All Weather Vehicle Recognition System (AWVRS)

Raghvendra Kumar (231010057)
raghk23@iitk.ac.in

Rupesh Kumar Meena (210881)
rupesh21@iitk.ac.in

Prof. Koteswar Rao Jerripothula Dept. of EE, IIT Kanpur
kotesrj@iitk.ac.in

Abstract

The License Plate detection and recognition (LPDR) is a challenging task that plays a significant role in intelligent transportation systems (ITS).It could be used as a core in various applications, such as security, traffic control, and electronic payment systems (e.g. freeway toll payment, traffic rules voilation charge and parking fee payment). However, the recognition of vehicle license plates are extremely difficult in dense fog-haze environments because the fog and haze blur the boundaries and characters of license plates significantly, which makes the license plates hard to be detected or recognised. To Solve this ALPR in foggy condition, Multi Attention Dehazing model based on Deep Neural Network is proposed to handle the problem of Dehazing and vehicle license plate location and recognition in dense fog. Firstly Input Image is passed through a preprocessing model which is composed of Channel Attention, Multi-scale Attention and Swin Transformer blocks. Then processed image is passed to the YOLOv8 for vehicle detection and ROI then after based on the detected bounding box of the vehicle Licence Plate Detection and Character Recognition is performed using two trained model in the pipeline ahead.The character recognition module is composed of swin transformer and the attention mechanism.

1. Introduction

Traffic problems are increasing rapidly due to rapid urban development and increasing car ownership. Traffic congestion, traffic violations, stealing cars, and fugitive criminals impose big challenges on traffic management systems. To manage the numerous vehicles conveniently, the Intelligent Traffic System (ITS) is needed to provide innovative services relating to the transport and the traffic management. Several systems are developed to solve these problems such as self-driving systems, Traffic surveillance systems , Tracking vehicle systems and Vehicle speed detection systems. License plate detection and character recognition (LPDR) is one of the most important topics in intelligent infrastructure systems, like electronic payment systems (for tolls, parking, and public transportation), and rule and regulation imposition. The LPDR system is an image processing technology used to identify vehicles using their license plate to help regulate the traffic, for monitoring purposes and fines imposing. Usually, a license plate detection and vehicle recognition (LPDR) system has mainly three phases. The first phase is image pre-processing, once the image is captured further processing of the image is carried out like converting the image from a color space to another, resizing the image resolution, and removing noises, removing blur for improving the image quality. The second phase is license plate localization, the region of interest is detected based on some license plate characteristics and image features. The final phase is the optical character recognition, this phase is considered the most crucial step because it helps to read the plate number and identify the vehicle. But Due to the existence of smoke, dust, fumes, mist and other floating particles in the atmosphere, images taken in such atmosphere are often subject to color distortion, blurring, low contrast and other visible quality degradation, and the hazy image input will which become progressively more serious as the fog and haze blur the boundaries and characters of license plates significantly, which makes the license plates hard to be detected and recognised. Most of the existing license plate detection methods are based on the feature extraction technique, and they cannot detect the license plates accurately in fog-haze environments, due to the fact that the fog-haze interference and image ambiguity make the features very difficult to be extracted. Likewise, the character recognition is also affected by the haze interference significantly. In view of this, image dehazing aims to recover the clean image from the corrupted input, this will be the preprocessing step of the high-level vision tasks. The atmosphere scattering model (Cartney 1976)(Narasimhan and Nayar 2000)(Narasimhan and Nayar 2002) provides a simple approximation of the haze effect, it is formulated as: I(z) = J(z)t(z) + A(1 t(z)) Where I(z) is the observed hazy image, A is the global atmosphere light, and t(z) is the medium transmission map, J(z) is the haze-free image. The atmosphere scattering model shows that image dehazing is an underdetermined problem without the knowledge of A and t(z). Moreover, with and d(z) being the atmosphere scattering parameter and the scene depth, respectively we have t(z) = exponential(-beta * d(z)).

Based on the atmosphere scattering model, early dehazing methods did a series of works(Berman, Avidan, and others 2016)(Fattal 2014)(He, Sun, and Tang 2010)(Jiang et al. 2017)(Ju, Gu, and Zhang 2017)(Meng et al. 2013)(Zhu, Mai, and Shao 2015). DCP is one of the outstanding prior-based methods, they propose the dark channel prior based on the assumption that image patches of outdoor haze-free images often have low-intensity values in at least one channel. However, the prior-based methods may lead to an inaccurate estimation of transmission map because of the prior may be easily violated in practice, so the prior-based methods may not work well in dense fog. Many neural network approaches have also been proposed to estimate the haze effect, including the pioneering work of DehazeNet(Cai et al. 2016), the multi-scale CNN(MSCNN)(Ren et al. 2016), the residual learning technique(He et al. 2016), the quadtree CNN(Kim, Ha, and Kwon 2018), and the densely connected pyramid dehazing network(Zhang and Patel 2018). Compared to traditional methods, deep learning methods try to directly regress the intermediate transmission map or the final haze-free image. With the big data being applied, they achieve superior performance with robustness.

This paper proposes Multi-Attention Dehazing (MAD) for addressing haze issue and All Weather Vehicle Recognition System (AWVRS). This Pipeline provides a framework which includes the image-dehazing technique based on Swin Transformer, Channel Attention and Multi-Scale Attention. YOLOv8 for ROI Detection that detect the vehicle and the number plate, then after OCR for Character Recognition once the number plate region is detected, passed to an Optical Character Recognition (OCR) model to read characters then lastly also Vehicle Classification using YOLOv8 Labels that leverage YOLOv8's detections to categorize vehicle types based on the predicted labels.

2. Related Works

2.1. Image Dehazing

Previously, most of the existing image dehazing methods depend on the formulation of physical scattering model equation1, which is a highly ill-posed problem because of the unknown transmission map and global atmospheric light. These methods can be roughly divided into two classes: traditional prior-based methods and modern learning-based methods.

DCP(He, Sun, and Tang 2010) proposed a dark channel prior for the estimation of the transmission map. However, the priors are found to be unreliable when the scene objects are similar to the atmospheric light. (Zhu, Mai, and Shao 2015) propose a simple but powerful color attenuation prior by creating a linear model for modeling the scene depth of the hazy image. (Fattal 2008) present a new method for estimating the optical transmission in hazy scenes, the scattered light is eliminated to increase scene visibility and recover haze-free scene contrasts., (Berman, Avidan, and others 2016) proposed a non-local prior to characterize the clean image, the algorithm relies on the assumption that colors of a haze-free image are well approximated by a few hundred distinct colors, which forms tight clusters in RGB space. Although these methods have made a series of success, the prior is not robust to handle all the cases, such as the unconstraint environment in the wild.

In view of the prevailing success of deep learning in image processing tasks and the availability of large image datasets, (Cai et al. 2016) proposed an end-to-end dehazing model based on convolution neural network DehazeNet, it takes a hazy image as input, and outputs its medium transmission map, which is subsequently used to recover a hazefree image via atmospheric scattering model. (Ren et al. 2016) employed a Multi-Scale MSCNN that is able to perform a refined transmission map from the hazy image. (Yang and Sun 2018) combines the advantages of traditional prior based dehazing methods and deep learning methods by incorporating haze-related prior learning into deep network. The Feature Fusion Attention Network (FFANet) is another model which uses CNN. The Feature Attention (FA) module considers the haze's uneven distribution across several image pixels.

Another model based on transformer Vision Transformers for Single Image Dehazing (Yuda et al. 2022) DehazeFormer, which consists of various improvements, such as the modified normalization layer, activation function, and spatial information aggregation scheme. Small model outperforms FFA-Net with only 25% Param and 5% computational cost. Larger one dramatically outperforming the previous state-of-the-art methods.

2.2. Object Detection and ROI Detection

CNN-based model has exhibited a great progress in terms of the object detection accuracy. In recent years, some light-weight deep learning models for object detection yield some prefer- able results. Most of these models are constructed based on SSD , SqueezeNet , AlexNet or GoogleNet . In these models, the pipelines of object detections comprise many components, such as pre-processing, large number of convolu- tion layers and post-processing. The sliding window approach or region proposal method can be taken to evaluate the clas- sifiers. These complex pipelines are computationally intensive and consequently slow. To improve this shortcoming, in [16], the You Only Look Once (YOLO) is proposed, and it is framed as a single regression problem. The outstanding advantage of YOLO is to encode the contextual information, and it makes less mistakes in term of confusing the background patches of images. The first vision transformer(ViT) successfully used transformer for the vision task and out-perperforming many state of the art CNN based object classification.ViT pioneered the direct application of the Transformer architecture , which projects images into token sequences via patch-wise linear embedding. The shortcomings of the original ViT are its weak inductive bias and the quadratic computational cost. To this end, PVT uses the pyramid architecture to introduce multi-scale inductive bias and downsamples the key and value to reduce the computational cost. T2T-ViT uses the unfolding operation just like CNNs for tokenization, and it uses the Performer to lower the computational cost.Swin Transformer [30] partitions tokens into windows and performs self-attention within a window to keep the linear computational cost. It employs the cyclic shift scheme to bridge windows so that adjacent blocks adopt different window partitions.

2.3. Character Recognition

Character recognition has been extensively studied due to its wide-ranging applications in fields like document analysis, license plate recognition, and scene text detection. Traditional optical character recognition (OCR) methods primarily relied on handcrafted feature extraction techniques, including edge detection and histogram of oriented gradients (HOG), followed by machine learning classifiers such as support vector machines (SVMs) or k-nearest neighbors (k-NN) (Jain Zhong, 1996). Shi et al. (2016) introduced Convolutional Recurrent Neural Networks (CRNNs), which combined CNNs with recurrent layers to handle sequential dependencies in character recognition tasks. Character segmentation is the foundation of the character recognition. Reference [22] presents a character segmentation algorithm, which makes use of priori knowledge and realises the character segmentation based on connected domains. Besides the licence plates, character segmentation is applied in the hand- writing text segmentation. Jun et al. propose a new hand- writing character segmentation method based on the non-linear clusterings. The whole text line is first divided into some strokes, and the similarity matrix is calculated. Then, the cluster labels of the strokes are obtained by a non-linear clustering method. Based on the cluster labels, the strokes are combined to form the characters. Roy et al. present an end-to-end real- time text localisation and recognition method to handle the dig- ital documents in real scenes. [23] proposes a segmentation-free license plate recognition algorithm that utilises deep learning techniques in the character detection and recognition process. It improves the recognition results of low-resolution images. Automatic License Plate detection and Recognition (ALPR) is a quite popular and active research topic in the field of com- puter vision. Reference [24] presents an automatic framework for License Plate (LP) detection and recognition from complex scenes, which is based on mask region convolutional neural networks used for LP detection, segmentation and recognition. An end-to-end ALPR method based on a hierarchical Convolu- tional Neural Network (CNN) is proposed also. The core idea of the proposed method is to identify the vehicle and the license plate region using two passes on the same CNN, and then to recognise the characters using a second CNN.

3. Methodology: All Weather Vehicle Recognition System (AWVRS)

3.1. Overall

The All-Weather Vehicle Recognition System is designed to identify vehicles, recognize license plates, and detect human presence in diverse, adverse weather conditions. This system combines dehazing, object detection, and OCR techniques to ensure high recognition accuracy under challenging visual environments.

The system begins with a custom Multi-Attention Dehazing Model, specifically built with a Swin Transformer backbone and multi-scale attention modules. This model effectively restores clarity to hazy images, allowing for a robust subsequent analysis. By leveraging depthwise separable convolutions and Swin Transformer blocks with attention layers, the dehazing model reduces the haze effect at different scales, improving the visibility and contrast of the visual input.

Next, the dehazed image is processed by YOLOv8, an efficient object detection model. This step detects regions of interest (ROIs), particularly vehicles, license plates, and humans. YOLO's rapid detection capabilities ensure accurate bounding boxes, identifying and isolating these key components within the dehazed image.

Finally, the isolated license plate region is processed using an OCR tool to extract textual information from the plate, allowing for license number recognition. The system also uses YOLO labels to infer the vehicle type and check for human presence near the vehicle.

This pipeline creates a reliable, weather-resilient solution for vehicle recognition, license plate identification, and human detection, tailored to real-world applications in security, traffic monitoring, and automated vehicle inspection in challenging environmental conditions.

3.2. Multi-Attention Dehazing (MAD) model as preprocessing filter

The proposed Multi Attention Dehazing Model is designed to improve dehazing in adverse conditions using a multi-attention framework, enabling selective feature refinement at multiple scales. This model integrates a sequence of novel attention modules, which enhance feature extraction and texture consistency for enhanced dehazing outcomes.
The model has:

Depthwise Separable Convolutions: To maintain computational efficiency, the model leverages depthwise separable convolutions in both preprocessing and post-processing layers. These layers significantly reduce the number of parameters by separating spatial filtering and channel mixing, making the model more suitable for real-time applications without sacrificing accuracy.

Swin Transformer Blocks with Attention: The core of the model consists of 4 Swin Transformer blocks, designed for localized attention mechanisms. These blocks capture intricate feature dependencies within different regions of the input image. Following each Swin Transformer block, a channel attention module and pixel attention module are incorporated. Channel Attention modules selectively emphasize the most informative channels, boosting the model's capacity to adaptively respond to complex patterns associated with varying levels of haze.Pixel attention module works for the indivisual pixel affected by haze.

Multi-Scale Attention Mechanism: The model introduces a Multi-Scale Attention layer to manage the multi-scale nature of haze across images. This layer applies depthwise convolutions with (3x3) and (5x5) kernels, capturing both fine and coarse features from hazy images. The multi-scale outputs are combined, and an adaptive attention mechanism weights them, allowing the model to focus on relevant texture details across different scales.

Components:

1. **Layer Normalization 1 (norm1):** Layer normalization is applied to stabilize training by normalizing the inputs across the feature dimension:

$$x = \text{norm1}(x),$$

where $x \in \mathbb{R}^{B\times(M\times M)\times C}$ is the reshaped feature tensor after window partitioning, B is the batch size, M is the window size, and C is the number of feature channels.

2. **Multi-Head Self-Attention (attn):** Self-attention is applied to capture long-range dependencies within the non-overlapping windows of size $M\times M$. Multi-Head Attention computes:

$$\text{attn\_out}, \_ = \text{attn}(x, x, x),$$

where x is used as the query, key, and value inputs. The attention mechanism computes:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where $Q = W_q x$, $K = W_k x$, and $V = W_v x$ are the query, key, and value projections, and d_k is the dimensionality of the key vector.

3. **Residual Connection 1:** A residual connection is added between the input x and the attention output:

$$x = x + \text{attn\_out}.$$

4. **Layer Normalization 2 (norm2):** Another layer normalization is applied to the updated x:

$$x = \text{norm2}(x).$$

5. **Multilayer Perceptron (MLP):** The MLP is a two-layer feedforward network with GELU activation in between. The MLP computes:

$$\text{mlp\_out} = W_2\big(\text{GELU}(W_1 x)\big),$$

where:

- $W_1 \in \mathbb{R}^{C\times 4C}$ expands the feature dimensions by a factor of 4.
- $W_2 \in \mathbb{R}^{4C\times C}$ reduces the dimensions back to C.

The output is added back to the input via another residual connection:

$$x = x + \text{mlp\_out}.$$

6. **Channel Attention (CA):** After Swin Transformer block computations, *channel attention* is applied to refine features across channels. The channel attention mechanism computes:

$$\text{CA}(x) = x \cdot \sigma\big(W_{ca} * \text{AvgPool}(x) + W_{ca} * \text{MaxPool}(x)\big),$$

where σ is the sigmoid activation function, AvgPool and MaxPool are global pooling operations, and W_{ca} is the convolution kernel.

7. **Pixel-Wise Attention (PA):** Pixel-wise attention is applied to emphasize spatially important regions. The attention map is computed as:

$$\text{PA}(x) = x \cdot \sigma\big(W_{pa} * x\big),$$

where W_{pa} is a 1×1 convolution kernel that outputs a single-channel attention map, and σ is the sigmoid function.

8. **Residual Connection in Each Block:** The output of each Swin Transformer block is added back to its input via a residual connection:

$$x'_{k+1} = x'_k + \text{swin}(x'_k),$$

where swin includes Swin Transformer operations, channel attention, and pixel-wise attention.

9. **Reshaping to Original Dimensions:** After all blocks, the feature tensor is reshaped back to its original dimensions:

$$x \in \mathbb{R}^{B\times C\times H\times W}.$$

10. **Multi-Scale Attention (MSA):** Multi-scale attention is applied to capture both local and global features using depthwise separable convolutions:

$$x'' = \text{Conv}_3(x') + \text{Conv}_5(x'),$$

where Conv_3 and Conv_5 are depthwise convolutions with kernel sizes 3×3 and 5×5, respectively. The outputs are concatenated and reduced:

$$x''' = W_{ms}\cdot \text{Concat}\big(\text{Conv}_3(x'), \text{Conv}_5(x')\big).$$

11. **Post-Processing:** The processed features are passed through a *Depthwise Separable Convolution* layer to reconstruct the output image:

$$x_f = \text{post}(x'''),$$

where:

$$\text{post}(x) = \sigma\big(W_{post1} * (W_{post2} * x)\big).$$

12. **Reconstruction with Residual Connection:** The final output is obtained by adding the residual connection back to the processed features:

$$\text{output} = x_f + \text{residual}.$$

Forward Pass

1.Input and Residual Connection

Let the input image be represented as:

$$x \in \mathbb{R}^{C\times H\times W};\ \text{residual} = x.$$

where C is the number of channels (e.g., 3 for RGB images), H and W are the height and width of the image, respectively. The input is saved as a residual connection to ensure that the original structure of the image is retained.

2.Preprocessing Layer

The input image x is passed through a *Depthwise Separable Convolution* layer to extract initial features:

$$x' = \text{pre}(x),$$

$$\text{pre}(x) = \sigma\big(W_{p1} * (W_{p2} * x)\big).$$

- W_{p1} and W_{p2} are the depthwise and pointwise convolution kernels, respectively.
- $*$ represents the convolution operation.
- σ is the activation function.

3.Swin Transformer Blocks

The processed features x' are passed through N Swin Transformer blocks. Each block performs the following operations:

3.1. Self-Attention

Using the Swin Transformer, features are divided into non-overlapping windows of size $M \times M$ and processed with self-attention:

$$z = \text{Attn}\big(\text{LN}(x'')\big),$$

where Attn is the multi-head self-attention operation. LN represents layer normalization. x'' is the input to the current Swin block.

3.2. Feedforward Network (FFN)

The output of the self-attention mechanism is processed through a feedforward network:

3.3. Channel Attention

The enhanced features are refined using **Channel Attention**:

$$x'' = x''\cdot\sigma\big(W_{ca} * \text{AvgPool}(x'') + W_{ca} * \text{MaxPool}(x'')\big),$$

where AvgPool and MaxPool are global average and max pooling operations, respectively. W_{ca} is the convolutional kernel for channel attention. σ is the sigmoid activation function.

3.4. Pixel-Wise Attention

Finally, pixel-wise attention is applied to emphasize important spatial regions:

$$x'' = x''\cdot\sigma\big(W_{pa} * x''\big),$$

where W_{pa} is the convolutional kernel for pixel-wise attention.

Residual Connection in Each Block

The final output of each Swin Transformer block is combined with its input through a residual connection:

$$x'_{k+1} = x'_k + \text{swin}(x'_k).$$

4.Multi-Scale Attention

The output of the Swin Transformer blocks is passed through a **Multi-Scale Attention** module to capture features at different receptive fields:

$$x'' = \text{Conv}_3(x') + \text{Conv}_5(x'),$$

where Conv_3 and Conv_5 are depthwise separable convolutions with kernel sizes of 3×3 and 5×5, respectively. The outputs are concatenated and reduced:

$$x''' = W_{ms} \cdot \text{Concat}\big(\text{Conv}_3(x'), \text{Conv}_5(x')\big),$$

where W_{ms} is the reduction convolution kernel.

5.Post-Processing Layer

The enhanced features are passed through another *Depthwise Separable Convolution* to reconstruct the dehazed image:

$$x_f = \text{post}(x'''),$$
$$\text{post}(x) = \sigma\big(W_{post1} * (W_{post2} * x)\big).$$

6.Reconstruction with Residual Connection

The final output is obtained by adding the residual connection back to the processed features:

$$\text{output} = x_f + \text{residual}.$$

Overall Mathematical Flow

The complete forward pass can be expressed as:

$$\text{output} = \text{post}\big(\text{multi\_scale\_attn}\big(x + \sum_{k=1}^{N} \text{swin}(x)\big)\big) + x.$$

The model's forward pass can be summarized as follows: output = post(multi_scale_attn(residual + features)) + residual, where:

- residual is the input image.
- features are the enhanced features extracted by the Swin Transformer blocks and attention mechanisms.

model = MultiAttentionDehazingModel(num$_blocks = 4)criterion = nn.L1Loss()optimizer = optim.Adam(model.parameters(), lr = 1e-4)$

Optimization and Training Strategy: The training process employs an L1 loss function and utilizes gradient scaling to stabilize updates. The training setup leverages a data loader with images resized to **224*224** and normalized to improve convergence. The model is trained on a dataset comprising various haze levels, with ground truth and hazy images, to ensure the ability to generalize across diverse scenes and haze densities.

This multi-attention architecture enhances the model's ability to adaptively prioritize essential features and suppress irrelevant background information, effectively reducing the haze in both dense and sparse regions of the image. This approach provides a robust solution for real-time dehazing, particularly beneficial in applications requiring high visual clarity in challenging environments, such as autonomous navigation or license plate recognition under low visibility conditions.

Trained Model is stored then for preprocessing pipeline integration to dehaze the input image for the numberplate detection.

3.3. Object Detection and ROIs

The object detection and ROI (Region of Interest) detection stages in our pipeline leverage a pretrained YOLOv8 model to identify specific classes of interest: vehicles, license plates. This process can be summarized as follows:

Object Detection Using YOLOv8

YOLOv8 (You Only Look Once, Version 8) is an advanced, real-time object detection model capable of detecting objects in a single pass through the network. The model performs detection by applying a single forward pass on the input image and produces a set of bounding boxes and class labels associated with detected objects. We apply the YOLOv8 model to identify instances of:

- **Vehicles:**Any general type of vehicle, based on YOLOv8's pre-trained classes.
- **License Plates:** identified as potential areas for character recognition.

The YOLOv8 model outputs bounding box coordinates and class predictions for each detected object. Each prediction provides:

- **Bounding box coordinates** (x_1, y_1, x_2, y_2) for the object's location in the image.
- **Class label**, assigned based on the YOLOv8 detection categories, which allows us to determine if the object belongs to one of our target classes (vehicle, license plate).

ROI Detection and Processing

Using the output of the YOLOv8 detection, we extract Regions of Interest (ROIs) corresponding to specific object classes. This step is crucial for further specialized processing, such as Optical Character Recognition (OCR) on detected license plates. The ROI detection and processing steps include:

1. **Define ROIs**: For each detected object, we check its class label. We assign the bounding box coordinates to specific ROIs based on the label:
    - **Vehicle ROI**: Assigned if the label matches any vehicle-related class.

- **License Plate ROI**: Assigned if the label indicates a license plate or similar descriptor.

2. **Process ROIs for Subsequent Tasks**: Each ROI is stored for later steps. For instance, the *License Plate ROI* is cropped from the image and passed to an OCR model for text recognition.

We have trained the YOLOv8 based model for the detection of the plate by training on the dataset that has image vs xml file. xml file has the information about the location of the plate in the image in for of the co-ordinates.

License Plate Character Recognition

The license plate character recognition in the pipeline leverages Optical Character Recognition (OCR) to identify text within the license plate's detected region.

1. License Plate Detection

Our Yolo based model is trained for the detection of the licence plate region of the any vehicle.

YOLOv8 Training Configuration

A YAML file is generated to specify the dataset structure and label information for YOLOv8 training:

- `path`: Path to the dataset directory.
- `train`: Directory containing training images.
- `val`: Directory containing validation images.
- `nc`: Number of classes (in this case, 1 for license plates).
- `names`: List of class names.

These coordinates define the Region of Interest (ROI) R_{plate} in the dehazed image, which we crop for OCR processing.

Licence Number Recognition

The optical character recognition (OCR) process is conducted using the `Tesseract OCR` library, which applies text recognition techniques to extract and interpret characters from R_{plate}. Let the cropped image be represented as I_{plate} of size $H_{\text{plate}} \times W_{\text{plate}}$. We aim to identify a sequence of characters $C = \{c_1, c_2, \ldots, c_n\}$ in I_{plate}, where n represents the total number of characters detected. Tesseract OCR is an open-source optical character recognition engine. It is particularly effective on binarized images, so preprocessing steps such as grayscaling and thresholding are applied to improve text readability.

For each assigned license plate, the region within the license plate bounding box B_l is extracted from I to isolate the license plate image I_l:

$$I_l = I[x_{l1} : x_{l2}, y_{l1} : y_{l2}]$$

To improve OCR performance, I_l is converted to grayscale I_l^{gray} and thresholded for binarization:

$$I_l^{\text{gray}} = \text{grayscale}(I_l)$$

$$I_l^{\text{bin}} = \text{threshold}(I_l^{\text{gray}}, \theta)$$

where θ is a threshold value chosen to maximize OCR accuracy. Finally, the Tesseract OCR model $\mathcal{R}$ reads the text T from the binarized image:

$$T = \mathcal{R}(I_l^{\text{bin}})$$

We have also Trained an CNN based Character recognition model, which has been integrated instead of the above approach,in the another OCR model as we have build 2 model for the vehicle licence number recognition system. section*DeeperCharRecogCNN Model Documentation

Model Architecture

The ‘DeeperCharRecogCNN‘ is a Convolutional Neural Network (CNN) designed for character recognition. It operates on grayscale images resized to 28×28 pixels. The model includes four convolutional layers, each followed by ReLU activations and max-pooling layers. The feature maps are flattened and passed through three fully connected layers.

Final Text Generation

The final recognised license number is based on confidence score of DEEP CNN model recognition and Tesseract OCR. If the confidence of the Deep CNN model recognised character is more than 0.8, then the output recognised text will be result of Deep CNN model recognition, otherwise baed on the Teserect OCR.

4. Data sets and Experiments

For training the model we have used Tesla 4 GPU in google collab. AWVRS has 3 trained model

1. MAD model: The Multi Attention Dehazing model is trained for 100 epochs with the L1 loss function and learning rate of lr=1e-4 ad the adam optimizer. The loss at the end was **0.2228**.

The traine model is then saved as .pth file for implementation in the LPR model.

2. License Plate Detector model:

- 50 epochs completed in 0.198 hours.
- Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
- Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

License Plate Detector Model Summary

- **Number of layers:** 168
- **Total parameters:** 3,005,843
- **Gradients:** 0
- **GFLOPs:** 8.1

Model Performance Metrics

Metric	Value	Description
Precision (Box P)	0.998	Accuracy of positive predictions.
Recall (R)	0.995	Sensitivity of true positive detection.
mAP@50	0.995	Mean average precision at IoU = 0.5.

Table 1. Performance Metrics - Precision, Recall, and mAP@50

Metric	Value
mAP@50-95	0.841
Preprocessing Time	0.0 ms
Inference Time	0.8 ms
Postprocessing Time	1.7 ms

Table 2. Advanced Performance Metrics

3. Deep Character Recognition Model : The model takes 28*28 input size gray scale image. The model size is 4.56 MB (.pt file)
Epoch 20/20, Loss: 0.011840935222434748 (use grayscale image)
The color image based model that take input 32*64 is of 11.5 MB (.pth file)
Epoch 10/10, Loss: 0.03816051555054

Result

The proposed framework is capable of recognising the vehicle in any waether condition, by itegrating the dehazing model as preprocessing like our MAD model. Other more accurate dehazing model which are trained on large dataset can be employed easily.
Our mdel pipeline also can use character recognition model (especially those traine on licence plate number) instead of our character recognition that compare with Tesseract OCR result for final output.

Example Result: 0: 640x320 1 license-plate, 119.7ms Speed: 3.3ms preprocess, 119.7ms inference,0.8ms postprocess per image at shape (1, 3, 640, 320) Recognized License Plate Text: * ANO1P9687.

We have experimented with many images that are in the link given below.Also the Github link provided below contains everything related to the work.

Model Training and Links

MAD, VLPR models : Google collab notebook where everything is trained and experimented

Deep Char-Recog-CNN model

Character recognition dataset

vehicle image dataset

datasets link (all)

AWVRS github link for trained model and result

References

[1] Margarita Martínez-Díaz, Francesc Soriguera, Autonomous vehicles: theoretical and practical challenges, Transportation Research Procedia,Volume 33, 2018,Pages 275-282,ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2018.10.103.

[2] Y. Lin, P. Wang and M. Ma, ”Intelligent Transportation System(ITS): Concept, Challenge and Opportunity,” 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity)

[3] Kumarmangal Roy, Muneer Ahmad, Norjihan Abdul Ghani, Jia Uddin, Jungpil Shin, ”An Automated Precise Authentication of Vehicles for Enhancing the Visual Security Protocols”, Information, vol.14, no.8, pp.466, 2023.

[4] Aditya Anoop, Harikrishnan G, Keerthi Nair, Sangeetha B, V. Praseedalekshmi, Tasneem Salam H, ”Traffic Surveillance System and Criminal Detection Using Image Processing and Deep Learning”, ICISTSD,2022

[5] D. Srivastava, S. Shaikh and P. Shah, ”Automatic traffic surveillance system Utilizing object detection and image processing,” 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9402496.

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, ”You Only Look Once: Unified Real-Time Object Detection”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[7] F. Alpher and F. Gamow, “Can a computer frobnicate?,” in *CVPR*, 2005.

[8] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. FFA-net: Feature fusion attention network for single image dehazing. In AAAI, pages 11908–11915, 2020

[9] Glenn Jocher and Ayush Chaurasia and Jing Qiu,Ultralytics YOLOv8 https://github.com/ultralytics/ultralytics,2023

[10] Namuk Park and Songkuk Kim. How do vision transformers work? In ICLR, 2022

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In ICCV, pages 10012–10022, 2021.

[12] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. In CVPR, 2022

[13] [Cai et al. 2016] Cai, B.; Xu, X.; Jia, K.; Qing, C.; and Tao, D. 2016. Dehazenet: An end-to-end system for single image haze removal. IEEE Transactions on Image Processing 25(11):5187–5198.

[14] [He, Sun, and Tang 2010] He, K.; Sun, J.; and Tang, X. 2010. Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence 33(12):2341–2353.

[15] [Jiang et al. 2017] Jiang, Y.; Sun, C.; Zhao, Y.; and Yang, L. 2017. Image dehazing using adaptive bi-channel priors on superpixels. Computer Vision and Image Understanding 165:17–32.

[16] [Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In Advances in neural information processing systems, 5998–6008.

[17] [Zhang and Patel 2018] Zhang, H., and Patel, V. M. 2018. Densely connected pyramid dehazing network. In Proceedings of the IEEE conference on computer vision and pattern recognition, 3194–3203.

[18] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In CVPR, pages 2157–2167,2020

[19] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In ICCV, pages 7314–7323, 2019.

[20] Zied Selmi et .al ,Deep Learning System for Automatic License Plate Detection and Recognition DOI: 10.1109/ICDAR.2017.187, 2017

[21] D. Berman, T. Treibitz, and S. Avidan, “Non-local image dehazing,” *IEEE TIP*, 2016.

[22] R. Fattal, “Dehazing using color-lines,” *ACM TOG*, vol. 34, no. 1, 2014.

[23] K. M. He, J. Sun, and X. Tang, “Guided image filtering,” in *ECCV*, 2010.

[24] A. Jiang, B. Foing, I. L. Schlacht, and P. A. Rhodes, “Colour schemes to reduce stress response in a space station,” Jan. 2022.

[25] M. Ju, Z. Gu, and D. Zhang, “Single image haze removal,” *Neurocomputing*, Oct. 2017.

[26] F. Meng, J. Tang, Q. An, and X. Chen, “Decision making with intuitionistic linguistic preference,” *Int. Trans. Oper. Res.*, Feb. 2017.

[27] Q. Zhu, J. Mai, and L. Shao, “Fast haze removal using color attenuation prior,” *IEEE TIP*, vol. 24, no. 11, 2015.

[28] Y. Zheng, J. Zhan, S. He, J. Dong, and Y. Du, “Curricular contrastive regularization for physics-aware single image dehazing,” West Building Exhibit Halls ABC 158.

[29] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, “VoxelNeXt: Fully sparse VoxelNet for 3D object detection and tracking,” CVPR 2023.

[30] X. Li, W. Zuo, and C. C. Loy, “Learning generative structure prior for blind text image super-resolution,” West Building Exhibit Halls ABC 179.

[31] X. Cong, J. Gui, J. Zhang, J. Hou, and H. Shen, “A semi-supervised nighttime dehazing baseline with spatial-frequency aware and realistic brightness constraint,” Arch 4A-E Poster #240.

[32] G. Youk, J. Oh, and M. Kim, “FMA-Net: Flow-guided dynamic filtering and iterative feature refinement with multi-attention for joint video super-resolution and deblurring,” Arch 4A-E Poster #249.

[33] K. Guirguis, J. Meier, G. Eskandar, M. Kayser, B. Yang, and J. Beyerer, “NIFF: Alleviating forgetting in generalized few-shot object detection via neural instance feature forging,” West Building Exhibit Halls ABC 343.

[34] Y. Du, C. Lei, Z. Zhao, and F. Su, “iKUN: Speak to trackers without retraining,” Arch 4A-E Poster #437.

[35] Adapting the Tesseract Open Source OCR Engine for Multilingual OCR., url = https://storage.googleapis.com/pub-tools-public-publication-data/pdf/35248.pdf,