Real-Time Vehicle License Plate Detection Using Computer Vision

Suraj Hudda (202211090), Sujal Chauhan (202211088)
Indian Institute of Information Technology Vadodara - ICD
Under the guidance of Dr. Jignesh Patel

***Abstract*—The Real-Time Vehicle License Plate Detection project leverages computer vision techniques to identify and extract license plates from video streams. By utilizing the Haar Cascade Classifier, this implementation provides an efficient and robust solution for real-time detection. The system captures video input, detects license plates, and extracts regions of interest (ROI) for further processing, which can be saved for later use. This project demonstrates a practical application of computer vision in traffic monitoring and law enforcement.**

***Index Terms*—Computer Vision, Haar Cascade Classifier, Image Processing, License Plate Recognition, Object Detection, OpenCV, Optical Character Recognition (OCR), Pattern Recognition, Real-time Processing, Region of Interest (ROI), Vehicle Detection, Video Processing**

I. Introduction

Vehicle license plate detection is a critical component in applications like traffic surveillance, toll collection, and automated parking systems. This project aims to develop a real-time system that detects license plates using a webcam. Implemented in Python with OpenCV, the system utilizes Haar Cascade Classifier for object detection. The project highlights the feasibility of real-time license plate detection with minimal hardware requirements.

II. Methodology

This project employs a systematic approach to detect and highlight vehicle license plates in real-time. By integrating OpenCV's video processing capabilities with the pre-trained Haar Cascade Classifier, the system ensures efficient and accurate plate detection. The extracted regions of interest can be saved for subsequent processing, making the solution highly practical for applications in traffic management and automated systems.

A. Background Knowledge

- **OpenCV:** OpenCV (Open Source Computer Vision Library) is an open-source library optimized for real-time computer vision tasks. It provides tools for image processing, video analysis, and machine learning. In this project, OpenCV is utilized for:
  - Capturing video streams via **cv2.VideoCapture**
  - Converting images to grayscale for efficient processing
  - Detecting objects using **Haar Cascade Classifiers**
  - Drawing bounding boxes and annotating detections
- **Haar Cascade Classifier:** Haar Cascades are pre-trained object detection models based on the Haar feature-based cascade classifier algorithm. Developed by Paul Viola and Michael Jones, the model scans images to detect specific features by comparing pixel intensities. The classifier is trained on thousands of positive and negative samples to recognize objects like faces, eyes, and license plates.
  In this project, the "haarcascade-russian-plate-number.xml" model is used, specifically trained for detecting license plates.

B. Workflow

The system follows a structured pipeline for real-time license plate detection and extraction. The workflow consists of five main stages, each handling specific aspects of the detection process:

1) **Video Capture:** The program initializes a webcam feed using OpenCV, with frame dimensions set for consistent processing:

```
cap = cv2.VideoCapture(0)
cap.set(3, 640)  # Set frame width
cap.set(4, 480)  # Set frame height
```

2) **Loading Pre-trained Model:** The system loads a pre-trained Haar Cascade XML file specifically designed for license plate detection:

```
plate_cascade = cv2.CascadeClassifier(
    "model/
        haarcascade_russian_plate_number.
        xml")
```

3) **Frame Processing:** Each video frame undergoes preprocessing for optimal detection performance. The system converts frames to grayscale to enhance processing efficiency:

```
img_gray = cv2.cvtColor(img, cv2.
    COLOR_BGR2GRAY)
plates = plate_cascade.detectMultiScale(
    img_gray, 1.1, 4)
```

Key parameters in the detection process include:

- *Scale Factor* (1.1): Controls image size reduction at each scale
- *Min Neighbors* (4): Defines the minimum neighborhood size for candidate detection

4) **Detection and ROI Extraction:** The system identifies license plates and extracts their regions of interest (ROIs):

```
for (x, y, w, h) in plates:
    area = w * h
    if area > min_area:
        cv2.rectangle(img, (x, y), (x + w,
             y + h),
                      (0, 255, 0), 2)
        img_roi = img[y: y + h, x: x + w]
```

A minimum area threshold filters out false positives and ensures detection accuracy.

5) **Real-time Feedback:** The system provides immediate visual feedback and allows ROI capture:

```
if cv2.waitKey(1) & 0xFF == ord('s'):
    cv2.imwrite("plates/scaned_img_" + str
        (count) +
                ".jpg", img_roi)
    cv2.rectangle(img, (0, 200), (640,
        300),
                  (0, 255, 0), cv2.FILLED)
    cv2.putText(img, "Plate Saved", (150,
        265),
                cv2.
                    FONT_HERSHEY_COMPLEX_SMALL
                    ,
                2, (0, 0, 255), 2)
    cv2.waitKey(500)
    count += 1
```

III. Results

The results from the methodology outlined above are visualized in real-time, demonstrating the detection of license plates and their extraction as regions of interest.

Fig. 1: Detection of Vehicle License Plate

IV. Applications

- **Traffic Monitoring:** Enables authorities to monitor traffic violations and vehicle tracking.
- **Toll Automation:** Facilitates automated toll collection by recognizing vehicle license plates.
- **Parking Systems:** Assists in managing and tracking vehicle entries and exits in parking lots.

V. Conclusion

The Real-Time Vehicle License Plate Detection system demonstrates an effective approach to detecting and extracting license plates in real time. By leveraging OpenCV and Haar Cascade Classifier, the project achieves accurate results with minimal computational overhead. This implementation can be extended with Optical Character Recognition (OCR) for plate number recognition and integrated into larger systems for traffic management and security.

References

[1] S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311-325, Feb. 2013.

[2] M. H. Tsai, J. S. Pan and B. Y. Liao, "Real-Time Vehicle License Plate Detection Based on Cascaded Haar-Like Features and AdaBoost Algorithm," *IEEE Access*, vol. 8, pp. 110698-110709, 2020.

[3] S. Z. Masood, G. Shu, A. Dehghan and E. G. Ortiz, "License Plate Detection and Recognition Using Deeply Learned Convolutional Neural Networks," *arXiv preprint arXiv:1703.07330*, 2017.

[4] R. Panahi and I. Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767-779, April 2017.