License Transformer : Unified End-to-End License Plate Detection and Recognition

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
Master of Technology
IN
Artificial Intelligence

BY

Shishir Madhab Roy

under the guidance of

Prof. Chandra Sekhar Seelamantula

Department of Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

May, 2025

Declaration of Originality

I, **Shishir Madhab Roy**, with SR No. **04-03-06-10-51-23-1-22452** hereby declare that the material presented in the report titled

License Transformer : Unified End-to-End License Plate Detection and Recognition

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2023-25**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: 30 May 2025 Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Prof. Chandra Sekhar Seelamantula Advisor Signature

Acknowledgements

I extend my heartfelt gratitude to Prof. Chandra Sekhar Seelamantula, my faculty advisor, for his invaluable guidance and thought-provoking discussions that profoundly shaped the ideas in this report. His constructive feedback, insightful suggestions, and deep expertise in the field significantly elevated the quality of this work and guided the direction of my research.
I am also grateful to my lab mates for their engaging discussions and unwavering support, which were instrumental throughout the course of this project.
Lastly, I express my deepest appreciation to my family and friends for their constant encouragement and support, which have been a pillar of strength during this academic journey.

Abstract

Automatic License Plate Recognition (ALPR) systems play a vital role in applications such as traffic management, toll collection, and parking management. While significant advancements have improved the robustness and reliability of these systems, traditional ALPR approaches often depend on multiple independent models for license plate detection and character recognition. This reliance introduces inefficiencies, requiring either extensive and diverse training data for segmentation-free methods or robust segmentation algorithms for segmentation-based methods. We introduce the License Transformer, a transformer-based model that unifies detection and recognition into a single, efficient process. By eliminating the need for separate intermediate steps typically executed between detection and recognition, our approach enhances both the speed and accuracy of ALPR systems. This innovation not only optimizes the performance of existing applications but also improves the accessibility and scalability of ALPR technology, making it adaptable to a broader range of real-world use cases.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Automatic License Plate Recognition (ALPR) systems are widely used across the world for traffic management, highway toll station, parking entrance and exit management. Extensive progress has been made to make ALPR more robust and reliable. However, these ALPR systems often include two or more independent models to detect and recognize the license plates. To alleviate the issue, we proposed a transformer based model which is capable of detecting and recognizing the license plate characters.

ALPR systems can generally be classified into two categories - (i) Segmentation Free Methods (Detection - Recognition) and (ii) Segmentation Based Methods (Detection - Segmentation - Recognition). Segmentation Free Methods recognizes the LP characters directly from the LP features or LP image, this results in robustness and efficiency. But to achieve this segmentation free methods need extensive and diverse training data. On the other hand, Segmentation Based Methods segments out the LP characters and then recognize the characters individually. This allows segmentation based methods to achieve high accuracy with less training data provided the segmentation algorithm is robust and accurate. This makes the segmentation based methods very inefficient.

In both approaches, several operations are typically carried out by the CPU between detection and recognition. The proposed License Transformer streamlines the process by performing the detection and recognition simultaneously. This integration reduces the need for separate intermediate steps, enhancing both the speed and accuracy of ALPR systems, and contributing to the goal of making these systems more accessible and scalable for diverse use cases.

Chapter 2

Related Work

ALPR systems consists of two key-steps - (i) LP Detection and (ii) LP Recognition.

2.1 LP Detection

Traditional methods rely on handcrafted features and shallow architectures. These approaches typically exploit characteristics such as the rectangular shape of the license plate, fixed character length, and specific colors. Common techniques include edge-based, color-based, texture-based, and character-based methods. While edge and color-based methods are often fast and suitable for real-time applications, they are sensitive to changes in illumination and complex backgrounds. These methods fail in unconstrained environments where lighting conditions vary, or colors shift under strong light.

To address these limitations, texture-based and character-based methods have been explored. Texture-based methods use techniques like wavelet transforms and other advanced processing to enhance robustness against illumination changes. Character-based methods, on the other hand, focus on detecting and recognizing the individual characters of the license plate, allowing for greater adaptability in diverse environmental conditions. However, these methods are often computationally intensive and can still be affected by character-like backgrounds. Overall, traditional methods are effective in controlled settings, their reliance on low-level features and additional processing steps limits their performance.

Deep learning-based methods typically leverage neural networks to extract features and regress location parameters. These methods are generally supervised and often rely on popular deep learning backbones like ResNet [12], DenseNet [13] for feature extraction. While these networks offer high accuracy, they can be computationally expensive. To address this, some works focus on designing lightweight networks that are more efficient, enabling faster real-time

applications without compromising performance.

For location detection, deep learning methods commonly use object detection techniques, which can be categorized into anchor-based, two-stage, and anchor-free approaches. Anchor-based methods, such as YOLO [9] and SSD [10], are widely adopted for license plate detection due to their lower computational cost. Two-stage methods like Faster-RCNN [11] offer higher accuracy but are less efficient. Anchor-free methods, on the other hand, avoid the need for anchors and intersection over union (IoU) calculations, making them more efficient and easier to train. Overall, deep learning-based methods are more robust due to their ability to learn from large datasets and adapt to diverse conditions.

2.2 LP Recognition

LP Recognition can be classified into two categories - (i) Segmentation Based Methods and (ii) Segmentation Free Methods.

Segmentation-based methods focus on separating individual characters from a license plate for subsequent recognition. This process typically begins with the creation of a binary image of the license plate, followed by character boundary detection using techniques such as horizontal pixel projection. The accuracy of character segmentation is crucial, as any errors at this stage directly impact the overall recognition performance.

Character recognition can then be approached using two main methods: template matching and learning-based approaches. Template matching compares segmented characters to predefined templates, classifying them based on the closest match. While simple and effective with limited data, this approach may struggle with font variations or distortions. On the other hand, learning-based methods, including traditional techniques like Hidden Markov Models and Support Vector Machines, offer more robust performance by learning from larger datasets.

In recent years, deep learning techniques have gained significant popularity for both character segmentation and recognition. Frameworks such as YOLO have been adapted to simultaneously detect and classify characters from license plates. Additionally, advanced models like Mask R-CNN [15] have been employed for more accurate segmentation. These deep learning-based methods, with their ability to handle large, varied datasets, have become increasingly dominant in modern ALPR systems, providing superior robustness and accuracy in unconstrained environments.

Segmentation-free methods have gained popularity by avoiding the complexity of character segmentation. These methods treat license plate characters as a sequence, framing the task as a sequence labeling problem. Both CNN-based and RNN-based approaches have been employed to address this challenge.

In [16, 1], CNNs are used directly for character recognition. For instance, CNN classifiers are applied without explicit segmentation, while others use deep feature extraction followed by classifiers. However, CNN-based methods may struggle with varying license plate lengths, as a single model may not adapt well. To overcome this, Fully Convolutional Networks, using semantic segmentation to identify character areas, have been effective.

RNN-based methods [18, 19], which excel in sequence labeling tasks, have also been widely applied in ALPR. Bidirectional Recurrent Neural Networks combined with Connectionist Temporal Classification [17] have been used for character recognition. A limitation of RNNs is their reliance on previous time steps, which makes parallelization difficult.

Segmentation-free methods streamline ALPR by bypassing character segmentation, enabling more direct recognition of characters. These approaches, particularly those using RNNs and CNNs, show strong potential for real-time and robust ALPR in diverse environments.

Chapter 3

Dataset

3.1 Overview of Chinese City Parking Dataset

All experiments are performed on the Chinese City Parking Dataset (CCPD) [1], which is a large-scale benchmark designed to evaluate License Plate (LP) Detection and Recognition systems under uncontrolled real-world conditions. It consists of over 350,000 unique vehicle images captured in various challenging environments, including different weather conditions, lighting variations, rotations, and occlusions. The resolution of each image is 720 (width) $\times$ 1160 (height) $\times$ 3 (channels). Each image in the data set has only one LP. Each LP number is made up of a Chinese character, followed by a letter and then five letters or numbers. The data set is divided into many sub-data sets 3.2. The data set is divided into a train validation test set in the ratio 5:1:4.

	Description
CCPD-Base	The only common feature of these photos is the inclusion of a license plate.
CCPD-DB	Illuminations on the LP area are dark, uneven or extremely bright.
CCPD-FN	The distance from the LP to the shooting location is relatively far or near.
CCPD-Rotate	Great horizontal tilt degree ($20^\circ \sim 50^\circ$) and the vertical tilt degree varies from -10° to 10°.
CCPD-Tilt	Great horizontal tilt degree ($15^\circ \sim 45^\circ$ degrees) and vertical tilt degree ($15^\circ \sim 45^\circ$).
CCPD-Blur	Blurry largely due to hand jitter while taking pictures.
CCPD-Weather	Images taken on a rainy day, snow day or fog day.
CCPD-Challenge	The most challenging images for LPDR to date.
CCPD-NP	Images of new cars without a LP.

Figure 3.1: Descriptions of different sub-datasets of CCPD dataset.

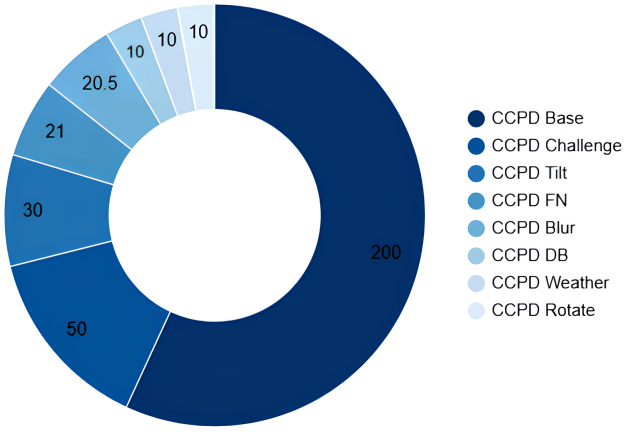

Figure 3.2: Number of images in different sub-datasets of CCPD dataset. All figures are in thousands.

Figure 3.3: Sample images from the CCPD dataset illustrating diverse environmental and imaging conditions.

Chapter 4

Revisiting Attention

4.1 Multi-Head Attention

In the multi-head attention mechanism [14], given a query element (e.g., a target word in the output sentence) and a set of key elements (e.g., source words in the input sentence), the module dynamically aggregates key contents based on attention weights that reflect the compatibility of query-key pairs. To enable the model to focus on content from diverse representation subspaces and positions, the outputs of multiple attention heads are linearly combined using learnable weights.

Formally, let $q \in \Omega_q$ denote a query element with representation feature $z_q \in \mathbb{R}^C$, and $k \in \Omega_k$ denote a key element with representation feature $x_k \in \mathbb{R}^C$, where C represents the feature dimension, and Ω_q and Ω_k define the sets of query and key elements, respectively. The multi-head attention feature is computed by:

$$\text{MultiHeadAttn}(z_q, x) = \sum_{m=1}^{M} W_m \left[\sum_{k \in \Omega_k} A_{mqk} W'_m x_k \right] \tag{4.1}$$

where m indexes the attention head, $W'_m \in \Re^{C_v \times C}$ and $W_m \in \Re^{C \times C_v}$ are of learnable weights and $C_v = C/M$. The attention weights $A_{mqk} \propto exp\{\frac{z_q^T U_m^T v_m x_k}{\sqrt{C_v}}\}$ are normalized as $\sum_{q \in \Omega_q} A_{mqk} = 1$, in which $U_m, V_m \in \Re^{C_v \times C}$ are also learnable weights. To differentiate different spatial positions, positional embeddings are added to the representation features z_q and x_k.

4.2 Deformable Attention

Inspired by deformable convolution [7], the deformable attention module only attends to a small set of key sampling points around a reference point, regardless of the spatial size of the feature

maps, as shown in Figure 4.1. By assigning only a small fixed number of keys for each query, the issues of convergence and feature spatial resolution can be mitigated.

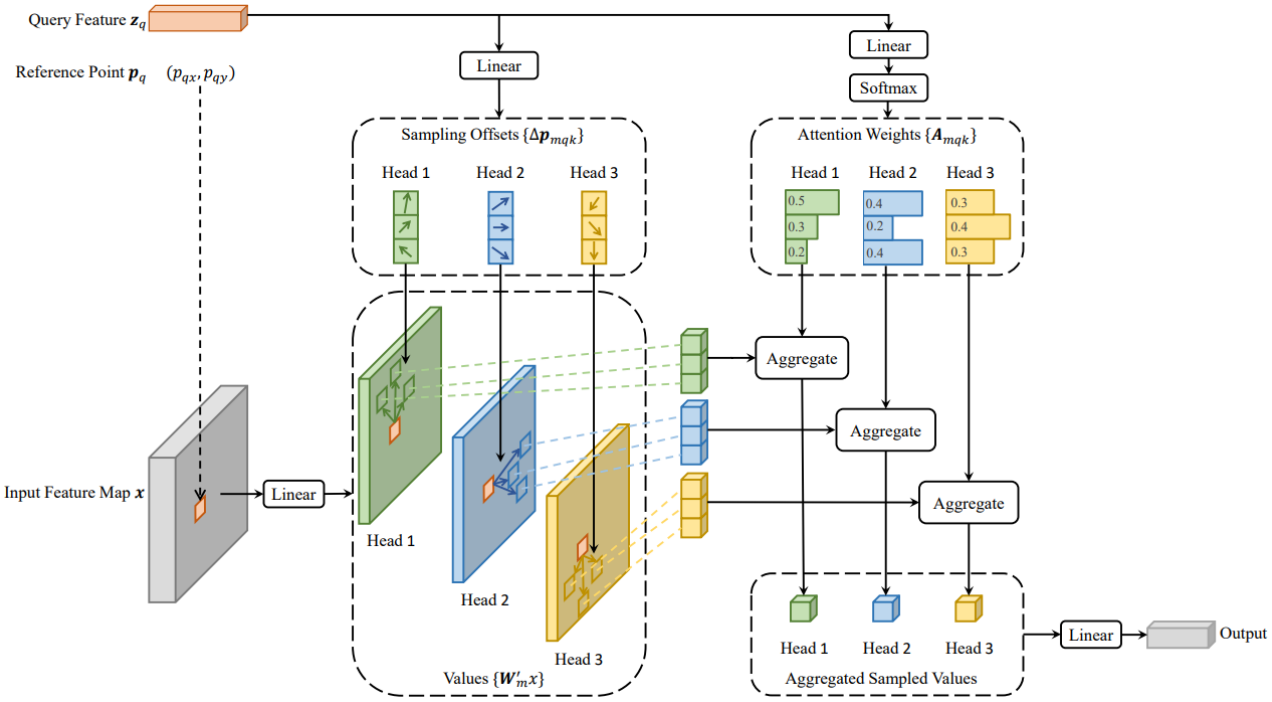

Figure 4.1: Illustration of Deformable Attention Module.

Given an input feature map $x \in \Re^{C \times H \times W}$, let q index a query element with content feature z_q and a 2-d reference point p_q, the deformable attention [3] feature is calculated by:

$$\text{DeformAttn}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} W'_m x(p_q + \Delta p_{mqk}) \right] \tag{4.2}$$

where m indexes the attention head, k indexes the sampled keys, and K is the total sampled key number $K << HW$. Δp_{mqk} and A_{mqk} denote the sampling offset and attention weight of the k^{th} sampling point in the m^{th} attention head, respectively. The scalar attention weight A_{mqk} lies in the range $[0, 1]$, normalized by $\sum_{k=1}^{K} A_{mqk} = 1$. $\Delta p_{mqk} \in \Re^2$ are of 2-d real numbers with unconstrained range. As $p_q + \Delta p_{mqk}$ is fractional, bilinear interpolation is applied in computing $x(p_q + \Delta p_{mqk})$. Both Δp_{mqk} and A_{mqk} are obtained via linear projection over the query feature z_q.

Chapter 5

License Transformer

5.1 The LITR : Overview

LITR consists of a ResNet-based Backbone, a Hybrid Encoder, and a Transformer Decoder with auxiliary Prediction Heads. The architecture of LITR is illustrated in Figure 5.1. We feed the features from the last three stages of the backbone, namely, $\{\mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_5\}$ into the encoder. The Hybrid Encoder transforms multi-scale features into a sequence of image features through Attention-based Intra-scale Feature Interaction (AIFI) and CNN-based Cross-scale Feature Fusion (CCFF). Subsequently, the Uncertainty-minimal Query Selection selects a fixed number of encoder features to serve as initial object queries for the decoder. Finally, the Decoder with auxiliary Prediction Heads iteratively optimizes object queries to generate categories and boxes. Furthermore, to facilitate efficient training and quick convergence, we use Denoising Queries.

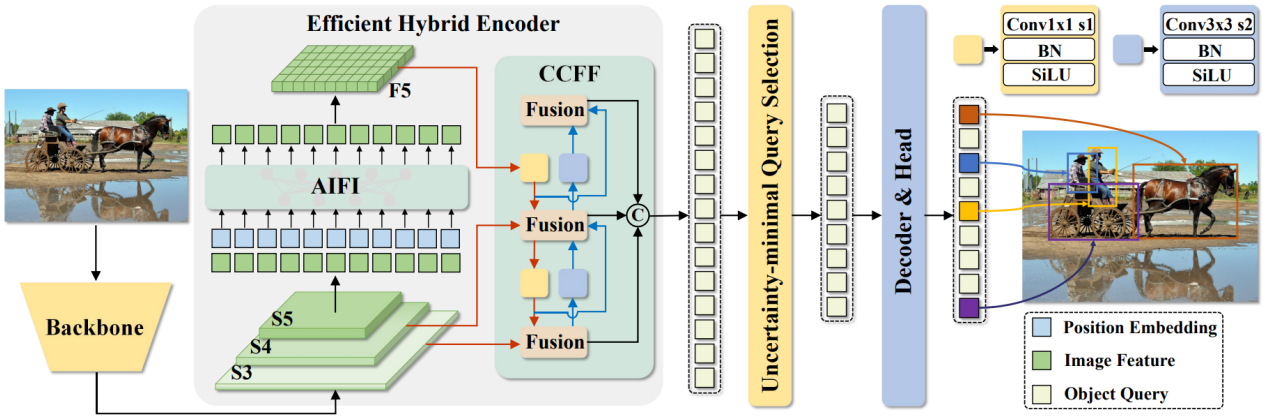

Figure 5.1: The License Transformer (LITR) Architecture.

5.2 Hybrid Encoder

Based on the analysis of different encoder architectures, [5] has proposed an efficient Hybrid Encoder. Specifically, this Hybrid Encoder consists of two blocks, viz. the Attention-based Intra-scale Feature Interaction (AIFI) and the CNN-based Cross-scale Feature Fusion (CCFF).

The AIFI block reduces the computational cost by performing the intra-scale interaction only on $\mathbf{S}_5$ with the multi-head self-attention encoder [14]. The reason is that applying the self-attention operation to high-level features captures the connection between features of the same object (license plate) and separates different objects (license plates) with each other and background, which helps in the localization and recognition of objects (license plates) by subsequent modules. However, the intra-scale interactions of lower-level features are not necessary due to the risk of duplication and confusion with high-level feature interactions.

The CCFF block is optimized using a cross-scale fusion module that consists of multiple fusion blocks with convolutional layers along the fusion path. Each fusion block integrates features from two adjacent scales into a new feature, as shown in Figure 5.2. It employs two 1×1 convolutions to adjust channel dimensions, followed by N RepBlocks [8] for feature fusion, with the outputs combined via element-wise addition.

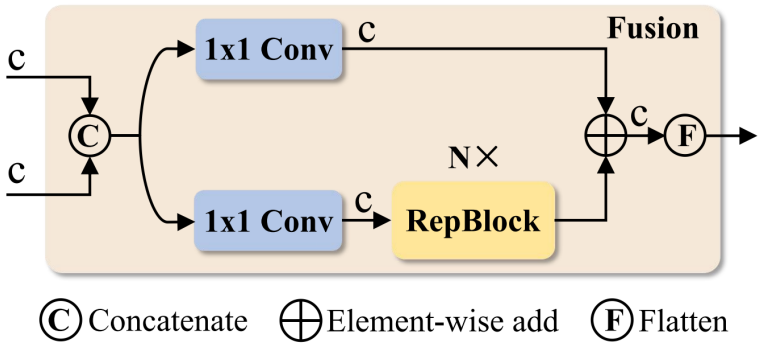

Figure 5.2: Architecture of Fusion Block.

Mathematically, the hybrid encoder can be represented as:

$$\begin{aligned} \mathbf{Q} = \mathbf{K} = \mathbf{V} &= \text{Flatten}(\mathbf{S}_5) \\ \mathbf{F}_5 &= \text{Reshape}(\text{AIFI}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \\ \mathbf{O} &= \text{CCFF}(\{\mathbf{S}_3, \mathbf{S}_4, \mathbf{F}_5\}) \end{aligned} \tag{5.1}$$

where $\mathsf{Reshape}$ represents restoring the shape of the flattened feature to the same shape as $\mathbf{S}_5$.

5.3 Prediction Heads

The final prediction is computed by a 3-layer perceptron (MLP) and a linear projection layer corresponding to each 7 characters of license plate. The MLP predicts the normalized center coordinates, height and width of the license plate w.r.t. the input image, and the linear layers predict the character labels via a sigmoid function. Since we predict a fixed-size set of N license plates, where N is usually much larger than the actual number of license plates in an image, we compute the average of the confidence scores for label predictions and threshold over it to discard low confidence predictions, thus eliminating the need of Non-Maximum Suppression (NMS) post-processing.

Each decoder layer and hybrid encoder has its own set of prediction heads. The predictions from all decoder layers and hybrid encoder are optimized as per the Set Loss. Furthermore, the predictions from hybrid encoder helps in selecting the object queries from the encoder outputs and acts as the initial guess for the decoder. The outputs of a decoder layer acts as the input for the next decoder layer while (refined) bounding box prediction acts as the guess for the next decoder layer.

5.4 Uncertainty-minimal Query Selection

The DETR model [2] employed learnable object queries, but due to the difficulty of optimizing these object queries the models used to take a lot of time to converge. Furthermore, their detection performance were below par the State-of-the-Art (SOTA) models of that time. Subsequent works [3, 4, 5] propose various query selection schemes to reduce the difficulty of optimizing object queries. These query selection schemes have one thing in common that is they use the confidence score to select the top-K features from the encoder to initialize object queries. The confidence score represents the likelihood that the feature includes foreground objects.

Here, we take the sum of the confidence scores corresponding to each label and from this consolidated confidence score we select the top-K features from the encoder output as object queries for the decoder.

5.5 Decoder

In the decoder, cross-attention and self-attention modules utilize object queries as query elements. Cross-attention module extract features from encoder output feature maps, using them as key elements, while self-attention module enable interaction among object queries, with keys derived from the queries themselves. We replace cross-attention modules with multi-scale deformable attention modules, while self-attention modules are kept unchanged. Each object

query predicts a 2D normalized reference point coordinate $\hat{p}_q$ from its embedding via a learnable linear projection and sigmoid function.

The multi-scale deformable attention module extracts image features near the reference point. To simplify the optimization process, the detection head is designed to predict bounding boxes as relative offsets with respect to this reference point, which serves as the initial estimate of the box center. By predicting these relative offsets, the decoder's learned attention mechanism exhibits a strong correlation with the predicted bounding boxes, thereby resulting in faster training convergence.

5.6 Denoising Queries

We collect all ground truth objects in an image and add noise to their bounding boxes and labels. We call this collection a group. We have multiple groups (noised version of all ground truth objects) to maximize the effect of denoising learning. Noised bounding box generation includes scaling the width and height of the bounding box and shifting the center of the bounding box. We make sure the shifted center is still in the original bounding box. Ground truth labels are flipped randomly to generate noised labels. This forces the model to learn strong box-label relationships [6]. Denoising is only considered in training, during inference the denoising part is removed, leaving only the matching part.

The noised labels gives us the queries and the noised boxes gives us the corresponding initial reference points. The noised labels is an integer and the query is supposed to be a vector of dimension hidden_dim. Thus, we use learnable character embeddings for each of the 7 labels of dimension hidden_dim. We add the 7 character embeddings corresponding to the noised labels and these are used as object queries for the decoder.

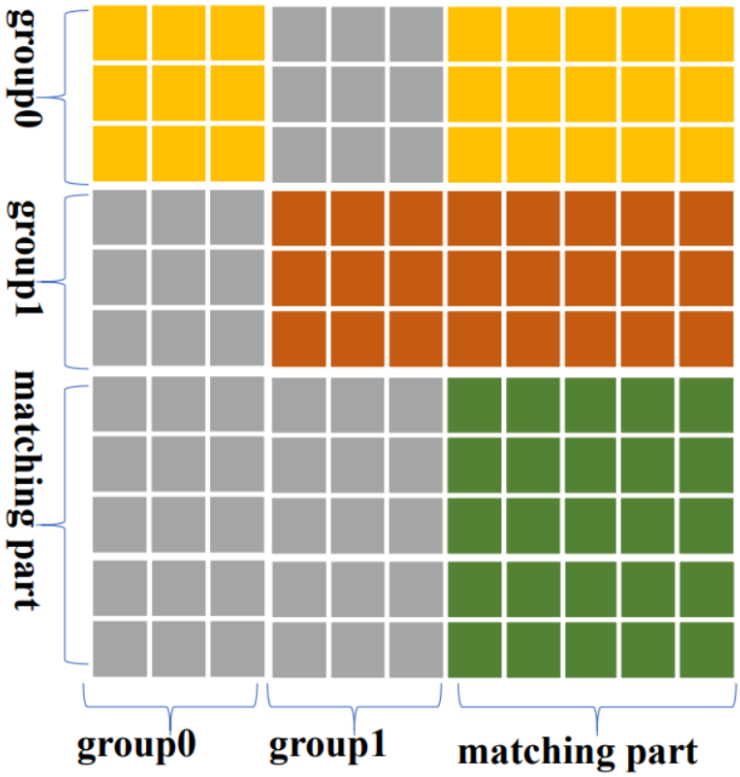

Figure 5.3: Attention Mask Example. The group0 (group1) queries can attend to group0 (group1) queries and matching queries only.The matching queries can attend to matching queries only.

We don't want information leakage between different groups and matching part (real object queries). Thus, we initialize attention mask, example shown in Figure 5.3, such that real object queries can only attend to each other; and not attend to any denoising queries. Similarly, the denoising queries from a group can only attend to other denoising queries from the same group and not attend to any denoising queries from other groups.

Chapter 6

Results

6.1 Results

The following table shows the results of all the models on different sub-datasets of CCPD as percentage. We consider a prediction to be correct only when all the LP numbers predicted are correct.

	FPS	**Base**	**DB**	**FN**	**Rotate**	**Tilt**	**Weather**	**Challenge**
Faster-RCNN+HC	13	97.2	94.4	90.9	82.9	87.3	85.5	76.3
RPNet	61	98.5	96.9	94.3	90.8	92.5	87.9	85.1
CenterNet+RCNN	26	99.6	92.0	93.9	98.2	95.6	98.4	83.4
YOLOv10s+HC	46	99.3	87.9	85.3	92.5	87.0	96.3	83.7
YOLOv11s+HC	58	98.9	88.8	86.1	92.7	86.9	95.6	82.8
YOLOv12s+HC	59	99.5	87.9	85.3	92.8	87.9	96.7	84.5
LITR-R18	50	99.4	86.7	87.0	93.5	86.5	97.9	84.0
LITR-R50	41	99.5	89.0	90.1	93.6	86.9	98.5	84.2

Table 6.1: Recognition performance of the models on different sub-datasets as percentage. A prediction is considered correct only when all the LP numbers predicted are correct. FPS is the number of images the model processes per second.

Chapter 7

Discussions

7.1 Limitations

It is widely recognized that Detection Transformers exhibit suboptimal performance in detecting small objects, a limitation that also affects our proposed License Transformer. We anticipate that as Detection Transformer architecture advances, this challenge will be comprehensively addressed, thereby improving the performance of the License Transformer accordingly. Furthermore, Detection Transformers, and by extension License Transformer, need huge data for training.

7.2 Key Findings

Detection Transformers have demonstrated their versatility across a wide range of computer vision tasks, including object detection, semantic segmentation, panoptic segmentation, instance segmentation, keypoint detection, and object tracking. In this work, we propose License Plate Recognition as an additional task that can be effectively addressed using Detection Transformers, further expanding their applicability.

7.3 Future Work

The proposed License Transformer model is limited to predict only 7 character in a license plate, which is by design as we are training to detect and recognize Chinese license plates. We can easily extend this work to recognize variable length license plate numbers which are fairly common by changing the loss function to Connectionist Temporal Classification loss, setting a max length and defining a fixed vocab.

Model Hyperparameters

Item	**LITR-R18**	**LITR-R50**
optimizer	AdamW	AdamW
base learning rate	1e-4	1e-4
backbone learning rate	1e-4	1e-5
weight decay	0.0001	0.0001
clip gradient norm	0.1	0.1
number of AIFI Layers	1	1
number of Rep Blocks	3	3
model dimension	256	256
mlp dimension	768	768
number of encoder heads	8	8
number of decoder heads	8	8
number of decoder layers	6	6
number of sampling points	5	5
number of object queries	100	100
number of denoising queries	100	100
bbox loss weight	11.0	11.0
GIoU loss weight	5.0	5.0
label loss weight	1.0	1.0
α in label loss	0.25	0.25
γ in label loss	2.0	2.0

Table 7.1: Values of main hyperparameters of LITR-R18 and LITR-R50.

Model Predictions Visualized

Figure 7.1: Visualization of LITR-R50 predictions under different conditions.

Bibliography

[1] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, Liusheng Huang. *Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline*. European Conference on Computer Vision (ECCV), 2018. 4, 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko. *End-to-End Object Detection with Transformers*. European Conference on Computer Vision (ECCV), 2020. 12

[3] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, Jifeng Dai. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. International Conference on Learning Representations (ICLR), 2021. 9, 12

[4] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, Lei Zhang. *DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR*. International Conference on Learning Representations (ICLR), 2022. 12

[5] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, Jie Chen. *DETRs Beat YOLOs on Real-time Object Detection*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 11, 12

[6] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M. Ni, Lei Zhang. *DN-DETR: Accelerate DETR Training by Introducing Query DeNoising*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. 13

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei. *Deformable Convolutional Networks*. IEEE International Conference on Computer Vision (ICCV), 2017. 8

[8] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, Jian Sun. *RepVGG: Making VGG-style ConvNets Great Again*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. 11

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 3

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, *SSD: Single Shot MultiBox Detector*. European Conference on Computer Vision (ECCV), 2016. 3

[11] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. Conference on Neural Information Processing Systems (NeurIPS), 2015. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. *Deep Residual Learning for Image Recognition*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 2

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. *Densely Connected Convolutional Networks*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017. 2

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. *Attention Is All You Need*. Conference on Neural Information Processing Systems (NeurIPS), 2017. 8, 11

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. *Mask R-CNN*. IEEE International Conference on Computer Vision (ICCV), 2017. 3

[16] Sergey Zherzdev, Alexey Gruzdev. *LPRNet: License Plate Recognition via Deep Neural Networks*. arXiv, 2018. 4

[17] Alex Graves, Santiago Fernández, Faustino Gomez, Jürgen Schmidhuber. *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. International Conference on Machine Learning (ICML), 2006. 4

[18] Hui Li, Chunhua Shen. *Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs*. arXiv, 2016. 4

[19] H. Li, P. Wang, C. Shen. *Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks*. IEEE Transactions on Intelligent Transportation Systems, 2019. 4