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

Shishir Madhab Roy, Guided by : Prof. Chandra Sekhar Seelamantula

Indian Institute of Science, Bangalore

Motivation and Scope

- 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 combine two or more independent models to detect and recognize the License Plates (LP).
- **Problem Statement :** This project aims to alleviate this issue by developing an Unified End-to-End model capable of detecting and recognizing License Plates (LP).

Dataset : CCPD

- Large-scale benchmark designed to evaluate ALPR systems under uncontrolled real-world situations.
- Consists of over 350K unique images captured in various challenging environments, including different weather conditions, lighting variations, rotations, and occlusions.
- LP number is made up of one Chinese character, followed by a letter and then five letters or numbers.

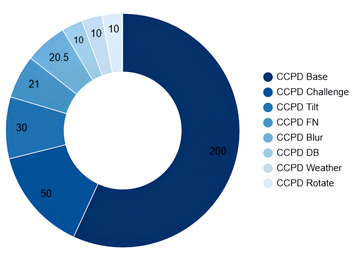

Figure 1. Number of images in different sub-datasets of CCPD. All figures are in thousands.

LITR : Architecture

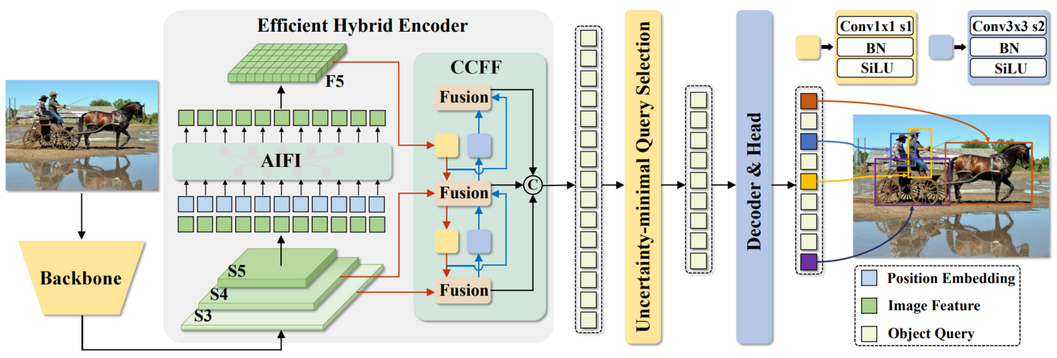

Figure 2. License Transformer (LITR) Architecture.

Architecture Overview

- A ResNet-based backbone is used to extract feature maps from the input images.
- Feature maps from the final three stages of the backbone are fed into the encoder.
- The Hybrid Encoder transforms these multi-scale features into a sequence of image features through the AIFI Block and the CCFF Block.
- Then, the Uncertainty-minimal Query Selection selects a fixed number of encoder features as initial object queries for the decoder.
- Finally, the Multi-Scale Deformable Decoder with auxiliary prediction heads iteratively optimizes object queries to predict bounding boxes and corresponding labels.

Attention-based Intra-scale Feature Interaction (AIFI) Block

- This block performs intra-scale interaction only on S_5 with the multi-head self-attention encoder.
- Applying the self-attention operation to high-level features captures the connection between features of the same object (LP) and separates different objects (LP) with each other and background, which helps in the localization and recognition of objects (LP) by subsequent modules.
- The intra-scale interactions of lower-level features are not necessary due to the risk of duplication and confusion with high-level feature interactions.

CNN-based Cross-scale Feature Fusion (CCFF) Block

- This block is optimized using a cross-scale fusion module inspired by Path Aggregation Network that consists of multiple fusion blocks with convolution layers along the fusion path.
- Each fusion block integrates features from two adjacent scales into a new feature, as shown in Figure 2.

Prediction Heads

- Prediction Head consists of a 3-layer perceptron (MLP) which predicts the bounding box and a linear projection layer which predicts the label logits for each 7 characters of LP.
- Since we predict a fixed-size set of N license plates, 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.

Uncertainty-minimal Query Selection

- The 7 label logits predictions from the encoder outputs are added to get a consolidated score.
- We select top N encoder output features, for which the corresponding consolidated score is maximum, as object queries for decoder.
- The corresponding bounding box prediction becomes the initial bounding box guess for decoder.

Decoder

- Decoder uses multi-head self attention to capture interactions between object queries and multi-scale deformable attention to capture interactions between object queries and encoder outputs.

Deformable Attention

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 3.

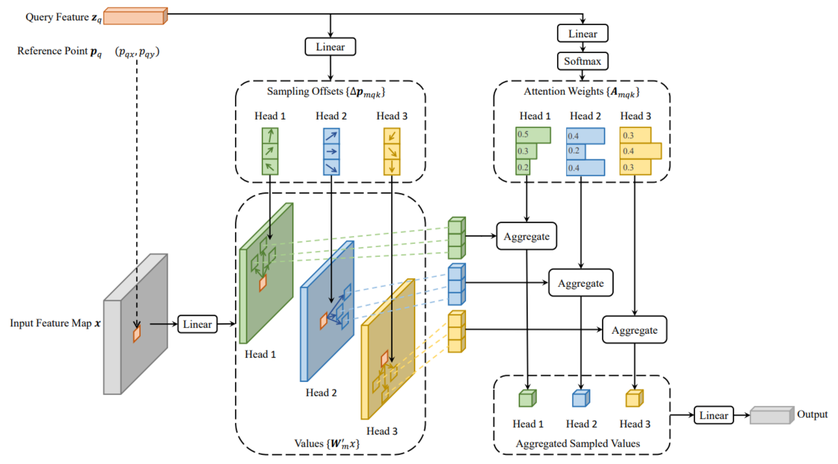

Figure 3. Illustration of Deformable Attention Module.

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 feature is calculated by:

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

Denoising Queries

- We collect all ground truth LP in an image and add noise to their bounding boxes and labels. This collection is called a group. We have multiple groups to maximize the effect of denoising learning.
- The noised labels gives us the object queries and the noised boxes gives us the corresponding initial reference points for the decoder.
- We don't want information leakage between different groups and matching part (real object queries).
- Thus, we initialize attention mask such that real object queries can only attend to each other, and the denoising queries from a group can only attend to other denoising queries from the same group.

Hungarian Matching and Set Loss

- Let us denote by y the ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ the set of N predictions.
- To find a bipartite matching between these two sets we search for a permutation of N elements $\sigma \in \mathfrak{S}_N$ with the lowest cost:

$$\hat{\sigma} = \arg \min_{\sigma \in \mathfrak{S}_N} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right)$$

- We define pair-wise matching cost as:

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = \sum_{j=1}^{7} -\mathbf{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_{ij}) + \mathbf{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

- We define the loss as:

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \Big[\sum_{j=1}^{7} -\log \hat{p}_{\hat{\sigma}(i)}(c_{ij}) + \mathbf{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\hat{\sigma}(i)}\right) \Big]$$

Results

	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 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.

Limitaions

- DETR, and by extension LITR, need huge amount of data for training.
- It is widely recognized that DETRs exhibit suboptimal performance in detecting small objects, a limitation that also affects LITR.

Future Work

- This LITR model is limited to predict only 7 character in a LP, which is by design as we are training and testing over Chinese LP dataset.
- We can extend this work to recognize variable length LP numbers, which are fairly common, by changing the loss function to a variant of CTC loss, setting a max length and defining a fixed vocab.