Automatic Number Plate Recognition (ANPR)

Group-28

1)Kasthuri Vyshnavi
2)Sujit Shibaprasad Maity
3)Aditi Das
4)Tania Yadav

Contents

1. INTRODUCTION

With the growing number of vehicles, finding a car park is a serious issue today for a large number of students and faculty at Educational Institutions. Most of the car parks are managed manually by security guards who do not keep a track of the number of vehicles entering and exiting the premises. Hence, the vehicle driver have to keep circling the car park in order to find a vacant slot leading to a wastage of time, not to mention the anxiety and frustration of the driver. The absence of the security guards may also lead to vehicle thefts.Over the last few years, the ANPR has become a useful approach for vehicle surveillance.**Automatic Number Plate Recognition (ANPR), also known as License Plate Recognition (LPR), is a technology that uses optical character recognition (OCR) and machine vision to automatically read vehicle license plates. The primary purpose of ANPR systems is to efficiently and accurately capture and interpret license plate information from vehicles in real-time**

AI and deep learning are being used everywhere, from voice assistants to self-driving cars.One such application is the Automatic Number Plate Recognition (ANPR) of Vehicles.ANPR is a technology that uses the power of AI and deep learning to automatically detect and recognize the characters of a vehicle’s license plate.With the increase in the number of vehicles, vehicle tracking has become an important research area for efficient traffic control, surveillance, and finding stolen cars. The specific use cases may be traffic violation control, parking management, toll booth payments, etc. For this purpose, efficient real-time license plate detection and recognition are of great mportance.

2.PROBLEM STATEMENT

Build a CV model for recognizing the Number Plate and Displaying the Number.

Task 1: Data collection and creating a bounding box for the number plate region in the additional images.
Task 2: ANPR Automatic Number Plate Recognition (ANPR) implementation involves following three steps:

Step 1: Detect and localize a license plate in an input image/frame
Step 2: Extract the characters from the license plate
Step 3: Apply some form of Optical Character Recognition (OCR) to recognize the extracted character.

3.LITERATURE REVIEW

ANPR system consists of three main stages:

1) Number Plate Localization (NPL),

2) Character Segmentation (CS), and

3) Optical Character Recognition (OCR).

The NPL stage is where the Number Plate is being detected. The Character Segmentation stage is an important pre-processing step before applying OCR, where each character from the detected Number Plate is segmented before recognition. In the last stage, characters are segmented from the Number Plate so that only useful information is retained for recognition where the image ormat will be converted into characters .Various research journals were consulted to find relevant information regarding ANPR based applications.ANPR systems are based on common approaches like Artificial Neural Network (ANN) ,Probabilistic neural network (PNN),Optical Character Recognition (OCR) ,MATLAB Configurable method, Sliding Concentrating window (SCW), Back-Propagation Neural Network , Support Vector Machine (SVM), Inductive Learning , Region-based, Color Segmentation , and Fuzzy-Based Algorithm , Scale Invariant Feature Transform (SIFT) .

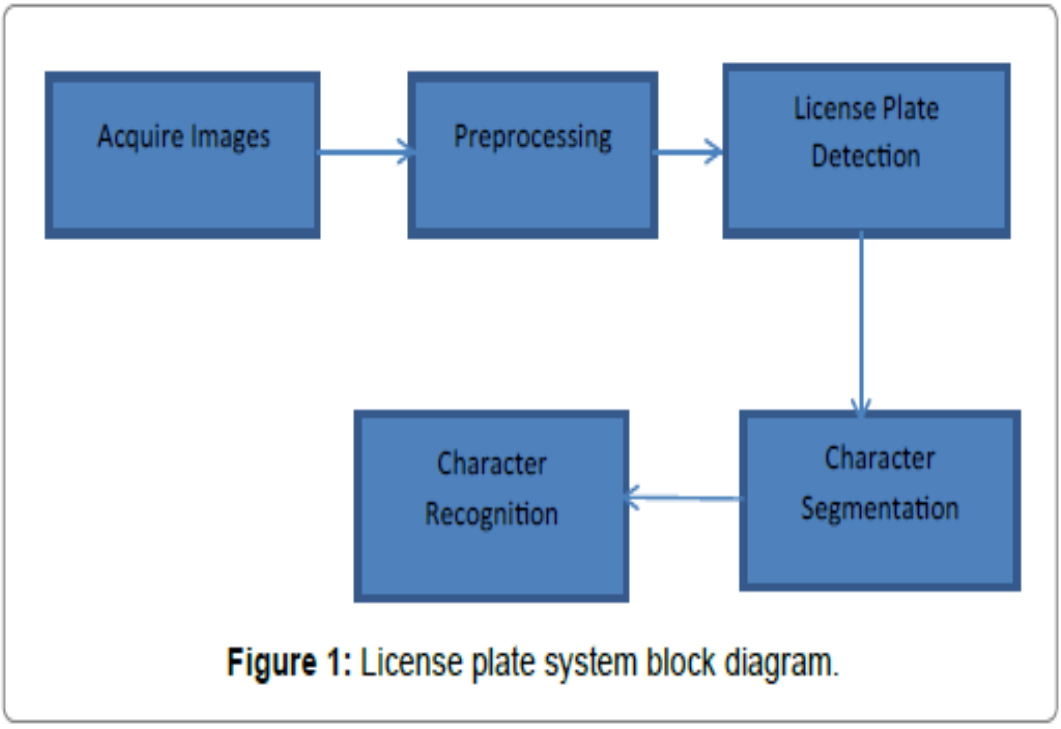

Figure 1: License plate system block diagram.

Object detection is one of the classic problems in computer vision that should not be confused with classification, as it is much more complex. The latter can recognize objects but does not indicate the area in which they are located. In addition, classification cannot work on frames that contain multiple objects.

YOLO ("You only look once") is part of the category of systems that allow the detection of several objects within the same frame. It is popular because, in addition to achieving high accuracy, it can also run-in real time. The algorithm is named so because it requires a single forward propagation through the neural network in order to produce results. After non-max suppression, the algorithm exposes the objects present in the frame, surrounded by delimitation boxes, ensuring that they are not detected several times. Models for object detection can be divided into two categories (1) Two-Stage Detectors, (2) One-Stage Detectors. YOLO, as expected, is part of the category of one-stage detection models (also called one-shot detection models), which do not need the last preliminary step, present in the case of two-stage detectors. This last phase, within these detectors, is a costly one from a computational point of view, slowing down the detection process. This happens because it involves the detection of important regions and then, based on them, the detection of the objects sought. Thus, after receiving the input, there are 3 phases for fulfilling the purpose of the model (1) Backbone, (2) Neck, (3) Dense Prediction (Head).

Backbone is a neural network made up mostly of convolutional layers. Its purpose is to extract the essential features from the input data, having a particularly important role in the final performance of the model.

Neck - The next step to achieve the proposed goal is to mix and combine the features extracted in the backbone in the "Neck" block to prepare them for the detection step. The authors of YOLOv4 propose in this block a series of tools already used in the world of deep learning algorithms, but modified to better fit the needs of the model and to increase its final accuracy. After CSPDarknet53, an SPP (Spatial Pyramid Pooling) block is added in order to increase the receptive field and separate out the most important features extracted by the backbone. Original SPP arose from the need to have an approach that can have multi-scale input images for training because, in the past, only fixed-size images could be used due to the presence of the fully connected layers. The problem with these types of layers started from the fact that for them we have to specify which the input layers and the output layers are. Before these layers, there is always a series of convolutional or max pool layers that need to be flattened in order to be used as input. This last step represents the bottleneck in being able to receive inputs of

variable size, since the input of a fully connected layer must remain of fixed length. The first deep learning models used simple networks that manipulate the input data through a succession of layers, each layer having as input data the output of the previous one. Due to the evolution of models of this type, it has been observed that as we progress towards the end of the network, some information that may help fine-tune the prediction may be lost. In response to this problem, the authors of the YOLOv4 model used PaNet, whose architecture allows better information propagation. The components of the neck typically flow up and down among layers and connect only the few layers at the end of the convolutional network. One aspect to note is that in the original PaNet implementation, low-level information is brought to the top layers by adding together the current layer and information from the previous layer to form a new vector. Instead, the authors of YOLOv4, propose a modified variant that uses a new vector created by concatenating the input and the vector from the previous layer.

Head - the last part of the model deals with the following: • Locating bounding boxes; • Classifying what is inside each box. Due to the architecture present in the neck, the head's input will contain spatially rich information from the bottom-up stream and the semantic rich information from the top-down stream. The output of the algorithm is composed of 4 values that are meant to describe the predicted bounding box (x, y, h, w) and the probability of k classes and an additional one for the background. An anchor box is predefined and has an aspect ratio set, which is defined beforehand by running, on the dataset, a k-means clustering.

4.METHODLOGY

Task 1: Data Collection and Bounding Box Creation

Data Collection:

Collect a diverse set of images containing vehicles with visible license plates. Ensure that the dataset covers different lighting conditions, angles, and vehicle types.

Annotate the images with ground truth bounding boxes around the license plate regions.

Data Preprocessing:

Resize and normalize the images to ensure consistency.

Convert annotations into a format compatible with your chosen deep learning framework (e.g., YOLO format, COCO format).

Model Selection:

Choose a pre-trained object detection model suitable for license plate detection. Popular choices include YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), or Faster R-CNN (Region-based Convolutional Neural Network).

Fine-tuning:

Fine-tune the selected model on your license plate dataset. This step helps the model adapt to the specific characteristics of license plates in your dataset.

Training:

Train the model on the annotated dataset, adjusting hyperparameters as needed. Monitor the training process for convergence and performance.

Evaluation:

Evaluate the trained model on a separate validation dataset to ensure it generalizes well to new, unseen data.

Adjust the model or training parameters if needed.

Task 2: ANPR Implementation

Step 1: Detect and Localize License Plate

Preprocess Input Image/Frame:

Resize and normalize the input image/frame.

License Plate Detection:

Use the trained object detection model to identify and localize the license plate in the preprocessed image/frame.

Extract the bounding box coordinates for the detected license plate.

Step 2: Extract Characters from License Plate

Crop License Plate Region:

Use the bounding box coordinates to crop the license plate region from the preprocessed image.

Character Extraction:

Use image processing techniques or a deep learning model to segment and extract individual characters from the license plate.

Step 3: Apply OCR for Character Recognition

OCR Setup:

Choose an OCR library or tool suitable for your programming environment (e.g., Tesseract OCR, EasyOCR).

Character Recognition:

Apply the OCR tool to recognize the characters extracted from the license plate.

Post-processing:

Post-process the OCR results to improve accuracy, such as removing unwanted characters or correcting common errors.

Output:

Obtain the final recognized license plate number as the output of the ANPR system.

5.APPLICATIONS

1.Industrial estates
2.Site entrances
3.Leisure, hotels and commercial
4.Shopping centres
5.Golf clubs
6.Petrol station drive-offs
7.Hospitals
8.Waste management sites
9.Clubs and pubs
10.Logistics and vehicle tracking
11.Car Park management and access control

6.REFERENCE

- https://www.researchgate.net/publication/356717707_An_Automatic_Car_License_Plate_Recognition_System
- https://www.researchgate.net/publication/356717707_An_Automatic_Car_License_Plate_Recognition_System
- https://d-nb.info/1217243585/34
- https://www.researchgate.net/publication/290787583_Vehicle_Number_Plate_Recognition_System_A_Literature_Review_and_Implementation_using_Template_Matching
- https://www.scitechnol.com/peer-review/a-review-paper-on-licenseplate-recognition-system-TSp0.php?article_id=11548
- https://ictactjournals.in/paper/IJIVP_Vol_9_Iss_2_Paper_2_1867_1871.pdf
- https://www.itm-conferences.org/articles/itmconf/pdf/2022/04/itmconf_icacc2022_03044.