

Report on Broken License Plate Detection

1. Introduction

License plates are a crucial component for vehicle identification. In many cases, plates may become broken, damaged, or tampered with, making it difficult for recognition systems to extract valid characters. This project aims to develop a program that can: 1. Detect license plates in vehicle images (front and rear). 2. Segment individual characters from the plates. 3. Determine whether any characters are broken or damaged. We implemented this using YOLOv8 (Ultralytics) for license plate detection and a custom image processing pipeline for broken character detection.

2. Tools and Technologies Used

- Python 3.10+ - Ultralytics YOLOv8 → License plate detection - Roboflow API → Dataset download and management - OpenCV → Preprocessing, character segmentation, connected components - NumPy → Image array operations - Matplotlib → Visualization - Hardware: Runs on CPU or GPU (GPU recommended for faster training).

3. Methodology

Step 1: Dataset Preparation - Dataset was downloaded from Roboflow in YOLOv8 format. - It included labeled license plate images (Front and Rear). Step 2: Training YOLOv8 Model - The YOLOv8 nano model (yolov8n.pt) was initialized. - Model was trained with 50 epochs, image size 640x640, batch size 16. - Best weights saved in runs/train/license_plate_detector/weights/best.pt Step 3: License Plate Detection - Trained YOLO model detects plates in input images. - Detected plates are cropped for further processing. Step 4: Preprocessing & Character Segmentation - Cropped plate converted to grayscale, noise reduced with bilateral filtering. - Adaptive thresholding applied to extract character shapes. - Characters segmented using contour detection. Step 5: Broken Character Detection - For each character: pixel density and connected components analyzed. - If density too low, fragmented, or irregular → flagged as broken. Step 6: Final Decision - If any character in FR or RE plates is broken → vehicle flagged. - Otherwise, plate considered intact.

4. Results

- YOLOv8 successfully detected license plates in test images. - Broken detection flagged damaged characters when pixel density was too low or fragmented. - Example output: Image: pic1.jpg
Detected Plate: FRONT Vehicle has broken characters? False

5. Assumptions

1. Each input image contains one license plate. 2. Broken detection uses heuristics (not OCR). 3. FR and RE images are paired, but system can process single images too. 4. YOLOv8 nano chosen

for speed; higher models improve accuracy.

6. Conclusion

This project demonstrated how deep learning (YOLOv8) and traditional image processing can detect broken license plate characters. Extensions: - Use OCR (Tesseract/EasyOCR) to verify recognition. - Improve broken detection with ML classifiers. - Deploy in real-time monitoring systems.

7. References

- Ultralytics YOLOv8 Docs: <https://docs.ultralytics.com> - Roboflow Docs: <https://roboflow.com> - OpenCV Docs: <https://docs.opencv.org>