

Robust End-to-End Automatic License Plate Recognition (ALPR)

Farhan Tanvir Utshaw

Missouri State University

December 11, 2025

Motivation

- ALPR is widely used in security, parking systems, and tolling automation.
- Real-world images are degraded: blur, rotation, weather, illumination.
- Goal: Build an end-to-end ALPR system robust to **non-ideal images**.
- Dataset: Full CCPD2019 (base + blur + rotate + weather + challenge).

Dataset Overview

- CCPD2019: ~250k images
- Subsets:
 - **ccpd_base** – clean images
 - **ccpd_blur** – motion blur, defocus
 - **ccpd_rotate** – rotated plates
 - **ccpd_tilt** – perspective distortion
 - **ccpd_weather** – rain, fog, low-light
 - **ccpd_challenge** – extremely hard cases
- YOLO used to extract 140k plate crops
- Final OCR training set: mixture of clean + degraded crops

Dataset Examples: Clean vs Degraded

Input to the YOLO



Figure: Base



Figure: Blur

Dataset Examples: Clean vs Degraded

Input to the YOLO

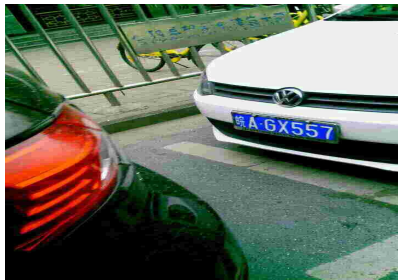


Figure: Rotation



Figure: Weather

YOLO cropped license plate — Input to the CRNN



OCR Model: CRNN Architecture

CRNN Goal: Convert the 128×32 Plate Image into a Variable-Length Text Sequence.

① CNN (Feature Extraction)

- **Role:** Extracts visual features (strokes, characters parts) from the 2D image.
- **Input:** Standardized Plate Crop (128×32 Grayscale).
- **Output:** A 3D feature map tensor that is internally transformed into a sequence of vectors by collapsing the height dimension (Map-to-Sequence step).

② Bi-Directional LSTM (Sequence Modeling)

- **Role:** Processes the feature vectors sequentially from left-to-right and right-to-left.
- **Function:** Learns the contextual dependencies and potential language patterns of the license plate sequence.
- **Output:** A probability distribution over all characters (0-9, A-Z, ...) and a special <blank> token for every time step (column).

CRNN Goal: Convert the 128×32 Plate Image into a Variable-Length Text Sequence.

① CTC (Connectionist Temporal Classification)

- **Role:** Solves the crucial problem of alignment (mapping variable-length image features to fixed-length ground truth labels).
- **Function:** Collapses the redundant, per-time-step sequence (e.g., C-C-C-<blank>-A-A) by removing repeated characters and <blank> tokens.
- **Output:** The final, clean recognized license plate text (e.g., C A).

Phase 1: Baseline Training on Clean CCPD Data

Objective: Train CRNN on high-quality, clean license plate crops to establish a strong baseline but it performed really well.

Dataset Source:

- CCPD_Base subset only (clean, centered, well-lit plates)
- Crops extracted using YOLOv8 detector
- Output folder: crops/

Dataset Size:

- ~200,000 good plate crops
- Minimal variation → ideal for OCR learning

Result:

- CRNN achieves **near-perfect accuracy** on test set
- Excellent performance on clean plates
- **Weak robustness** to blur, rotation, tilt, weather, etc.

Phase 2: Robust Training on Mixed (Good + Bad) Data

Objective: Improve robustness by including challenging plates in the OCR training set.

Bad Subsets Used:

- `ccpd_blur`, `ccpd_rotate`, `ccpd_tilt`
- `ccpd_weather`, `ccpd_challenge`

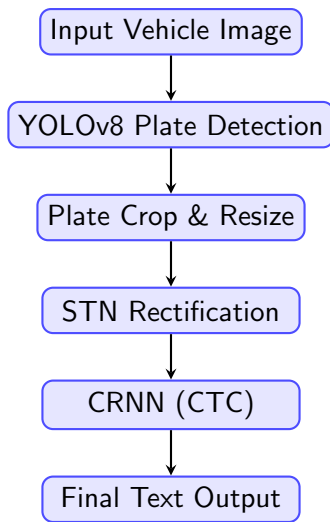
Dataset Construction:

- Crops extracted via YOLO from all “bad” subsets → **119,766 challenging crops**
- Plus a balanced sample of **20,000 clean crops**
- Combined into: `crops_mix/`

Advantages:

- CRNN exposed to diverse distortions
- Improved generalization to real-world conditions
- Significantly more robust detections

Pipeline Overview (Theory)



- ➊ **YOLOv8 License Plate Detection** Trained a detector on CCPD to localize plates in all images.
- ➋ **Plate Crop Extraction** Used the trained YOLO model to generate standardized 128×32 grayscale crops.
- ➌ **OCR Model Training (Two-Phase)**
 - Phase 1: CRNN trained on clean CCPD_Base crops.
 - Phase 2: CRNN retrained on mixed robust dataset (clean + blurred + rotated + weather-distorted plates).
- ➍ **End-to-End Inference Pipeline** YOLO → Crop → Optional STN → CRNN decoding.

License Plate Detection with YOLOv8

Goal: Accurately detect plate bounding boxes across all CCPD subsets.

Steps:

- Trained YOLOv8 on CCPD labels (train/val split).
- Achieved reliable detection even under blur, tilt, rotation, occlusion.
- Used trained model to crop $> 140,000$ plates from all CCPD subsets.

Detection Output:

- Bounding box selection (highest confidence)
- Tight crop
- Resized to 128×32 grayscale for OCR

OCR Training: Phase 2 (Robust Mixed Dataset)

Mixed Dataset Construction:

- 119,766 challenging crops from: blur / rotate / tilt / weather / challenge subsets.
- 20,000 clean CCPD_Base crops (balanced sampling).
- Total = 139,766 images in crops_mix/.

Training Result:

- Converged stable model with **Best Validation Loss: 0.0331**.
- Significantly better recognition on noisy images.
- Improved generalization to real car photos.

YOLO Plate Detection

- Trained YOLOv8n on 200k CCPD images.
- Achieved:
 - Precision: 1.00
 - Recall: 1.00
 - mAP50: 0.995
- Detector successfully used to crop plates from **all other CCPD subsets**.

OCR Model: CRNN (Baseline)

- CNN \rightarrow BiLSTM \rightarrow CTC decoding
- Inputs: 128×32 grayscale crops
- Initial training on clean CCPD crops gave:
 - Train accuracy: 100%
 - Test accuracy: 100% (on base)
- Poor robustness on blurred/tilted/rotated plates \rightarrow motivates STN + heavy augmentation.

OCR Model: STN-CRNN (Robust Version)

- Spatial Transformer Network performs:
 - geometric correction,
 - de-tilting,
 - mild de-blurring (affine stabilizing).
- Combined with CRNN improves robustness.
- Trained on:
 - 20k clean plates
 - 100k+ degraded plates (blur, rotate, weather)

Evaluation: CRNN vs. STN-CRNN

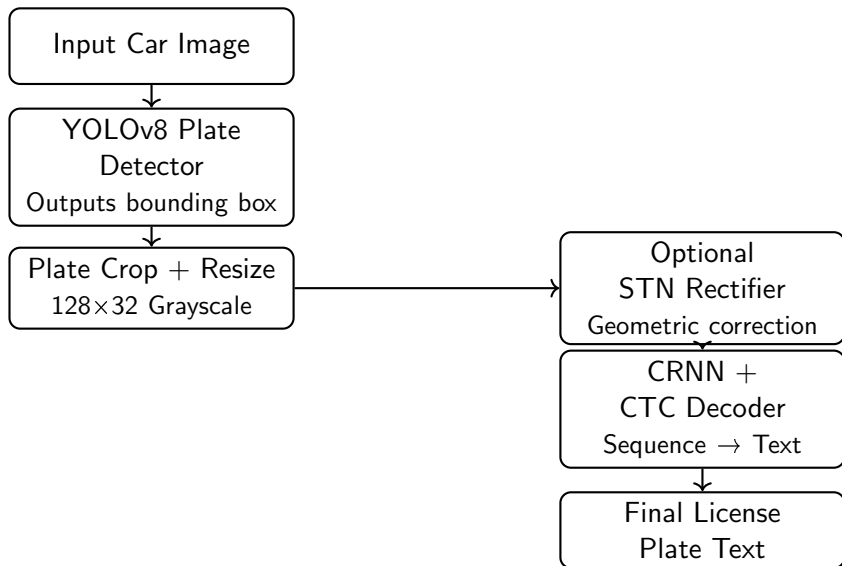
Evaluation Setup

- Held-out CCPD subset of **1000 plate crops** (not used in training).
- Goal: measure robustness on mixed-quality images (blur/tilt/weather/rotate).
- STN attempts to **geometrically rectify** plates before recognition.

Model	Correct / 1000	Accuracy
CRNN_v2 (no STN)	991 / 1000	0.9910
STN + CRNN_v2	981 / 1000	0.9810

Observation: In this dataset split, the vanilla CRNN slightly outperforms the STN variant. STN helps mainly on severe geometric distortions, but may reduce performance on already-normalized crops.

End-to-End ALPR Inference Pipeline (v2)



Why CRNN + CTC + STN?

1. Why CRNN (CNN + BiLSTM)?

- License plates are **sequences** of characters, not isolated symbols.
- CNN extracts visual stroke patterns; LSTM models left-right sequence context.
- Unlike pure CNN classifiers or Vision Transformers, CRNN scales well to variable widths and small training sets.

2. Why CTC Loss?

- Removes the need for **character-level bounding boxes** (which CCPD does not provide).
- Allows the network to learn alignment automatically.
- Simpler than attention-based decoders and more stable on noisy plates.

Why CRNN + CTC + STN?

3. Why STN (Spatial Transformer Network)?

- Many plates are skewed, tilted, blurred, or partially distorted in CCPD (blur, rotate, tilt, challenge).
- STN applies a learnable geometric correction *before* OCR.
- Avoids hand-crafted deskewing or classical image processing.

Why not other alternatives?

- **Attention decoders:** Require more data, slower inference, overkill for 6-character plates.
- **Transformer OCR models:** High compute cost; unstable on small crops like 128×32 .
- **Traditional methods (HOG + SVM):** Fail completely on blurred or rotated CCPD subsets.
- **Segmentation-based OCR:** Needs per-character boxes; not available in CCPD.

Conclusion

- Completed robust ALPR pipeline:
 - YOLOv8 detector
 - CRNN OCR
 - STN-based geometric rectification
- Robust performance even on degraded CCPD subsets.

Thank You!