Team Name: IHWP
Team Members:
Vedansh Sharma (12242000)
Tanmay Kumar Shrivastava (12241870)
Deepak (12240510)

Intelligent Highway Watch Project

Semester: 2024-25M
Course: DS501/AIML Lab

Introduction

Project Overview

Our project addresses the critical challenge of automated **traffic violation detection during nighttime conditions**. By combining computer vision and machine learning techniques, we've developed a system that can effectively monitor traffic signals, detect stop line violations, and identify violating vehicles through **license plate recognition**.

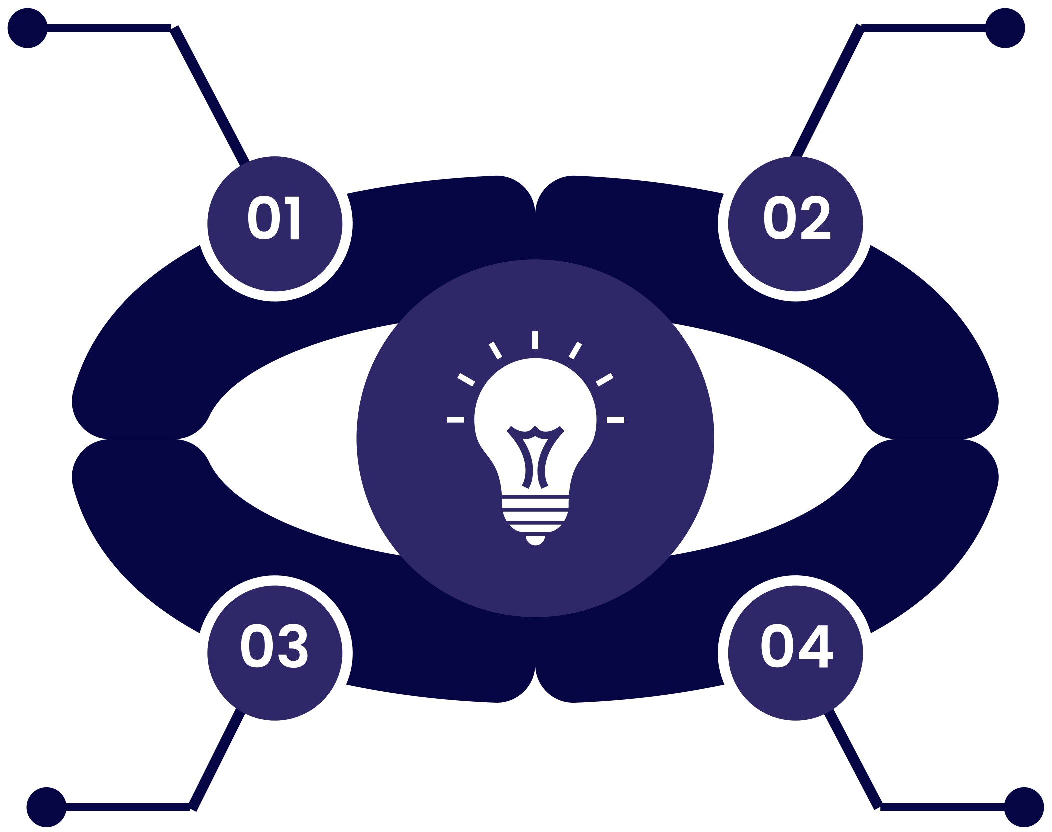
Key Innovations

- Adaptive night-vision processing
- Real-time violation detection
- Automated license plate recognition
- Integrated tracking system

Key Features of Project

Adaptive Traffic Light State Detection at Night: Utilizes advanced computer vision techniques to dynamically detect traffic light states in low-light conditions, ensuring accurate and reliable performance during nighttime.

Stop Line Detection System:
Employs robust algorithms to identify stop lines on roads, even under challenging lighting scenarios. This ensures that vehicles are penalized only when they cross clearly marked boundaries.



License Plate Extraction from Night-Time Images: Leverages state-of-the-art image processing methods to extract license plate regions from nighttime images, addressing challenges like low visibility and glare from headlights.

License Plate Processing and Violation Management
The system uses **PyTesseract OCR** for accurate recognition of license plate text from captured images. Penalized vehicle plates are displayed in real-time, providing immediate feedback and transparency for both authorities and drivers. Additionally, a **planned MySQL database** integration will enable seamless storage and retrieval of violation records, streamlining long-term data management and enforcement processes.

Dependencies & Requirements

OpenCV

- *Capturing and preprocessing video frames.*
- *Converting color spaces (e.g., BGR to HSV) for adaptive traffic light detection.*
- *Performing operations like edge detection (Canny) and contour detection for license plate extraction.*
- *Drawing masks, bounding boxes, and overlaying real-time text on video frames.*

Numpy and PyTesseract:

PyTesseract *was used for Optical Character Recognition (OCR) for converting license plate images into text.*

Numpy *was used for Efficient handling of image matrices (pixel-level operations). , processing binary masks for color and contour detection. Also used for calculating statistical properties (e.g., area of detected regions) to identify license plates or stop lines..*

MySQL Connector & PIL (Python Imaging Library

- ***MYSQL Connector*** *was used for storing records of penalized vehicles, including license plate numbers, timestamps, and violation details. Facilitating query-based retrieval for managing historical data or generating reports.*

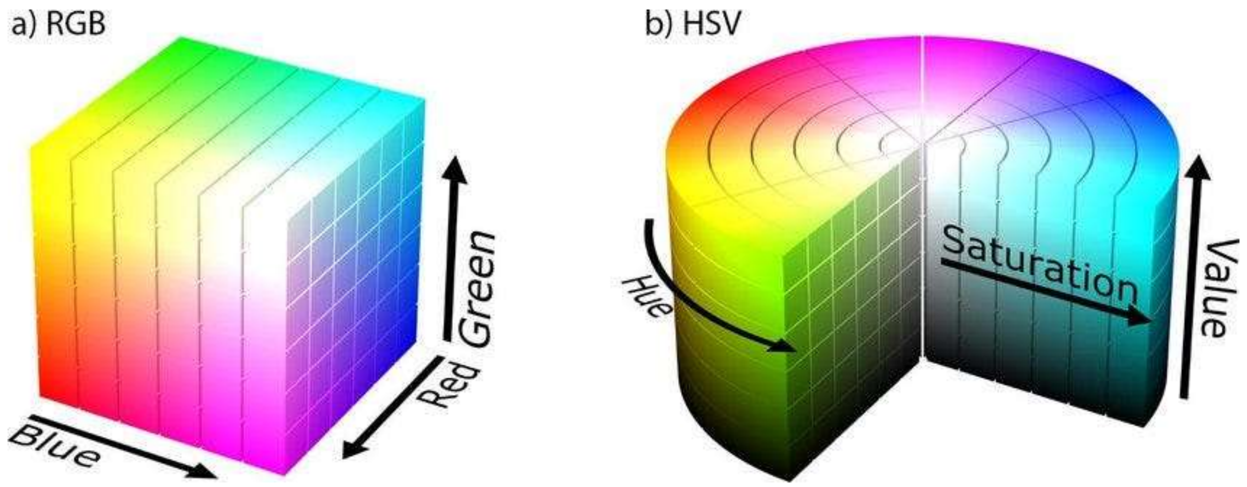
- ***PIL*** *was used for enhancing image quality (e.g., brightness and contrast adjustments) before feeding them to OCR systems. Resizing, cropping, and converting image formats to suit different processing stages.*

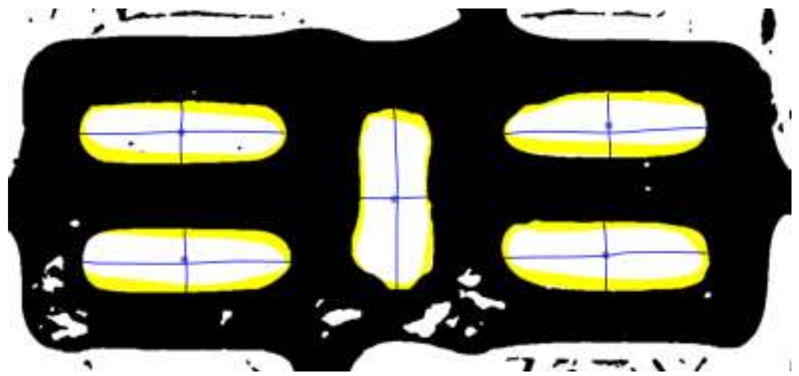
Adaptive Traffic Light State Detection at Night

Objective: Detect the state of a traffic light (Red, Yellow, Green) in video frames captured at night.

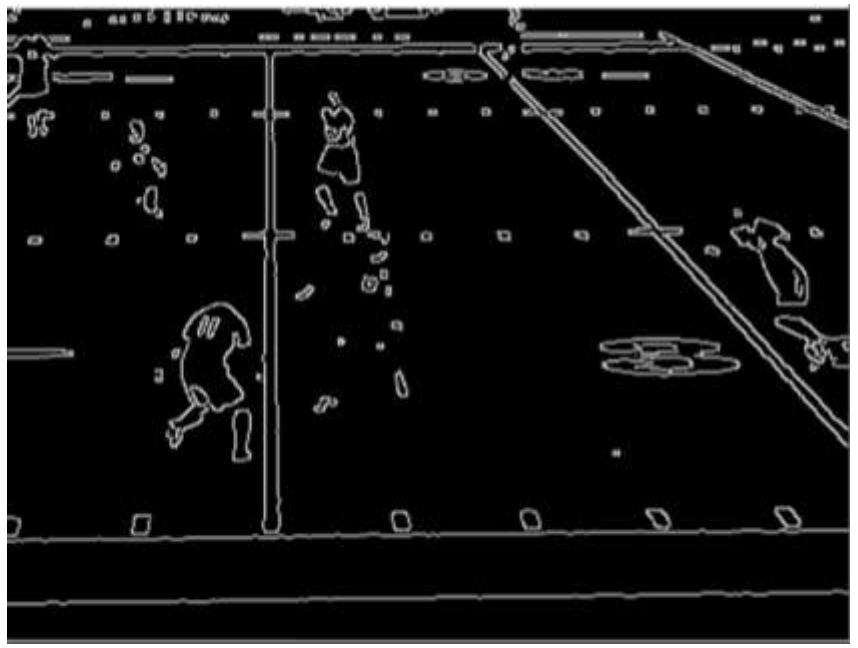
Techniques Used:

- **HSV (Hue, Saturation, Value) Color Space:** Frames are converted from BGR to HSV for better color segmentation. HSV separates color (Hue) from intensity (Value), making it ideal for detecting specific colors in low-light conditions.
- **Color Masks:** Threshold ranges are defined for red, yellow, and green lights. For example: Red: [0, 120, 70] to [10, 255, 255], Yellow: [20, 100, 100] to [30, 255, 255]. Binary masks are created to isolate regions matching these colors.
- **Mask Analysis:** Using cv2.countNonZero(mask), the system checks for significant color presence in the region of interest (ROI).
- **Real-Time Feedback:** Detected states are overlaid on the video frame using OpenCV's text overlay features.

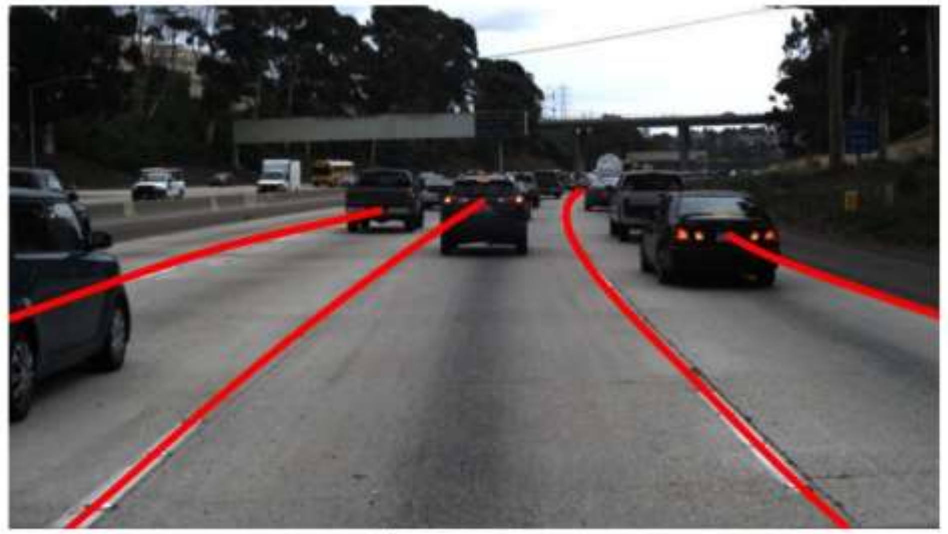




Concept of cv2.countNonZero(mask)



Concept of Hough Transform



(d) Traffic congestion

Stop Line Detection System

Objective: Detect the stop line in traffic scenes to correlate with traffic light states.

Techniques Used:

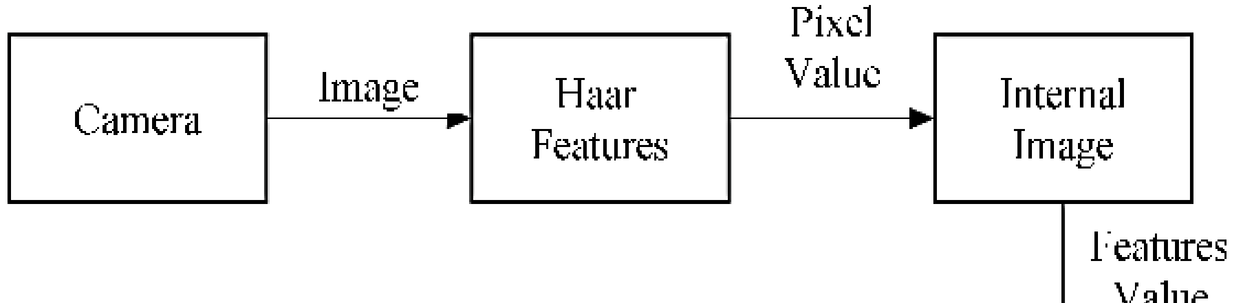
- **Area Segmentation with Line Equations** :Equations define boundaries for the area around the stop line. *Example*: *y = slope * x + intercept*. Areas outside these boundaries are masked to simplify detection
- **Edge Detection and Hough Transform:** Used Canny Edge Detection for highlighting edges, reducing the image to essential line structures.
- **Hough Transform:** Detects lines in the binary edge map. Parameters like minLineLength and maxLineGap ensure accurate and connected line detection.
- **Line Averaging:** Detected lines across frames are averaged using deques, ensuring stable and consistent detection.

License Plate Extraction from Night-Time Images

Objective: Isolate the license plate region in images for further processing.

Techniques Used:

- **Region of Interest (ROI):** Bounding boxes are applied to focus on potential license plate areas.
- **Image Preprocessing:** Used *CLAHE (Contrast Limited Adaptive Histogram Equalization)* for enhancing contrast in low-light images, making text and edges more distinguishable.
- **Edge Detection (Canny):** Detects edges in the image, helping to locate rectangular structures like license plates.
- **Contour Detection:** Filters contours based on size and aspect ratio to pinpoint license plates.
- **Haar Cascade Classifier:** The cropped *grayscale image* is then processed using a Haar cascade classifier, a traditional non-deep learning approach proficient in pattern recognition. his classifier has been pre-trained to identify license plate patterns, returning the locations of possible license plates as rectangles.




CLAHE output

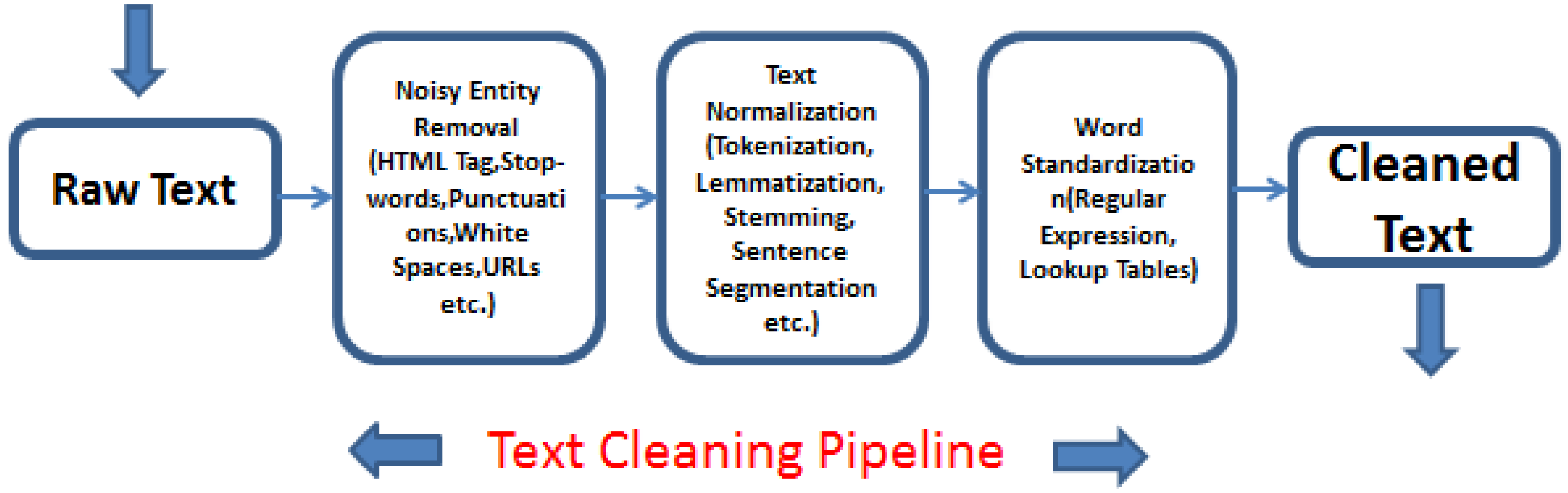

Original Image

License Plate Processing and Violation Management

Objective: Extract text from license plates and manage penalized vehicle records.

Techniques Used:

- **OCR with PyTesseract:** PyTesseract, an OCR engine, converts the license plate image into text. Preprocessing steps like thresholding, resizing, and denoising improve recognition accuracy.
- **Text Regularization:** Regex filters out unwanted characters, ensuring only alphanumeric data is retained.
- **Real-Time Display:** Penalty information is dynamically displayed on the processed video frame.
- **Database Integration:** A planned feature involves storing violations in a MySQL database for efficient record-keeping.



Key Findings

❑ **Accuracy of Adaptive Traffic Light Detection:**

- The HSV-based color detection method works effectively at night, even under low visibility, by isolating traffic light colors.
- Challenges include handling glare from other light sources, requiring further fine-tuning of thresholds.

❑ **Stop Line Detection Efficiency:**

- Edge detection and Hough Transform algorithms accurately identify stop lines, providing reliable inputs for monitoring violations.
- Detection becomes less accurate in frames with heavy shadows or overlapping vehicles.

❑ **License Plate Extraction and OCR Performance:**

- Preprocessing techniques like CLAHE and thresholding significantly improve OCR results for text extraction from license plates.
- Challenges include recognizing non-standard fonts and plates with severe reflections or damage.

❑ **Real-Time Processing Capabilities:**

- The system effectively processes video frames and displays penalized plates in real-time.
- Optimizations, such as region-based processing, reduce computational load but may miss plates on very fast-moving vehicles.

❑ **Planned Database Integration:**

- MySQL integration will enhance the system’s ability to store and manage violations, but its absence in the current stage limits historical analysis.

The project demonstrates a strong proof of concept for automated traffic monitoring at night. Future improvements should focus on handling edge cases, like occlusions, dynamic weather conditions, and complex traffic scenarios.

Thank You!