat it as a black box to detect the presence of a vehicle in the image. The images that are positively identified are then fed into a Warped Planar Object Detection Network (WPOD-NET) which searches for the license plates in the images. A WPOD-NET is a neural network that takes advantage of the fact that car license plates are rectangular planar objects. The network is then trained under different conditions of distortion to create a frontal view of the plate. The WPOD-NET consists of 21 convolutional layers, and the Rectified Linear Unit (ReLU) activation function is used throughout the network and several augmentation transforms are performed on the image, such as rectification, centering, scaling, rotation, cropping, and annotation. These images are then fed into a modified YOLO network for Optical Character Recognition (OCR). The system achieves an accuracy of about 89% which is better than the commercially used ALPR system which has an accuracy of about 81%.

Hendry and Chen (2019) perform automatic license plate detection using a sliding window darknet-You-Only-Look-Once (YOLO) method. The system architecture consists of 36 labels of which 25 are alphabets, 10 are numbers and one is a plate label. The letter ‘O’ and the number ‘0’ are treated the same on license plates in Taiwan and hence they are treated the same way for the model as well. The original YOLO model consisting of 27 convolutional neural network layers is reduced to 10 convolutional layers. A bounding box is created for each of the class labels and each image can have multiple bounding boxes. A sliding window process is used which means the window slides across each character on the plate and assigns it a single class label. The model is applied to a dataset consisting of 2049 images of Taiwanese license plates and performs well with an accuracy of 83.18% with the digits being recognized correctly 72.9% of the times and the letters 77.5%.

Pustokhina et al., (2020) presents an effective DL-based VLPR model that uses optimal K-means (OKM) clustering-based segmentation and CNN-based recognition. The model is called OKM-CNN. It has three main stages. First stage, LP localization and detection process using Improved Bernsen Algorithm (IBA) and Connected Component Analysis (CCA) model. Subsequently, OKM clustering with Krill Herd (KH) algorithm is executed to segment the LP image. Finally, the characters in LP are recognized using CNN model. The experiment was conducted using Stanford Cars, FZU Cars and HumAIn 2019 Challenge dataset. The LP image is dependent on the rule of plate and impact of light. The binary model with global threshold is not producing the expected outcome. So, the local binary technique is used and it is referred to as an image that is classified into m X n blocks. In his method two local binary methodologies were used namely Otsu and enhanced BA and that can be applied to all sub-blocks. The Ostu is a contingent on illumination constraints which has drastic variation. The enhanced BA helps to overcome the irregular illumination that has shadow images. In BA, both target and background are divided using a histogram that creates bimodal patterns.

The LP prediction is followed by LP computation, this will be the key for overall performance. CCA Model is an image processing model, it labels the pixels on the basis of connection between the pixeles and creates the group. This will create two categories black characters in white backdrop and white characters in a black backdrop. The CCA is used to recognize the white frame and black characters.

The OKM-based character segmentation process the neutrosophic analysis is applied to evaluate uncertainty of an image dataset. A membership function consisting of degree of falsity, indeterminacy and truth are applied for mapping the input images to a form that tends to produce images. CNN based recognition process is a familiar DL Model that recognizes the characters present in the segmented LPs. The CNN consists of a set of conv, pooling and Fully Connected (FC) layers. The accuracy level for this model evaluated at 98.1%.

Gnanaprakash et al., (2021) presents You Only Look Once (YOLO) is a deep learning model used to recognize objects. YOLO object detection algorithm is developed using CNN. The datasets consist of car images and number plate images. The data is then split into train and test images and then annotated to machine readable form. The YOLO Algorithm is used to train the model. First, the live CCTV footage is converted to frame and then it is passed through the YOLO algorithm to detect the cars. The images containing the cars that are separated are stored. The license plate numbers are recognized through OCR (optical Character Recognition) and stored separately. Usage of CNN increases the efficiency and as the hidden layers increase the system learns the relation between input and output layers. The YOLO works well for the detection in the case of live video but processing every frame is a tedious task. In order to handle this, it needs more GPU and complex code. This can be overcome by the ImageAI framework.

The ImageAI framework performs well in detection and recognition techniques. Python software and NVIDIA Jetson Nano are used for car and number plate recognition respectively. Images of Tamil Nadu license plates are utilized to evaluate the model's performance. The accuracy is 97% car detection, 98% for number plate localization, and 90% for character recognition.

Yousif et al. (2020) considered both the (Arabic-Egyptian) license plates and English license plates. It uses four main stages: detection, segmentation , recognition and database communication. The edge detection and morphological operations are used to detect the appropriate location of LP. The salient features in the LP are extracted from the neutrosophic set (NS) algorithm. The optimization is achieved using genetic algorithms. This aims to reduce indeterminacy in LP images. The clustering K-means algorithm is applied and connected component labeling analysis(CCLA) is applied to extract the characters. These characters are recognized by matching with the template stored in the database. Accuracy of 96.67% is achieved for a high resolution Egyptian (LP) and Accuracy of 94.27% is achieved for a low resolution-corrupted English (LP).

Zou et al., (2020) analyzed on the Chinese City Parking Dataset (CCPD). This is one of the challenging data for the character segmentation as the images are taken in dark, rotated, snowy, light, tilted or challenged scenes. To address this challenge to locate the character position it uses the model that is based on Xception, MobileNetV3, and spatial attention mechanism . Bi-directional Long Short-Term Memory (Bi-LSTM) network and 1D-Attention is used to accurately locate the character of each license plate. This method will suppress useless features and enhance useful features, and effectively process regular and irregular license plate images in actual scenes. The feature extraction network is based on a spatial attention mechanism. This can activate the regions of license plate character features to extract the LP features. To locate each character of license plates the Bi-LSTM algorithm combined with the contextual position information is adopted and visualized through heat maps. To enhance the useful character features and suppress useless character features One-Dimensional (1D) attention is added. This can effectively extract character features, and improve the accuracy of license plate recognition. By evaluating the reliability of the proposed algorithm on public datasets, the experimental results prove that the network model has a good accuracy compared with other algorithms and good generalization ability. The recognition rate ranges from 86.6 to 99.3 percent.

Henry et al., (2020) present a highly accurate deep ALPR system with three-deep stage. It combines deep learning with image processing based multinational LP layout detection algorithms. First, it uses tiny YOLOv3 network architecture to detect the LP region of the image and it is referred to as an "attention network". Second stage uses YOLOv3-SPP for unified character recognition as it deals with multiscale and small objects. This stage is called a recognition network. YOLOv3-SPP is a modified version of YOLOv3 that includes a spatial pyramid pooling (SPP) block. This step will not provide the order of the recognized characters. So a multinational LP layout detection algorithm is used. This can effectively classify single and double line LPs. The ALPR system was tested on the Korean (KarPlate dataset), Taiwanese (AOLP dataset), American (Caltech Cars (Rear) 1999), Greek (Medialab LPR Database), and Croatian (University of Zagreb) license plate datasets. The algorithm is divided into three main steps namely License Plate Detection, Unified Character Recognition and Multinational License Plate Layout Detection. The first phase uses YOLOv3, whereas the second employs

YOLOv3-SPP, a YOLOv3 version that contains the spatial pyramid pooling (SPP) block. The localized LP is sent into YOLOv3-SPP for character recognition. The predicted character bounding boxes are returned by the character recognition network. The accuracy is between 95.65 to 99.41.

Zhai et al. (2012) propose three stages for the Automatic Number Plate Recognition System (ANPRs). Number Plate Localisation (NPL), character segmentation, and Optical Character Recognition (OCR). "OCR is a widely used technology which translates scanned images of printed text into machine encoded text". ANPR needs high performance workstations and expensive computers. This can be satisfied by Field Programmable Gate Arrays (FPGAs) and Digital Signal Processors (DSPs). ANPR algorithms take specific advantage available within FPGAs, such as parallelism, reconfiguration, low power consumption, development time, and vast on-chip resources. It uses an Artificial Neural Network (ANN) based OCR algorithm and is tested using MATLAB. A database of 3700 UK binary character images is used to test the performance. In order to deal with noisy and unknown outdoor environment effects ANNs, statistical classifiers, and common pattern matching techniques are used. The OCR algorithm uses two-layer feed forward neural networks. This translates character images from NP into machine encoded text. MATLAB has been used to generate the weights of neural networks. The most suitable number of neurons were identified by testing the several neural networks with different numbers of neurons. Here two sets of training data were considered. The first set is the 45% training data and the second one is the same set with added noise, which can slightly increase the performance of neural networks. algorithm can process one character image in 8.4ms with a successful recognition rate of 97.3% with a dataset of 3700 character images.

Qadri and Asif (2009) implement ANPR, an image processing technology. The ANPR was invented in 1976 at the Police Scientific Development Branch in the UK. The ANPR system works in three stages. First, detection and capturing the vehicle image, then detection and extraction of the number plate. Finally, image segmentation to get individual character and optical character recognition. The software model consists of MATLAB 7.0.1 which is used for image processing techniques. The images that are captured using a USB camera are stored in RGB format. The hardware Model consists of sensors that sense the presence of a vehicle, capture the image through camera and a motor with motor driver circuit that control the barrier on the entrance and PC on which algorithm is executed, and microcontroller for controlling the complete hardware of the ANPR system. When the vehicle enters and settles in the field of the sensor, the infrared sensor senses a vehicle then sends a signal to the PC through microcontroller 89C51 to capture the image of the vehicle. The camera connected to the PC through the USB port captures the image of a vehicle. The ANPR algorithm receives the image and then performs the processing, which yields the vehicle number. This number is then compared to the authorized number to confirm its validity and finally provides a signal to the microcontroller to control the system hardware. If the input plate contains the authorized number then the barrier on the entrance will be raised using a motor, green indicator light will be switched on means Access granted, and if the input plate contains an unauthorized number then the barrier will not be raised, red indication will be switched on means access is denied. The system was implemented on the entrance for the security control of a highly restricted area like military zones or areas around top government offices e.g. Parliament, Supreme Court etc. The developed system detects the vehicle and then it captures the vehicle image. Vehicle number plate region is extracted using the image segmentation. Optical character recognition technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address, etc. The system is implemented and simulated using Matlab, and the performance is tested on the real images. From the experiment it is observed that the system successfully detects and recognizes the vehicle number plate on real images.

Laroca et al. (2018) propose a robust real-time ALPR system based on the YOLO object detection Convolutional Neural Networks (CNNs). The Convolutional Neural Networks (CNNs) are trained and fine tuned to be robust under different conditions like variations in camera, lighting, and background. The system uses a larger benchmark dataset, called UFPR-ALPR, focused on different real-world scenarios and the SSIG SegPlate Database (SSIG) is the largest public dataset of Brazilian LPs. The UFPR-ALPR dataset contains 4500 images that are taken from inside the vehicle in the regular traffic in an urban environment. The images were acquired with three different cameras in Portable Network Graphics (PNG) format with size of 1,920 × 1,080 pixels. The cameras that were used are GoPro Hero4 Silver, Huawei P9 Lite and iPhone 7 Plus. The dataset is split into 40 % train data, 40 % test data and remaining 20% is for validation. The approach consists of four stages namely vehicle and LP detection, character segmentation and character recognition. Two trained CNNs vehicle detection in the input image and LP detection. The character segmentation and recognition are achieved through the models Fast-YOLO, YOLOv2 and CR-NET. The accuracy in each stage has been calculated for the UFPR-ALPR dataset. The accuracy for vehicle detection is 100.00%, license plate detection is 98.33%, character segmentation is 95.97%, and Character Recognition is 90.37%.

Xie et al. (2018) propose CNN-based MD-YOLO framework for multi-directional car license plate detection. It uses an accurate rotation angle prediction method to realize multi-directional car license plate detection and fast intersection-over-union evaluation strategy between two rotational rectangles using angle deviation penalty factor (ADPF). This method can elegantly manage rotational problems in real-time scenarios. The original YOLO framework framework predicts the center coordinate, height, and width of each object. Hence this method introduces angle of rotation for a given car license plate image. Thus, before training the model, it should parameterize the original angle information. where i indicates the i-th car plate. N is the total number of car plates in the training set, and abs is the absolute value function. It enumerates each rotation angle θk of the car license plate in a training set to find maximum rotation angle. MD-YOLO network consists of 7 convolutional layers and 3 fully connected layers. The first five convolutional layers are followed by Max-Pooling (MP) with a 2 × 2 window size and a stride of 2. The final three fully connected (FC) layers have 256, 800, and 637 channels, respectively. To ensure that our CNN model can identify negative rotation angle values, leaky and identity functions are chosen as the activation functions, rather than ReLU function. We first pre-train our model using the ImageNet dataset. Then, the model is trained to detect car license plates. We select the sum-squared error as the loss function. Each grid cell will predict two bounding boxes in that some of which are neglected for low confidence. As the geometric center of the plate is located in the yellow grid cell, that cell predicts the true car license plate bounding box. The predicted box with Pf larger than the threshold is selected to be the final output. Each input image is divided into regular S × S grid cells, and the cell in which the car plate center is located is used to detect the car license plate. B bounding boxes and a confidence score P(object) are predicted for each grid cell. The confidence values help to understand how likely the bounding boxes contain car license plates. Further, each bounding box predicts six values: x, y, width, height, angle and confidence P(IoU). Here, x and y are coordinates and confidence P(IoU) represents the IoU between the bounding box and ground truth. Thus, P(object)× P(IoU) yields the final output probability Pf of each bounding box. If Pf is larger than a certain threshold, the corresponding bounding box is selected as the output. The regression targets t = {tx ,ty ,tw,th ,ta} and predicted values v = {vx , vy , vw, vh , va} are parameterized. The predicted values of each and every grid cell are stored in a sequence. First, the prepositive CNN model takes a full image as input and produces an attention region. Then, the attention region is cropped out and input to MD-YOLO which finally determines a precise rotational rectangular region. The Application Oriented License Plate (AOLP) [12] dataset contains 2049 images of Taiwanese car plates. This dataset is divided into three subsets namely Access control (AC) , traffic law enforcement (LE) and road patrol (RP). This method manages the problem of multi-directional car license plate detection very well. ALMD-YOLO performs better than the single MD-YOLO. Interestingly, the best performance of 99.5% is exhibited by the ALMD-YOLO framework.

Selmi et al. (2017) perform LP detection and recognition based on deep learning approach, it is divided into three parts namely detection,segmentation, and character recognition. LP detection is divided into pre-processing and CNN classifier steps. The image is Converted from RGB to HSV image, Morphology filtering to contrast maximization, Gaussian blur filter, Adaptive threshold,Finding all contours, Geometric filtering, CNN license plate detection and Boundary box of license plate. The character segmentation is a very important phase to facilitate the recognition process. It consists in extracting the numbers from the image of the LP. Several factors make this stage complex, such as the numbering system, the colors, the style (background), the low blur resolution, the noise and the plate rotation.The segmentation process is divided into several steps namely convert to grayscale, maximize contrast, canny edge detection, extraction hierarchy contours, geometric filtering and boundary boxes of characters are detected. The LP dataset from the public is around 2400 images and Non-LPs 4150 images. The LP resulted in an accuracy of 93.8%. The Caltech dataset has given 94.8 %. For the AOLP dataset in the three subsets they gave 96.2 % in the AC, 95.4% in the LE and 95.1% in the RP.

Benjdira et al. (2019) conduct a comparison of two approaches, YOLO-v3 and Faster R-CNN(Region-based Convolutional Neural Network), was carried out specifically for automobile identification. The findings demonstrate that YOLO-v3 outperforms the other approach. In this study, five metrics were used for this comparison: Specificity, Sensitivity, F1-score, total Accuracy, and time to detect. Specificity is 99.66% for Faster R-CNN vs 99.73% for YOLO-v3 which is very close for both methods. Sensitivity for Faster R-CNN is 79.40% for Faster R-CNN vs 99.07% for YOLOv3 which shows the significance between these two approaches. Each image in YOLOv3 takes an average of 0.057ms to process. For faster R-CNN, the average processing time for each picture is 1.39 seconds. This demonstrates the huge difference between the two algorithms and demonstrates that YOLO-v3 surpasses the quicker R-CNN in terms of processing time.

Ammour et al.(2017) proposed an approach in this study, which comprises four phases. First, employing the Mean-shift method for the input picture, which is over-segmented into a series of images. Second, potential pictures are framed and sent as input to a pre-trained CNN for feature extraction in this stage. In the third step, the SVM algorithm is trained to determine whether an input frame is an automobile or not. The binary maps generated by SVM are fine-tuned in the last step for additional inspection of isolated autos. This approach was tested on various picture sizes ranging from 2626x4680px to 3456x5184px with accuracy ranging from 73.3 percent to 90 percent and window sizes ranging from 60x60 to 200x200 with results ranging from 75 percent to 93.6 percent with 160x160px being picked as the best window.

Vaiyapuri et al. (2021) perform the Vehicle License Plate Recognition (VLPR) using different methods like colors, shapes, pattern recognition and non-uniform illumination at the time of capturing the images. The process involves the HT-SSA-CNN model. The input images are preprocessed for further processing. The preprocessed image is fed into the LP localization process to detect and crop the LP effectively. HT has been applied to segment the characters that exist in the LP. At the end SSA-CNN model is applied to examine and check the characters in the classified image. In the preprocessing stage the RGB car image undergoes downscaling to 50% of its actual scale in order to confine the processing duration. The reduction and a reforming of images is applied to ensure the minimization of candidate sites. The input image consists of RGB channels and each channel is restricted to within (0–255), the gray scale image has a single channel and the RGB image is converted into a grayscale template. The contrast of the images has been enhanced and LP is detected. In License Plate Localization the LP is filtered using a set of tasks, like using Median Filter (MF) with (3×3) for image development and noise elimination, exploiting sobel edge detector to detect proper edges and employment of morphological tasks to isolate the plate from back-ground. Erosion is applied to allocate the candidate plate regions under the application of the Squared Structuring component. Finally, the appropriate LP is placed. In this domain, it is applied with two fundamental checkers to ensure the plate region accurately, removing the unwanted sites. Then a rectangle shape checker is executed to check the presence of rectangular-sized objects in the image. Dimension of Plate Checker verify if height/width of the succeeding region. This green channel offers sufficient image contrast, blurs the image for LP edge smoothening and discards the artifacts. Character Segmentation Using Hough Transform is mainly employed to examine the lines in the images. The pixels in image space (x0, y0) can be represented by applying the transformation, $r = x. \cos\theta + y \cdot \sin \theta$ (1). A curve $r = x0. \cos \theta + y0 \cdot \sin\theta$ is touted to have attained the parameter space (θ, r). The HT-SSA-CNN approach achieved higher precision of 98.1%, 97.9% and 96.1% on Stanford Cars, FZU Cars and HumAIn 2019 datasets respectively.

Shashirangana et al.(2021) conduct a survey and compare different techniques and methods for performing automated license plate recognition. Under the automatic plate recognition approaches, the first approach they evaluated is the multi-stage license plate recognition system which consists of three main steps. The first step is to perform object detection and isolate the license plate in the image. The second stage involves segmentation being performed and characters being extracted using morphology. In the final stage, character recognition is performed using various pattern matching techniques. The second approach discussed is the single stage license plate recognition system and a majority of the attempts implement a single deep neural network that performs end-to-end recognition. This method allows the use of fewer features as compared to the multi stage approach which makes for faster training of the model.

Next, the paper talks about license plate detection and based on the features of the plate compares different methods for its detection. The first approach is the edge based method and makes use of the edge of the plate which sets it apart from the body of the vehicle. The edges in the image would be the horizontal and the vertical edges of the plate. This technique uses the Sobel filter which has two 3 x 3 matrices dedicated to the detection of the two edges. The drawback of using this filter is that it responds to noise. The edge based method has a success rate of 99.45%. Another transformation is the Hough Transformation which identifies the straight lines in the image upto an angle of 30° with an accuracy of 98.8%. The major drawback of this transformation is that it consumes a lot of memory and time and is very sensitive to deformations in the boundary. The color based method makes use of the fact that the color of the plate is different from the color of the vehicle and that the combination of the characters and the plate is unique to the plate and is not seen anywhere else in the image. It uses the HLS and the RGB color models to classify the pixels but the HSL model is sensitive to noise. The texture based method uses the presence of characters in the image and creates a transition of color on the plate. It creates a distribution of pixel intensity that results in the plate having a high edge density. The texture based method performs well with an accuracy of 96.5%. Character based methods detect the presence of characters in the license plates and only consider the area where characters are observed. Many of the implementations of this approach make use of neural networks to identify possible regions of the plate and different levels of accuracy are observed ranging from 95% to 99%. Statistical classifiers make use of statistics with Haar-like features along with Adaptive Boosting to train algorithms for enhanced detection of license plates which reach accuracy levels of 94.5%. Deep learning techniques such as neural networks have very high success rates in detecting license plates in images and include approaches such as convolutional neural networks, You-Only-Look-Once (YOLO) , and sometimes both of them together to reach accuracies of upto 93.8%.

After the presence of the license plates has been confirmed in an image, the next step is to identify the characters in the plate. In order to do this several pre-processing steps must be performed. There are several challenges that must be overcome such as the rotation which is handled using bilinear transformation. Binarization and adaptive binarization are performed to separate pixels that are a part of the characters. Character segmentation involves segmenting the characters using various methods. It can be performed using pixel connectivity where based on whether the pixels are connected or not, their size and aspect ratios, the pixels are determined to form part of a character. Segmentation is also performed using projection profiles which observe the color of the background and the characters after binarization. This technique, however, is sensitive to noise and depends a lot on the quality of the image. Segmentation can also be done using prior knowledge of the plate such as its aspect ratio, color of the pixels, etc. Deep learning is another technique used for segmentation where convoluted neural networks however they may take a lot of time to execute and consume a lot of resources.

After the characters are segmented, they must be recognized to get the plate numbers from the image. The first method to do this is template and pattern matching where the fact that plates use a specific font and size template matching is performed for each segment of the image. Feature extraction is another technique to perform character recognition. Feature extractors are used which handle rotations and noise in the image well. Deep learning is the final technique discussed for recognition where probabilistic neural networks are used though they tend to increase computational costs and are less accurate than template matching.

Constraint based license plate recognition models make use of multi stage recognition techniques for plate localization and character recognition. The input to the model is an RGB image, color edge detection and HSI transformation are performed on the image. Next, character categorization and topological sorting are performed with input from the character segmentation system which binarizes and removes noise from the image. This model achieves different accuracies ranging from 92% to 98% based on the dataset used by the authors.

A detailed review and analysis of the literature reveals that the deep learning techniques are preferred over other approaches. The use of neural networks exponentially increases the accuracy of the model and gives much better results than template and formatting approaches. Many of the implementations involve the use of the YOLO model, it has either been scaled down or used as is to perform tasks such as license plate localization and character recognition. Both these approaches have also been used together where a neural network has been used to localize the plates and YOLO has been used to segment and recognize the plate numbers. The accuracies range from 93% to about 98% compared to the other approaches which have an accuracy of 85-89%.

References

Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sensing*, 9(4), 312. https://doi.org/10.3390/rs9040312

Automatic *number-plate recognition*. (2022, February 22). Wikipedia. https://en.wikipedia.org/wiki/Automatic_number-plate_recognition#:~:text=ANPR%20was%20invented%20in%201976

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. 2019 *1st International Conference on Unmanned Vehicle Systems-Oman (UVS).* https://doi.org/10.1109/uvs.2019.8658300

California ALPR FAQs How do Automated License Plate Recognition (ALPR) Systems work? (n.d.). Retrieved February 17, 2022, from https://ncric.ca.gov/html/California%20Law%20Enforcement%20ALPR%20FAQ_.pdf

California, S. of. (n.d.). Automated License Plate Reader Privacy and Usage Policy. Bureau of Automotive Repair. Retrieved February 17, 2022, from https://www.bar.ca.gov/ALPR#:~:text=The%20ALPR%20system%20captures%20other

Detecting License Plate and Identifying the Registration Details using OpenCV and Python. (n.d.). www.youtube.com. Retrieved February 17, 2022, from https://www.youtube.com/watch?v=_UCqxuCzNE8

Duan, T., Tran, D., Tran, P., and Hoang, N. (2010). Building an automatic vehicle license-plate recognition system. *3rd International Conference on Computer Science and Information*

Technology. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.8984&rep=rep1&type=pdf

Gnanaprakash, V., Kanthimathi, N., & Saranya, N. (2021). Automatic number plate recognition using deep learning. *IOP Conference Series: Materials Science and Engineering*, *1084*(1), 012027. https://doi.org/10.1088/1757-899x/1084/1/012027

Hendry, & Chen, R. C. (2019). Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. *Image and Vision Computing*, *87*, 47–56. https://doi.org/10.1016/j.imavis.2019.04.007

Henry, C., Ahn, S. Y., & Lee, S.-W. (2020). Multinational License Plate Recognition Using Generalized Character Sequence Detection. *IEEE Access*, *8*, 35185–35199. https://doi.org/10.1109/access.2020.2974973

Hsieh, C.-T., Juan, Y.-S. and Hung, K.-M., (2005) Multiple license plate detection for complex background. *19th International Conference on Advanced Information Networking and Applications (AINA'05) Volume 1 (AINA papers).* https://doi.org/10.1109/AINA.2005.257

Hsu,-S., Ambikapathi, A., Chung, S.-L. and Su, C.-P. (2017). Robust license plate detection in the wild. *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* https://doi.org/10.1109/AVSS.2017.8078493

Introduction to License Plate Recognition Software. (n.d.). www.licenseplatesrecognition.com. Retrieved March 6, 2022, from https://www.licenseplatesrecognition.com/

Jamtsho, Y., Riyamongkol, P., & Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclists using YOLO. *ICT Express*, *7*(1), 104–109. https://doi.org/10.1016/j.icte.2020.07.008

Jelal. (2021). *License Plate Annotated Image Dataset* (Version 1) [Dataset]. Kaggle. https://www.kaggle.com/aladdinss/license-plate-annotated-image-dataset

Khazri, A. K. (2020). *Labeled license plates dataset* (Version 1) [Data set]. https://www.kaggle.com/achrafkhazri/labeled-licence-plates-dataset

Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D Object Representations for Fine-Grained Categorization. (Version V1)[Data set].IEEE. https://doi: 10.1109/ICCVW.2013.77

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Goncalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. *2018 International Joint Conference on Neural Networks (IJCNN)*. https://doi.org/10.1109/ijcnn.2018.8489629

Larxel. (2020, June 1). *Car License Plate Detection* (Version 1) [Data set]. Kaggle. https://www.kaggle.com/andrewmvd/car-plate-detection

Matas, J. and Zimmermann, K., (2005). Unconstrained licence plate and text localization and recognition. *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* https://doi.org/10.1109/ITSC.2005.1520111

Nukano, T., Fukumi, M. and Khalid, M. (2004). Vehicle license plate character recognition by neural networks. *Proceedings of 2004 International Symposium on Intelligent Signal*

Processing and Communication Systems, 2004. ISPACS 2004. https://doi.org/10.1109/ISPACS.2004.1439164

Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J. P. C., Gupta, D., Khanna, A., Shankar, K., Seo, C., & Joshi, G. P. (2020). Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems. *IEEE Access*, *8*, 92907–92917. https://doi.org/10.1109/access.2020.2993008

Qadri, M. T., & Asif, M. (2009). Automatic Number Plate Recognition System for vehicle identification using optical character recognition. *2009 International Conference on Education Technology and Computer*. https://doi.org/10.1109/icetc.2009.54

Redmon, J. (2012). YOLO: Real-Time Object Detection. Pjreddie.com. https://pjreddie.com/darknet/yolo/

Selmi, Z., Ben Halima, M., & Alimi, A. M. (2017). Deep Learning System for Automatic CanLicense Plate Detection And Recognition. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. https://doi.org/10.1109/icdar.2017.187

Selvan, T. S. (2018, November 1). *Indian License Plates* [Data set]. Kaggle. https://www.kaggle.com/thamizhsterio/indian-license-plates

Shashirangana, J., Padmasiri, H., Meedeniya, D., & Perera, C. (2021). Automated License Plate Recognition: A Survey on Methods and Techniques. *IEEE Access*, 9, 11203–11225. https://doi.org/10.1109/access.2020.3047929

Silva, S. M., & Jung, C. R. (2021). A Flexible Approach for Automatic License Plate Recognition in Unconstrained Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://doi.org/10.1109/tits.2021.3055946

Tafazzoli, F., Nishiyama, K,. and Frigui, H,. (2017). A Large and Diverse Dataset for Improved Vehicle Make and Model Recognition [Data set]. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).* https://ieeexplore.ieee.org/document/8014855

Vaiyapuri, T., Nandan Mohanty, S., Sivaram, M., V. Pustokhina, I., A. Pustokhin, D., & Shankar, K. (2021). Automatic Vehicle License Plate Recognition Using Optimal Deep Learning Model. *Computers, Materials & Continua*, *67*(2), 1881–1897. https://doi.org/10.32604/cmc.2021.014924

W. (2020). *Vienamese Plate Dataset* [Data set]. Github. https://github.com/winter2897/Real-time-Auto-License-Plate-Recognition-with-Jetson-Nano/blob/main/doc/dataset.md

Wang, F., Man, L., Wang, B., Xiao, Y., Pan, W. and Lu, X. (2008). Fuzzy-based algorithm for color recognition of license plates. *Pattern Recognition Letters, 29.* https://doi.org/10.1016/j.patrec.2008.01.026

Wang Weihong and Tu Jiaoyang (2020). Department of Computing, Zhejiang University of Technology, Hangzhou 310000, China. *IEEE Access*, *8*, 91661 - 91675. https://ieeexplore.ieee.org/document/9092977

What Is Optical Character Recognition (OCR)? (n.d.). Www.ibm.com. https://www.ibm.com/cloud/blog/optical-character-recognition

Xie, L., Ahmad, T., Jin, L., Liu, Y., & Zhang, S. (2018). A new CNN-based method for multi-directional car license plate detection. *IEEE Transactions on Intelligent Transportation Systems*, *19*(2), 507–517. https://doi.org/10.1109/tits.2017.2784093

Yousif, B. B., Ata, M. M., Fawzy, N., & Obaya, M. (2020). Toward an Optimized Neutrosophic k-Means With Genetic Algorithm for Automatic Vehicle License Plate Recognition (ONKM-AVLPR). *IEEE Access*, *8*, 49285–49312. https://doi.org/10.1109/access.2020.2979185

Zhai, X., Bensaali, F., & Sotudeh, R. (2012). OCR-based neural network for ANPR. *2012 IEEE International Conference on Imaging Systems and Techniques Proceedings*. https://doi.org/10.1109/ist.2012.6295581

Zou, Y., Zhang, Y., Yan, J., Jiang, X., Huang, T., Fan, H., & Cui, Z. (2020). A Robust License Plate Recognition Model Based on Bi-LSTM. *IEEE Access*, *8*, 211630–211641. https://doi.org/10.1109/access.2020.3040238

Zunino, R & Rovetta, S. (2000). Vector quantization for license-plate location and image coding *IEEE Trans. Ind. Electron.* https://doi.org/10.1109/41.824138