Car Plate Recognition and Reconstruction with Deep Learning

Alessandro Tagliaferri – Federico Navarra
Computer Vision 2024-2025
Sapienza University of Rome

Table of contents

Task overview

The goal is to develop a complete pipeline for **Car Plate Detection and Recognition** using Deep Learning.

Objectives

- Implement a simple **baseline** for both detection and recognition.
- Reproduce the **proposed pipeline:**
  → **YOLOv5** for detection
  → **PDLPR** for recognition
- **Train, evaluate, and compare** models on a real-world dataset (CCPD).
- Investigate **alternative recognition models** beyond those strictly required.

Two parts

Detection

Predict plate box on the image and crop it

Recognition

Recognize plate number from cropped image

Dataset

CCPD 2019 dataset

Key Features:

- Over **250,000 images** of cars with visible license plates.
- Captured in **unconstrained scenarios**: varying lighting, angles, distances, occlusions, and weather conditions.
- Each image comes with **annotated bounding boxes** and **plate text labels**.
- Plate numbers include both **Chinese characters** and **alphanumeric digits**.

CCPD 2019

State of the art

Two steps approach

Most recent works adopt a **two-stage pipeline**:

1. Detection: Locate the license plate in the input image.

2. Recognition: Decode the characters from the cropped plate.

- Detection is typically solved with **object detection models** (YOLO, Faster R-CNN, SSD...).
- Recognition relies on **OCR Engines** or **Deep Learning OCR Models**

S.O.T.A

Method	Overall Accuracy	Base (100k)	DB (20k)	FN (20k)	Rotate (10k)	Tilt (10k)	Weather (10k)	Challenge (10k)	Speed (FPS)
YOLOv5-PDLPR	99.4	99.9	99.5	99.5	99.5	99.3	99.4	94.1	159.8
YOLOv5	96.7	97.2	97.7	92.9	98.9	98.9	99.0	90.6	218
RPNET	95.5	98.5	96.9	94.3	90.8	92.5	87.9	85.1	61

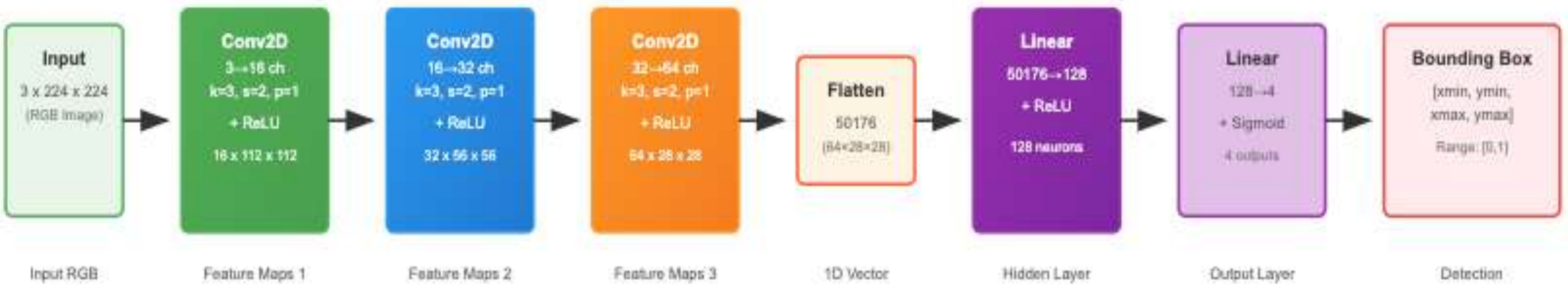
Proposed methods

From Baseline to Final Model

- We implemented a **simple baseline** for both detection and recognition
- We reproduced and trained the **YOLOv5 + PDLPR model** as proposed in the reference paper

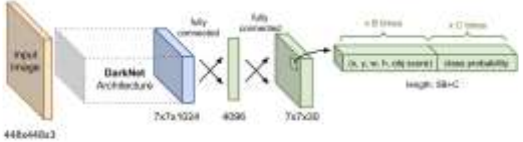
- Explored **alternative models** like Holistic CNN, LPRNET and CRNN for recognition
- Trained and evaluated all models on the **CCPD dataset**, using consistent metrics

Detection – Baseline with SimpleCNN



Detection – Yolov5s for Plate Recognition

- **One-stage** detector with real-time performance
- High detection speed with good accuracy
- Uses CSPDarknet **backbone** and **PANet** for feature fusion
- Outputs **bounding boxes** with confidence scores
- Fine-tuned on **CCPD** with pretrained COCO weights
- Cropped plates from YOLO detections with IoU>0.6 used for **recognition**

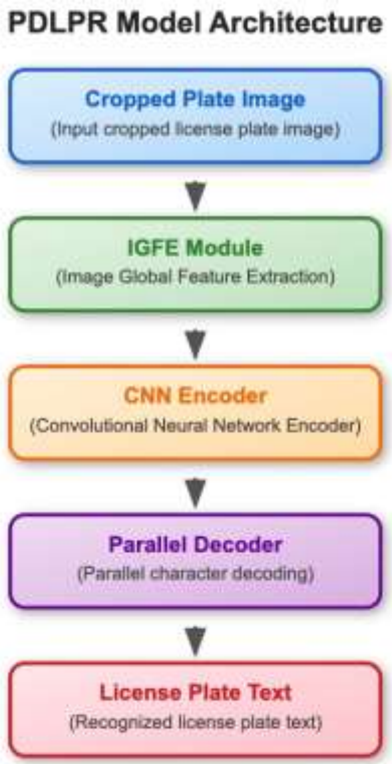


Detection with Yolo

Recogniction - Holistic CNN vs CRNN vs LPRNet

Model	Model Architecture	Output Type	Loss Function	Robust-ness	Strengths	Weaknesses
Holistic CNN	Fully Conv. CNN	Fixed-length classification vector	Cross Entropy	Low	Fast, simple, lightweight	Sensitive to distortions and variations
CRNN	CNN + BiLSTM + CTC	Variable-length sequence	CTC	Medium	handles variable-length outputs effectively	Slower, less efficient
LPRNet	Lightweight CNN + sequence head	Variable-length logits	CTC	High	High inference speed with good accuracy	Slightly lower accuracy than transformer models
PDLPR	CNN + Transformer decoder	Character sequence	Cross Entropy	Very High	Robust, state-of-the-art performance	Complex architecture, long training

PDLPR - License Plate Character Recognition



2 small CNN branches - Extracts geometry-aware features

CNN - Encodes visual/textual features

7 classification heads - Predicts all characters in parallel

Our setup

DATASET

- 100k images sampled from CCPD-base
- Train/Eval 80/20
- Used cropped plates from YOLOv5 with IoU > 0.6 for recognition training

DETECTION

- Pretrained YOLOv5s
- Fine-tuned on our subset

RECOGNITION

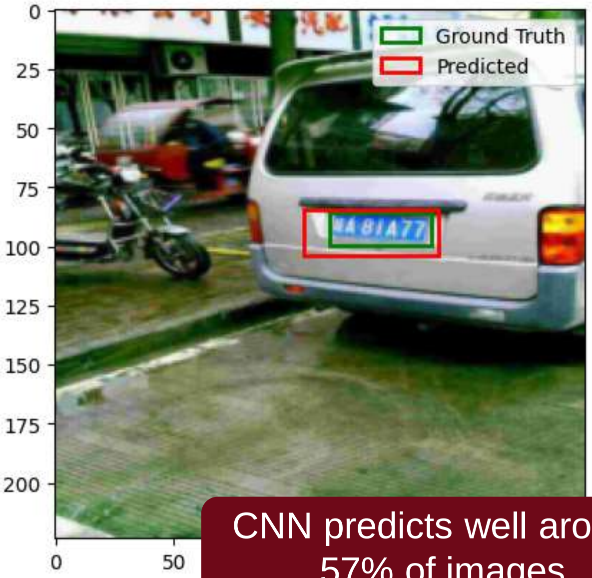
- Input: 48×144 cropped plates
- Augmentations: color jitter, rotation, erasing

Results

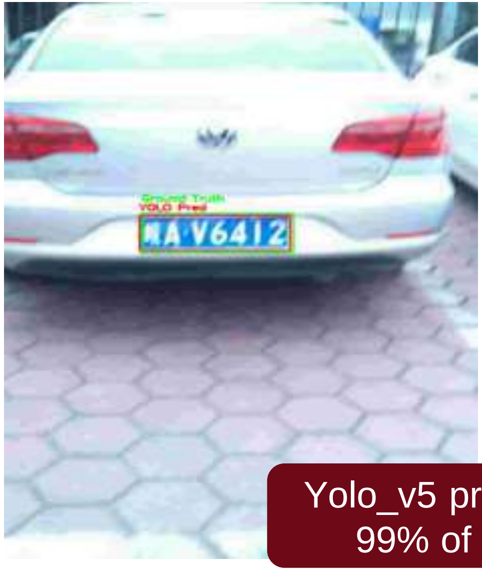
Performance results and output plots

- **Detection performance:** YOLOv5 outperforms CNN baseline
- **Recognition accuracy:** PDLPR shows best results among tested models
- CRNN and Holistic CNN show competitive performance but lower generalization
- **Visual samples:** detection boxes + predicted license plates
- **Error cases:** mainly due to occlusion, blur or incorrect crops

Detection – SimpleCNN vs Yolov5s



CNN predicts well around 57% of images



Yolo_v5 predicts well 99% of images

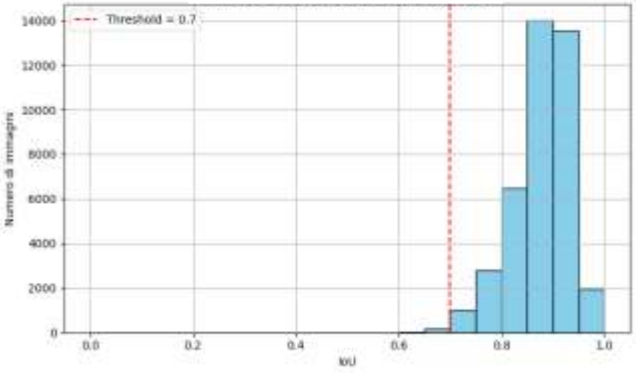
The bounding box is considered to be correct if and only if its IoU with the ground truth bounding box is more than 70%

YOLOv5s metrics

Category	Total Images	Images with IoU > 0.7	Percentage (%)
blur	20611	16643	80.8
challenge	50003	45586	91.2
db	10132	6193	61.1
fn	20967	15893	75.8
rotate	10053	9450	94.0
tilt	30216	24525	81.2
weather	9999	9825	98.3
base	199996	198764	99.4

Percentage of Images with IoU ≥ 70% across Datasets

Distribution of YOLO Prediction IoUs on base dataset



Recognition baseline models performance

Model	Sequence Accuracy (%)	Character Accuracy (%)
Holistic CNN	24.2	82.9
CRNN	98.3	99.5
LPRNet	99.1	99.8

Performance comparison of recognition models on the base test-set

- All models trained on images from **base dataset**.
- High accuracy on training-like data, **low generalization** on unseen styles.
- **Holistic CNN** struggles with full sequence prediction.

These baselines show promising results on the base set, but fail to generalize effectively beyond it

Possible improvements could be train on a larger amount of images and enhance data augmentation

PDLPR metrics on different test datasets

Dataset	Sequence Accuracy (%)	Character Accuracy (%)
base	98.7	99.7
blur	21.7	71.6
db	33.8	76.2
fn	62.1	89.6
rotate	65.3	91.7
weather	95.6	99.1
challenge	51.2	85.6
tilt	68.4	91.7

PDLPR model - Successes and Failures

Visual comparison of correctly and incorrectly predicted license plates

Conclusions

What we learned and how we can improve

- Summary of **key findings** from our experiments
- Insights on model **performance** and **comparison**
- Ideas for improving the system in **future work**

Summary of key findings

- **PDLPR** delivered the best recognition results due to its dedicated parallel decoding and feature extraction.
- **YOLOv5** significantly outperformed the baseline CNN in detection accuracy and generalization.
- **LPRNet** and **CRNN** were useful for comparison, but struggled with complex license plates.
- End-to-end performance was strongly influenced by **detection quality** and **input clarity.**

Future Work

Increase number of real training images

Collect and include more diverse real-world plate images to improve model generalization

Add robust augmentations

Add robust augmentations (blur, occlusion, brightness) to enhance generalization

Experiment with end-to-end joint training

Integrate both tasks in a unified end-to-end architecture for improved accuracy

References

- ***Papers:***

1. Tao, L., Hong, S., Lin, Y., Chen, Y., He, P. and Tie, Z. (2024). A Real-Time License Plate Detection and Recognition Model in Unconstrained Scenarios. Sensors, 24(9), 2791

2. Xu, Z.; Yang, W.; Meng, A.; Lu, N.; Huang, H.; Ying, C.; Huang, L. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.

3. R. K. Prajapati, Y. Bhardwaj, R. K. Jain and D. Kamal Kant Hiran, "A Review Paper on Automatic Number Plate Recognition using Machine Learning : An In-Depth Analysis of Machine Learning Techniques in Automatic Number Plate Recognition: Opportunities and Limitations," 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 527-532

- ***Dataset:***

  https://github.com/detectRecog/CCPD

- ***Slides:***

  https://github.com/pietro-nardelli/sapienza-ppt-template

Thank you for the attention!