Car Plate Recognition and Reconstruction with DeepLearning

Computer Vision
Prof.ssa Irene Amerini
A.A. 2024/2025

Luca Conti, 1702084
Kevin Giannandrea, 2202212
Mattia Tarantino, 1933469

SAPIENZA
UNIVERSITÀ DI ROMA

Introduction

Goal:

- Design and implement a two-stage pipeline for license plate (LP) detection and recognition, following the methodology outlined by Tao et al. (2024)
  1. First stage: **YOLOv5** for LP **detection**
  2. Second stage: **PDLPR** for LP **recognition**

sensors MDPI

Article

A Real-Time License Plate Detection and Recognition Model in Unconstrained Scenarios

Lingbing Tao [1], Shunhe Hong [1], Yongxing Lin [1,2], Yangbing Chen [1], Pingan He [3] and Zhixin Tie [1,2,*]

1 School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; lb_tao@zstu.edu.cn (L.T.); 202020502017@mails.zstu.edu.cn (S.H.); eluian@zstu.edu.cn (Y.L.); cyb@zstu.edu.cn (Y.C.)
2 Keyi College, Zhejiang Sci-Tech University, Shaoxing 312369, China
3 School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China; pinganhe@zstu.edu.cn
* Correspondence: tiezx@zstu.edu.cn

Introduction

Goal:

- Implement a **simple baseline**, train and evaluate it with the metrics used by Tao et al. (2024)
- **Compare** the performance of the proposed model with the baseline

sensors

MDPI

Article

A Real-Time License Plate Detection and Recognition Model in Unconstrained Scenarios

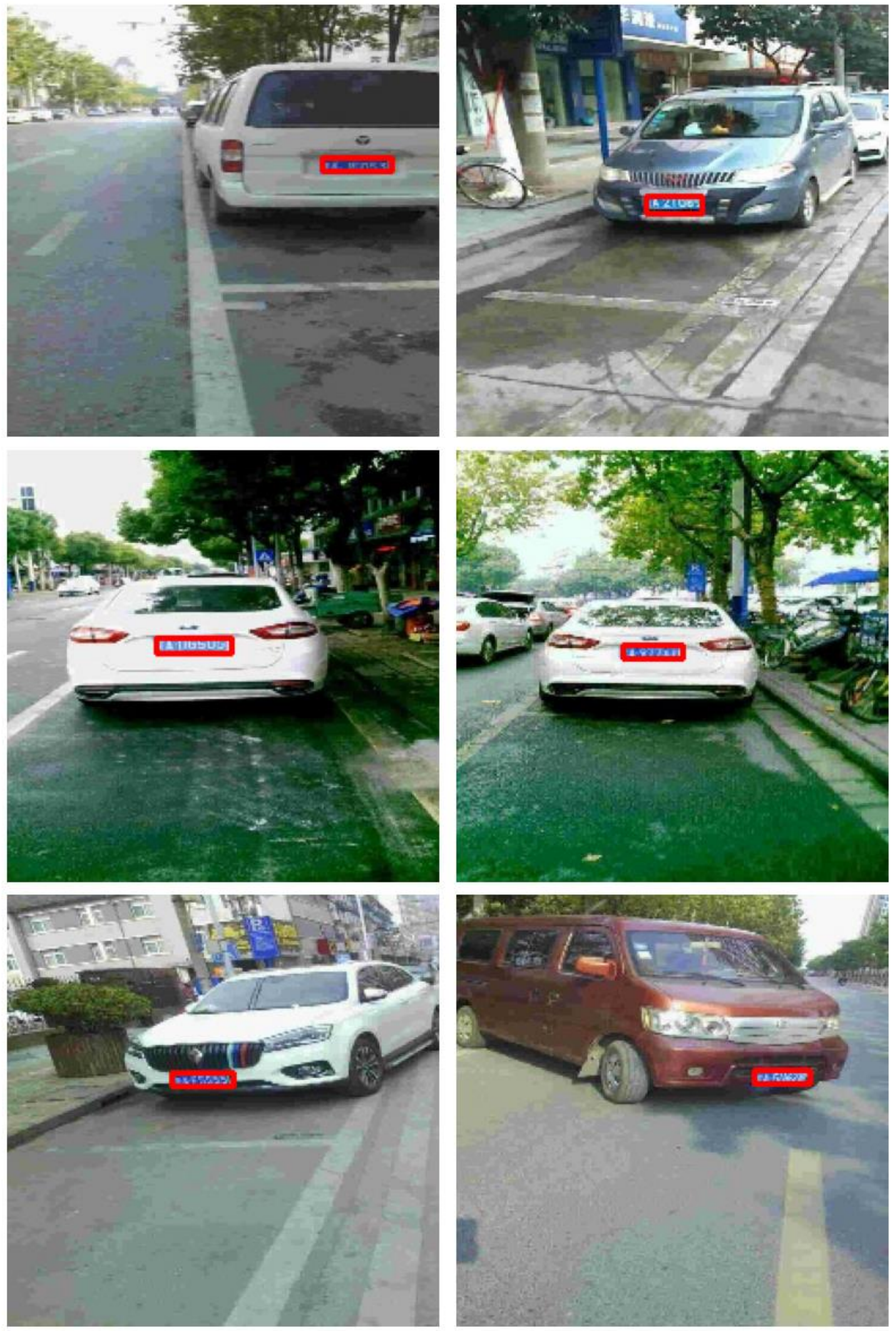
Lingbing Tao [1], Shunhe Hong [1], Yongxing Lin [1,2], Yangbing Chen [1], Pingan He [3] and Zhixin Tie [1,2,*]

1 School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; lb_tao@zstu.edu.cn (L.T.); 202020502017@mails.zstu.edu.cn (S.H.); eluian@zstu.edu.cn (Y.L.); cyb@zstu.edu.cn (Y.C.)
2 Keyi College, Zhejiang Sci-Tech University, Shaoxing 312369, China
3 School of Science, Zhejiang Sci-Tech University, Hangzhou 310018, China; pinganhe@zstu.edu.cn
* Correspondence: tiezx@zstu.edu.cn

Dataset

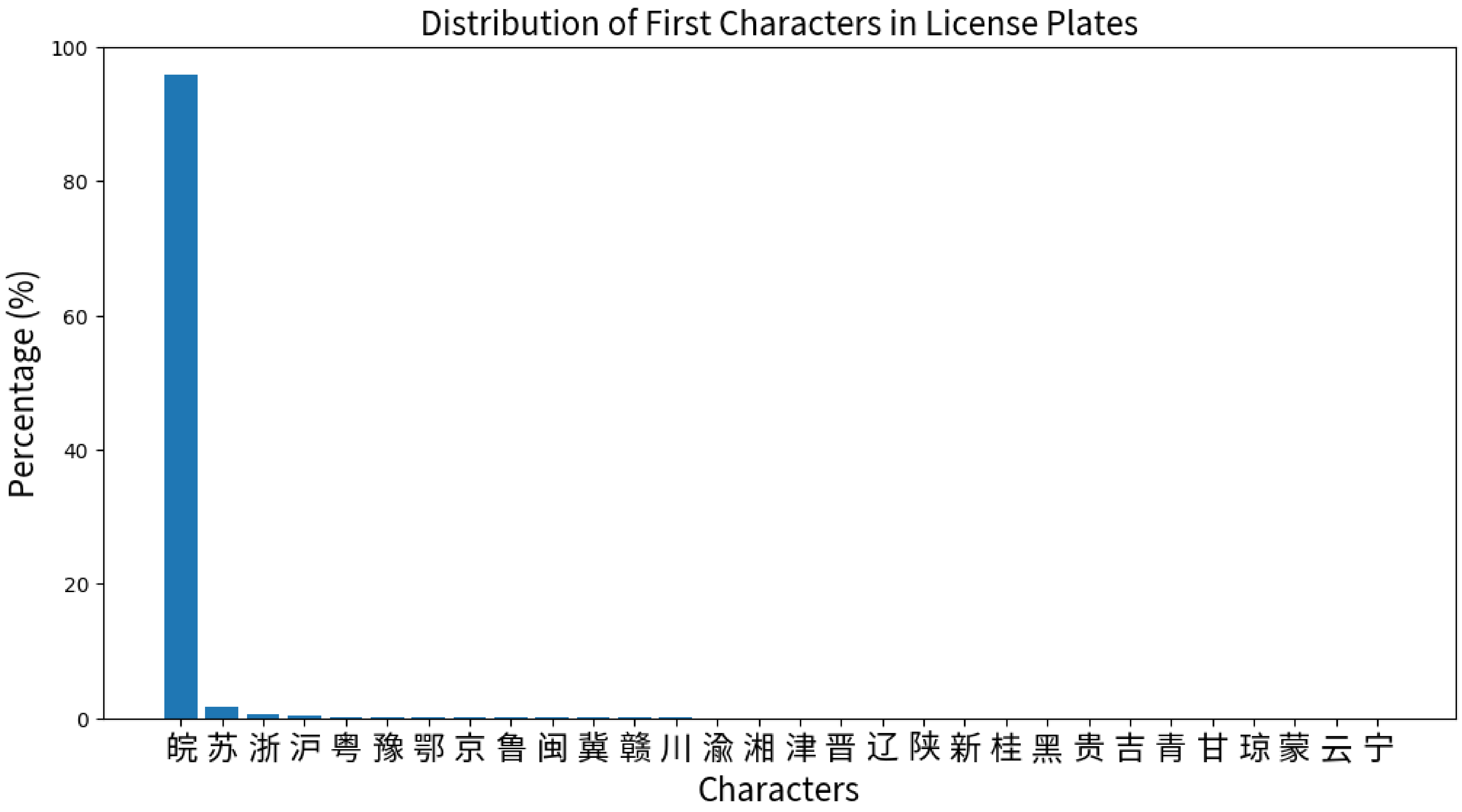
CCPD (Chinese City Parking Dataset) is a large open-source dataset of **Chinese license plates**.

Sub-Dataset	Description
CCPD-Base	The only common feature of these photos is the inclusion of a LP.
CCPD-DB	Dark, uneven, or extremely bright illumination in the LP area.
CCPD-FN	Captured from varying distances
CCPD-Rotate	Large horizontal tilt and vertical tilt.
CCPD-Tilt	Extreme horizontal and vertical tilt.
CCPD-Blur	Blurry images, mostly due to hand jitter during capture.
CCPD-Weather	Images taken in adverse weather: rain, snow, or fog.
CCPD-Challenge	The most difficult and complex images for license plate detection and recognition (LPDR).



Dataset

The distribution of **Chinese characters** in the CCPD dataset is highly **imbalanced**, with the character 皖 (Anhui) appearing in approximately 95% of the samples.



Data Augmentation

To improve generalization and to prevent overfitting, we employed **data augmentation**:

- Affine transformations:
  - (rotation, shear, translations, scale)
- Blur
- Color dithering
- Change in:
  - (contrast, saturation, brighteness)
- Image quality degradation
- Other visual pertubations

Data Augmentation

Detection (YOLOv5)

Recognition

Plate: 皖AJF053

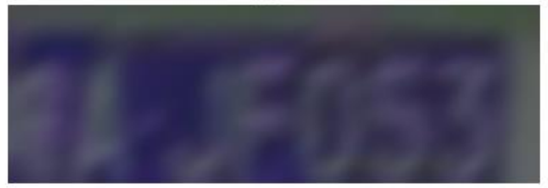


Plate: 皖A0X654

Plate: 苏DAU501

Plate: 皖ASG718

Plate: 皖AH570P

Plate: 皖A341A7

Plate: 皖AH5435

Plate: 皖A37868

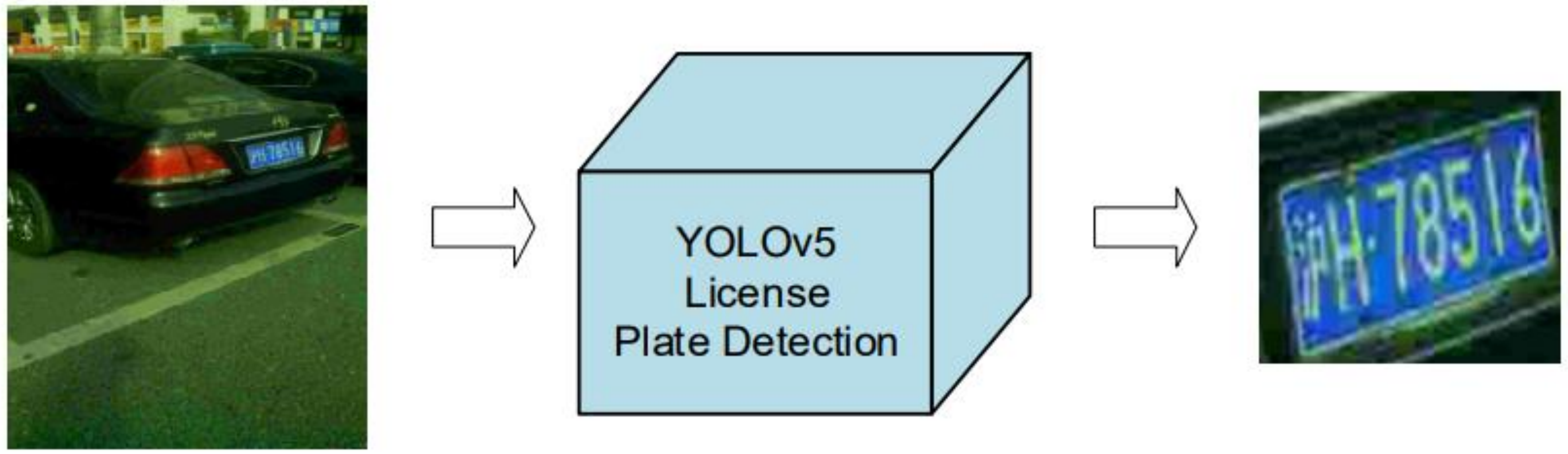
Plate: 皖A977X1

Plate: 皖AB7C66

YOLOv5

For the detection module, we used **YOLOv5s** (7.2M parameters):

- **Input:** image
- **Output:** a list of **bounding boxes** coordinates $(c_x, c_y, width, height)$ **normalized** (from 0 to 1) with a confidence score and the class label



An example of YOLOv5 output

Id	c_x	c_y	width	height	confidence
0	0.47326	0.36937	0.38143	0.06616	0.83943
0	0.47204	0.36831	0.37532	0.06425	0.87525
0	0.47276	0.36876	0.37554	0.06505	0.87570

YOLOv5

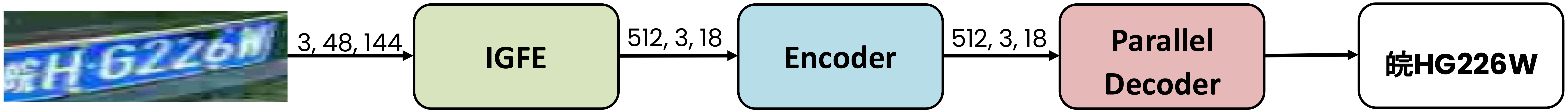
plate 0.9
plate 0.3
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.9
plate 0.8
JAC

PDLPR: overall structure

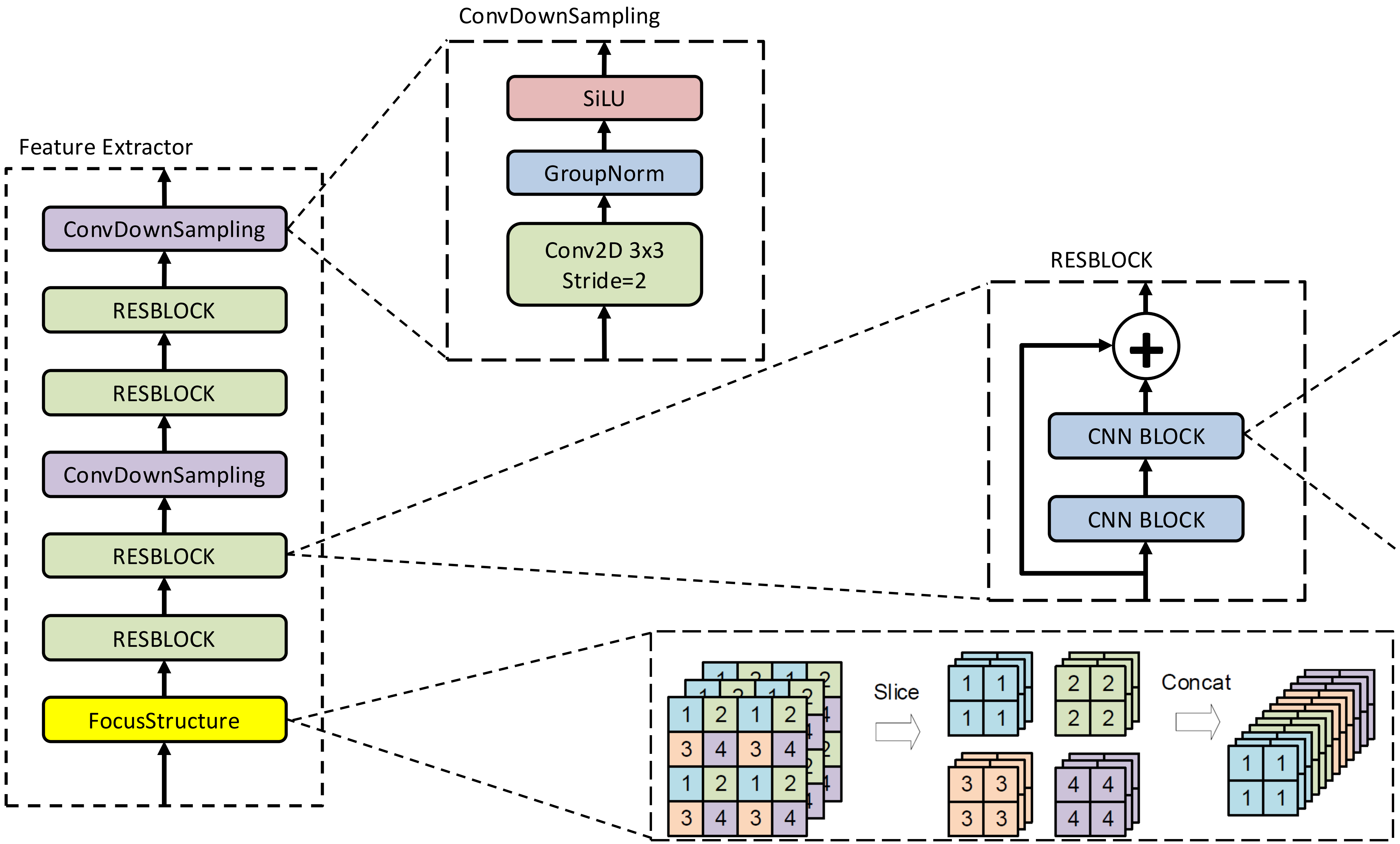
1. **Improved Global Feature Extractor (IGFE)**
   - Extracts features and converts them into a feature vector

2. **Encoder Module**
   - Encode the vector using MHA to produce a feature vector.

3. **Decoder Module**
   - Utilizes **Multi-Head Attention** to decode the encoder's output feature vector.
   - Predicts the final license plate sequence



PDLPR: IGFE

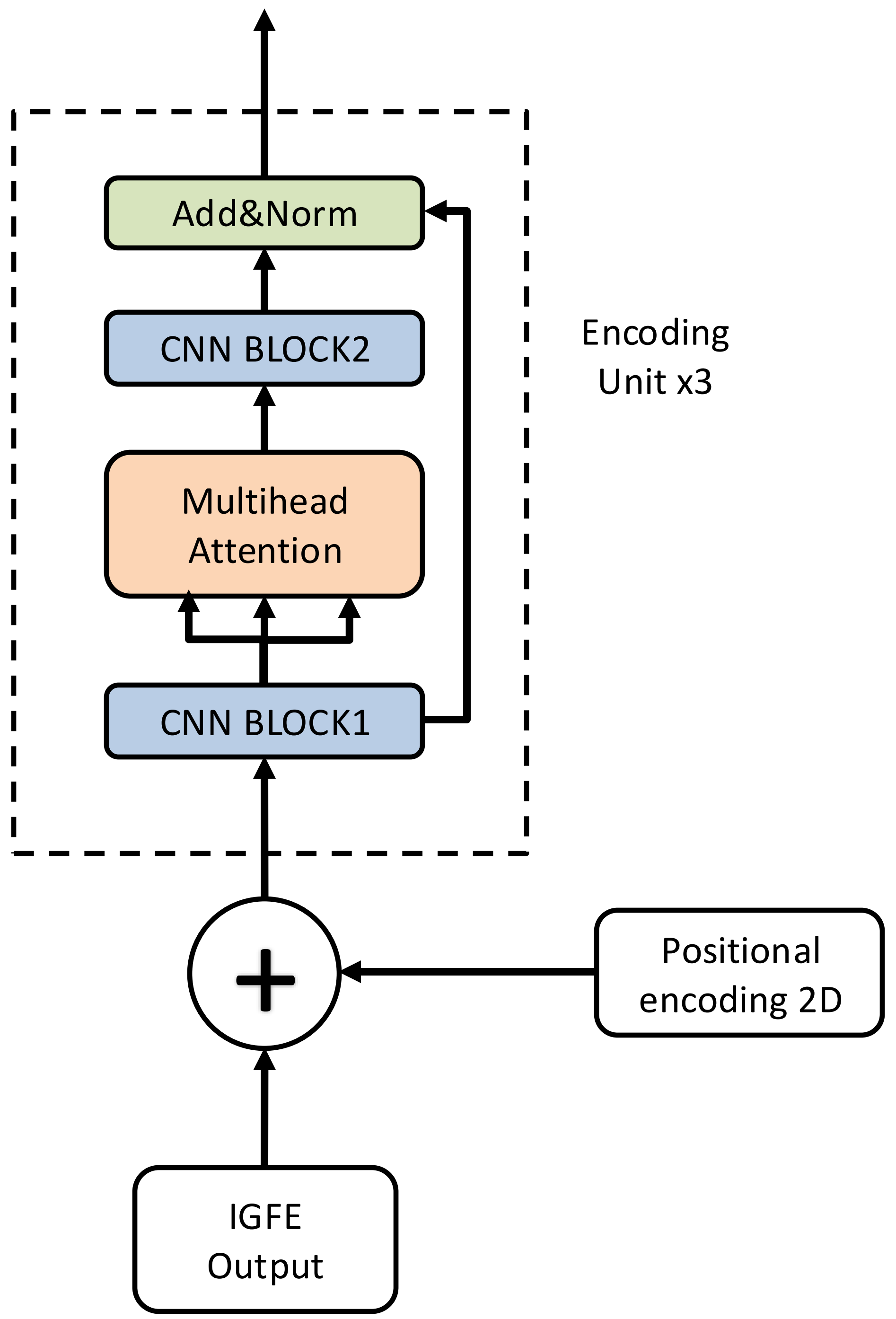

ConvDownSampling
SiLU
GroupNorm
Conv2D 3x3
Stride=2
Feature Extractor
ConvDownSampling
RESBLOCK
RESBLOCK
ConvDownSampling
RESBLOCK
RESBLOCK
FocusStructure
RESBLOCK
CNN BLOCK
CNN BLOCK
CNN BLOCK
SiLU
GroupNorm
Conv2D 3x3
Stride=1
Slice
Concat

PDLPR: Encoder

Input : [512, 3, 18]

- **Positional encoding 2D**
- **3x Encoder Unit:**
  - CNN BLOCK1
  - MHA
  - CNN BLOCK2
  - Add&Norm

Output : [512, 3, 18]

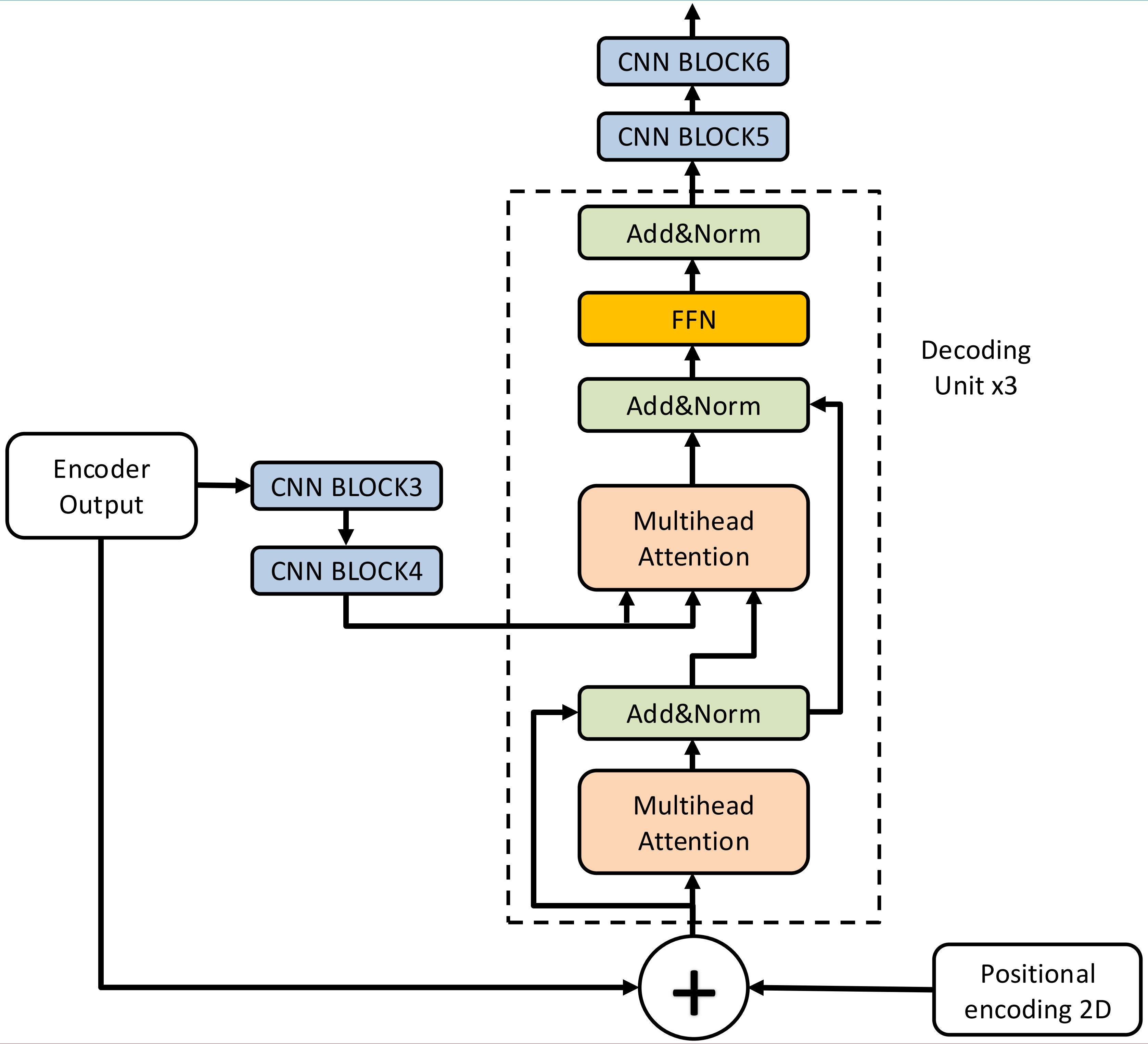


PDLPR: Parallel decoder

Input : [512, 3, 18]

- **Positional Encoding 2D**
- **3x Decoding unit:**
  - No masked MHA (not autoregressive)
  - Add&Norm
  - MH Cross-Attention
  - Add&Norm
  - FFN
  - Add&Norm
- **CNN BLOCK5**
- **CNN BLOCK6**

Output: [512, 18, 68]

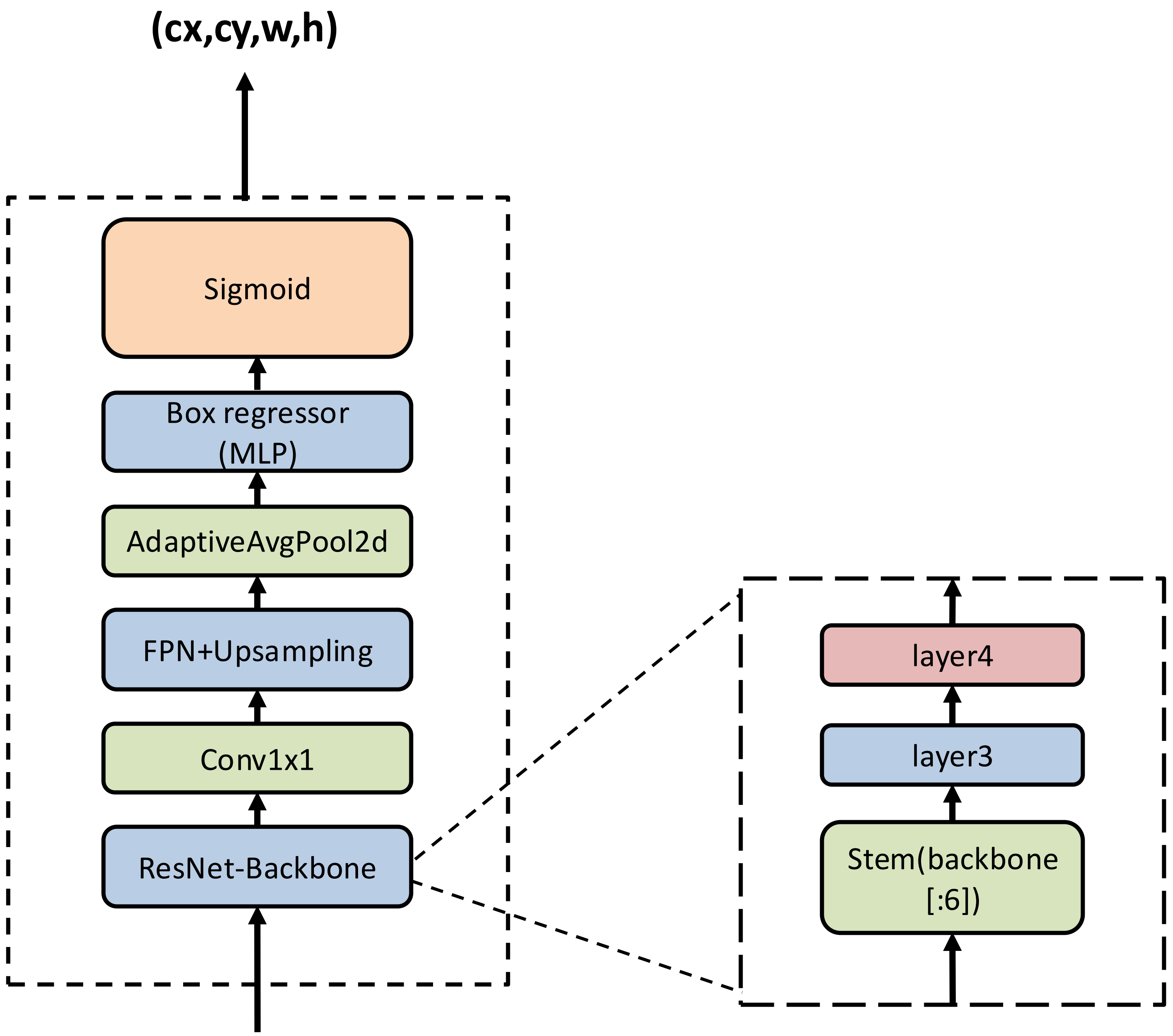


Baseline: Detection

Architecture Overview

- **Backbone:**
  - ResNet-18 (pre-trained on ImageNet)
- **Feature Fusion:**
  - Feature Pyramid Network **(FPN)**
  - Combines features from layer3 and layer4
  - Uses 1×1 conv + upsampling + element-wise addition
- **Head:**
  - Global Average Pooling (**GAP**)
  - **MLP** with one hidden layer (dropout+ReLU)
  - **Sigmoid**

Output: (cx,cy,w,h)∈[0,1]



Baseline: Recognition

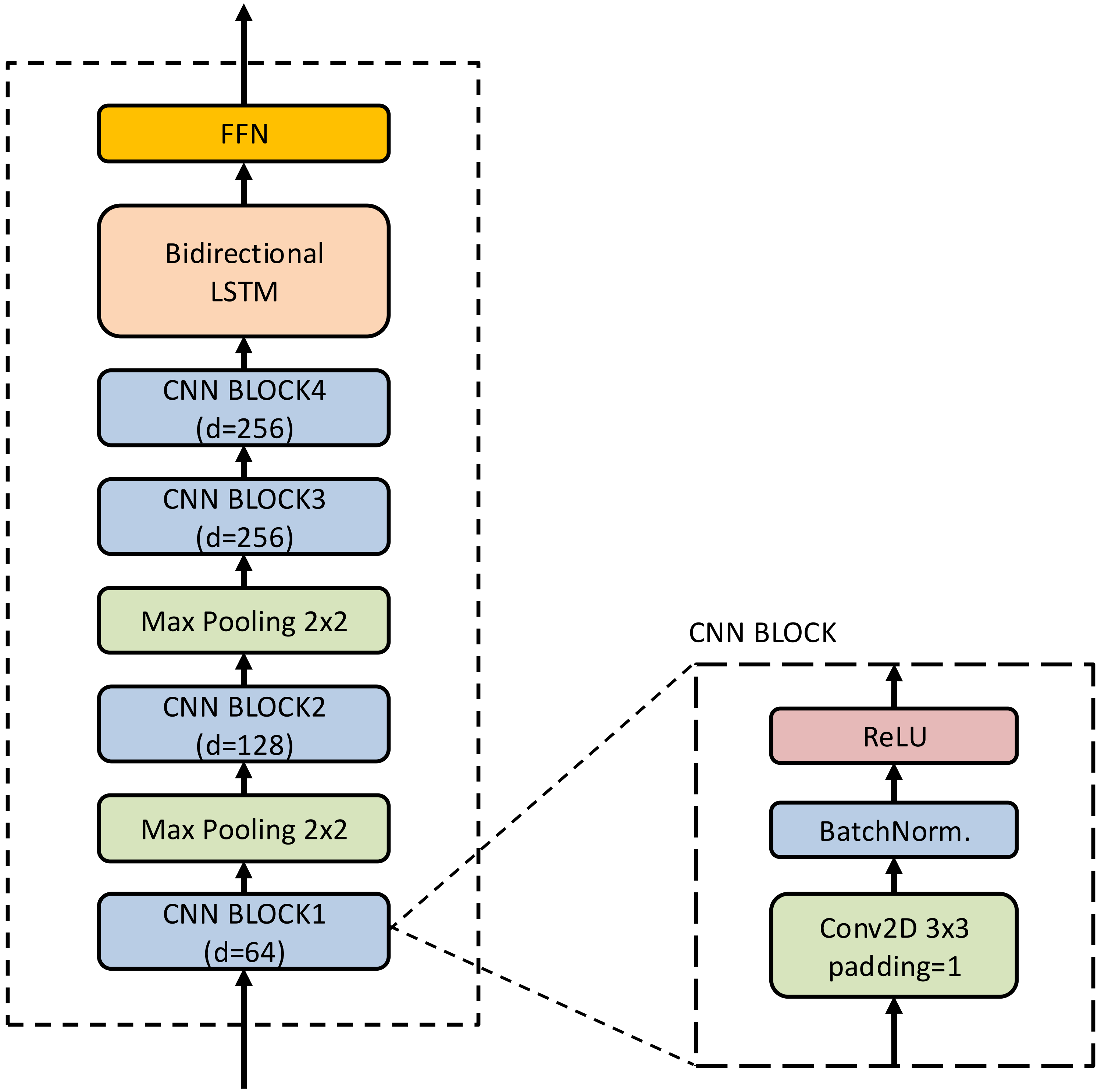
Input: image [3, 48, 144]

1. **Feature extractor:**
   - **Convolutional backbone** to extract spatial features:
     - with **increasing channel** dimensions

2. **Sequence Modeling:**
   - 2-layer **Bidirectional LSTM**
     - Input sequence length: 36 (from image width)
     - Feature vector per step: 256 × 12 = 3072
   - Output: **7×1024** (per image)

3. **Head:**
- Linear projection of 1024 → 68 class logits

Output: [7, 68]



Training Phase

YOLOv5 + PDLPR

YOLOv5

- **Training samples:** 40k
- **Epoch:** 10+40
- **Learning rate:** variable
- **Criterion:** BB Loss + OBJ Loss
- **Optimizer:** AdamW

PDLPR

- **Training samples:** 49k
- **Epoch:** 105
- **Learning rate:** 10^{-4}
- **Criterion:** CTC Loss
- **Optimizer:** Adam

Baseline

ResNET

- **Training samples**: 49k
- **Epoch:** 20
- **Learning rate:** 10^{-4} (with scheduler)
- **Criterion:** CIoU Loss
- **Optimizer:** AdamW

CNN+LSTM

- **Training samples:** 49k
- **Epoch:** 50
- **Learning rate:** 10^{-3}
- **Criterion**: CE Loss
- **Optimizer**: Adam

Evaluation metrics

- **Intersection over Union (IoU)** : the amount of overlapping area between two bounding boxes over the total area covered by both boxes

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

- **Sequence Accuracy** : measures the percentage of license plates where the entire predicted sequence exactly matches the ground truth

$$Sequence\ Accuracy = \frac{number\ of\ correctly\ predicted\ license\ plate}{total\ number\ of\ license\ plate}$$

Results: Detection

*Each subset has 1k samples

	$\mathbf{IoU \geq 0.7}$									
Method	**Overall**	**Base**	**Blur**	**Challenge**	**DB**	**FN**	**Rotate**	**Tilt**	**Weather**	**FPS**
ResNet-box	86.24	98.2	83.1	86.30	74.70	74.00	93.40	81.70	98.50	<u>94.7</u>
YOLOv5	<u>96.2</u>	<u>99.6</u>	<u>94.9</u>	<u>95.6</u>	<u>92.6</u>	<u>93.0</u>	<u>98.6</u>	<u>95.6</u>	<u>99.7</u>	94.3

YOLOv5:

- Achieves significantly **higher overall accuracy**
- Shows **strong generalization** across all subsets

ResNet-box:

- Slightly **faster**
- Struggles more in challenging scenarios

Results: Recognition

*Each subset has 1k samples

	Sequence Accuracy									
Method	**Overall**	**Base**	**Blur**	**Challenge**	**DB**	**FN**	**Rotate**	**Tilt**	**Weather**	**FPS**
CNN+LSTM	89.9	99.6	84.3	85.6	83.4	86.4	93.9	<u>87.0</u>	<u>99.0</u>	<u>552.65</u>
PDLPR	<u>91.85</u>	<u>99.8</u>	<u>90.6</u>	<u>89.6</u>	<u>84.7</u>	<u>90.5</u>	<u>93.9</u>	86.9	98.8	311.49

PDLPR:

- Achieves **higher overall sequence accuracy**
- Stronger performance on challenging subsets

CNN+LSTM:

- Much **higher speed**
- Sightly **less accurate**

Results: Detection + Recognition

*Each subset has 1k samples

	Sequence accuracy (IoU>0.6)									
Method	**Overall**	**Base**	**Blur**	**Challenge**	**DB**	**FN**	**Rotate**	**Tilt**	**Weather**	**FPS**
Baseline	86.86	99.7	77.7	82.6	76.4	80.0	93.3	86.5	98.7	85.96
YOLOv5-PDLPR	89.49	99.7	81.9	88.9	74.6	88.7	94.0	89.4	98.7	66.17

YOLOv5-PDLPR:

- Achieves **higher overall accuracy**
- **Outperforms** the baseline on challenging subsets

Baseline (ResNet18 + BiLSTM)

- Remains competitive, especially on CCPD-Base and CCPD-DB
- **Higher** inference speed

Results

True Bounding Box

Predicted Bounding Box

Conclusions

- YOLOv5 + PDLPR achieves **higher accuracy** than baseline (89.5% vs 86.9%)
  - Biggest gains in **difficult conditions**
- Baseline is **faster** (86 vs 66 FPS).
- Trade-off between accuracy (YOLOv5+PDLPR) and speed (baseline).

- **Future improvements**:
  - Training with more data (including synthetic samples)
  - Increasing training epochs
  - Using balanced data
  - Latest version of YOLO