IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Automatic Dipping Of Headlight Using Yolo

[1]ANISH KISHORE R, [2]ANANDHAN B, [3]SUDALAIKANNU M
[1]STUDENT , [2]STUDENT , [3]STUDENT
[1]Department Of Computer And Communication Engineering ,
[1]Sri Sai Ram Institute Of Technology , Chennai , India

Abstract: Developing road safety and driving comfort, the development and implementation of an automatic headlight dipping system leverages the YOLO (You Only Look Once) object detection algorithm for real-time vehicle detection. Every vehicle is equipped with a front-mounted camera that captures oncoming traffic, enabling the system to automatically adjust the headlights from high beam to low beam to prevent glare for the drivers. By using advanced object detection capabilities, the system ensures quick and accurate recognition of vehicles, adapting effortless to various traffic scenarios and environmental conditions. This integration reduces the human error in headlight control and contributes to safer nighttime driving experiences.

Keywords: YOLO, Object Detection, Automatic Headlight Dipping, Vehicle Detection, Road Safety, Nighttime Driving, Smart Vehicles

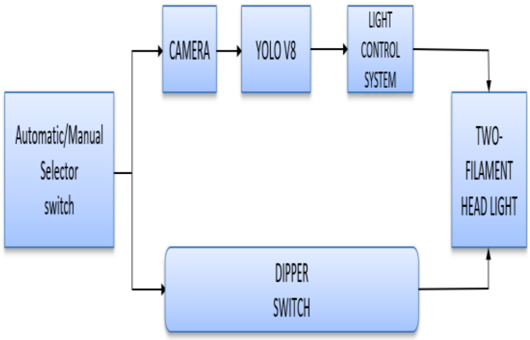
I.INTRODUCTION

As the number of vehicles on the road grows, headlight glare during nighttime driving has become a serious issue. Studies show that nighttime accidents happen more often than daytime ones, with glare from oncoming headlights being a main reason. Glare can temporarily blind drivers, making it harder to see road signs, pedestrians, or other vehicles, which increases the risk of accidents.

One major cause is human error in manually adjusting headlights. Drivers often forget to switch to low beams when facing oncoming traffic or fail to adjust their headlights in changing conditions. Because of this reason visibility is reduced.

To address this problem, a smart solution uses the YOLO (You Only Look Once) algorithm for real-time vehicle detection. YOLO is fast and accurate, capable of analyzing video from a front-mounted camera. The system detects oncoming vehicles and automatically switches to low beams to reduce glare. Once the road is clear, it returns to high beams for better visibility. This automated adjustment improves safety by preventing dazzling effects and helps drivers focus on the road without worrying about headlight operation.

Figure 1: Light control System



The system offers two modes: manual and automatic, providing drivers with flexibility based on their preferences and needs. In manual mode, the driver retains full control over the headlight adjustments, switching between high and low beams as they wanted. This mode is good for those who like direct control or want to override automation in specific context. In automatic mode, the system automatically detects oncoming vehicles and adjusts the headlights seamlessly, switching between high and low beams without driver input. This ensures proper visibility for the driver while reducing glare for others on the road. The dual-mode functionality enhances convenience and suitable for different driving styles and conditions, making the system both adaptable and user-friendly.

This technology aligns with the rise of smart vehicle technologies. Automation and advanced driver-assistance systems (ADAS) are becoming standard in modern cars, and this innovation fits perfectly into that trend. It not only improves safety but also paves the stage for more advanced features in the future. By addressing nighttime driving challenges, this technology is a step toward safer, smarter roads for everyone.

II.LITERATURE SURVEY

Research on automatic headlight control systems has been ongoing for many years, driven by the increasing need for improved road safety, especially during nighttime driving. Traditional methods for automating headlight adjustments primarily utilized light sensors and infrared (IR) technology to detect oncoming vehicles and switch between high and low beams accordingly. While these systems provided a basic level of automation, they often struggled with reliability and accuracy, particularly in complex and unpredictable driving environments.

Light sensor-based systems detect the brightness of oncoming headlights or ambient lighting and adjust the vehicle's headlights based on preset thresholds. However, these systems are prone to false readings, as they can easily confuse non-vehicular light sources—such as streetlights, traffic signals, or reflections from road signs—with oncoming vehicles. This misidentification often leads to inappropriate adjustments of the headlights, either failing to dip them when necessary or dimming them unnecessarily. Similarly, infrared systems, which rely on detecting heat signatures from oncoming vehicles, can be affected by weather conditions such as fog, rain, or snow, and even temperature variations, which may reduce their accuracy in real-world conditions.

To address these limitations, researchers began turning to more advanced technologies like computer vision and machine learning. These fields opened new possibilities for creating more sophisticated systems capable of accurate and dynamic detection of vehicles in various driving scenarios. In particular, deep learning methods, which have shown tremendous success in object detection, emerged as a promising approach to overcoming the challenges faced by traditional sensor-based systems.

Convolutional neural networks (CNNs) became the foundation for modern object detection techniques. Algorithms such as Faster R-CNN, Single Shot Multi-box Detector (SSD), and Region-based CNN (R-CNN) made significant strides in accurately detecting objects within images. However, despite their improved accuracy, these models often require significant computational power and can be slow, making them less suitable for real-time applications like automatic headlight control. The complexity of these algorithms, which involves multi-stage processes such as region proposal and object classification, adds to their latency, which is a critical factor in time-sensitive automotive applications.

The advent of the YOLO (You Only Look Once) algorithm represented a major breakthrough in the field of object detection, particularly for real-time applications. YOLO reimagines object detection as a single-stage process, bypassing the traditional approach of proposing regions of interest and instead predicting both the bounding boxes and the class probabilities for objects in a single pass. This streamlined approach allows YOLO to process images at high speeds without compromising detection accuracy, making it ideal for automotive systems where quick responses are essential for safety.

A key advantage of YOLO lies in its ability to detect objects at multiple scales and predict the location and class of objects in real time. Unlike traditional systems that rely on sensing light or heat, YOLO uses computer vision to analyze the visual input from a camera, distinguishing between vehicles and other objects with a high degree of accuracy. This capability significantly reduces the likelihood of false positives, such as confusing streetlights or reflections for oncoming vehicles. Additionally, YOLO's ability to handle multiple objects in the same frame ensures that it can detect all relevant vehicles in the driver's path, enhancing the reliability of the headlight control system.

Moreover, YOLO's performance in varying environmental conditions—such as low-light scenarios, heavy rain, and fog—makes it a robust solution for automatic headlight systems. Traditional methods often struggle in such conditions, but YOLO's deep learning architecture allows it to generalize well across different situations, making accurate detections in a wide range of lighting and weather environments. The algorithm's capacity to recognize different types of vehicles, from cars and motorcycles to larger vehicles like trucks, further enhances its applicability in real-world driving scenarios.

Compared to other object detection algorithms, YOLO excels in terms of simplicity and efficiency. It is designed to strike a balance between speed and accuracy, which is critical for applications in smart vehicle technologies, including advanced driver assistance systems (ADAS) and autonomous driving. By processing full images in one pass and eliminating the need for multi-stage detection, YOLO delivers faster results while maintaining a high level of accuracy, a combination that is highly desirable for real-time vehicle recognition.

In contrast to older sensor-based methods that struggled with environmental interferences and computationally heavy deep learning models, YOLO has emerged as a highly effective solution for real-time applications like automatic headlight control. Its superior detection speed, accuracy, and adaptability to different driving conditions make it an ideal choice for improving the reliability and functionality of headlight dipping systems, contributing to safer night time driving.

III.PROPOSED SYSTEM

The architectural design of the proposed automatic headlight dipping system comprises three main components: the Object Detection Module, the Communication Module, and the Control Module. These components work in concert to detect oncoming vehicles, process detection data, and adjust the vehicle's headlights in real-time to promote road safety. Each component and their interactions are outlined in detail below.

A. Object Detection Module (YOLO on Raspberry Pi or Laptop)

Role: This module forms the core of the headlight dipping system, utilizing the YOLO (You Only Look Once) algorithm to analyze video feeds and identify oncoming vehicles. Known for its speed and accuracy, YOLO can detect multiple objects in real-time, making it well-suited for applications requiring rapid responses.

Table 1:Features of various YOLO Algorithm

YOLO Version	Accuracy (mAP COCO @0.5:0.95)	FPS (Frames per Second)	Inference Time (ms)
YOLOv1	19-21%	45	25
YOLOv2	21-24%	40	25
YOLOv3	33%	35	30
YOLOv4	43.5%	28	30
YOLOv5	48%	60	15
YOLOv6	52%	70	10
YOLOv7	56.8%	75	12
YOLOv8	57-58%	90	10

Process:

Camera Input: A camera mounted on the vehicle (or connected to a laptop in simulation mode) captures a continuous video feed of the road.

Real-Time Processing: YOLO processes each frame of the video, identifying objects that resemble oncoming vehicles based on shape, size, and movement. This allows it to distinguish vehicles from other light sources, such as streetlamps or reflections.

Classification and Bounding Box Detection: When a vehicle is detected, YOLO places a bounding box around it and assigns a confidence score that represents the likelihood of the detected object being a vehicle. This detection process is continuous, enabling a real-time response.

Data Output and Transfer:

If an oncoming vehicle is detected within a specific range, the module sends a command to the Communication Module (ESP32) to prompt a headlight adjustment. In the absence of a vehicle, it sends a "high beam" signal, allowing the system to maintain optimal lighting conditions for the driver.

B. Communication Module (ESP32)

Role: The ESP32 acts as a bridge between the Object Detection Module and the Control Module, ensuring the fast transfer of detection data while minimizing latency. This module interprets detection signals from the Object Detection Module and relays commands to the Control Module for adjusting the headlights**.**

Data Transmission and Processing:

Signal Reception: The ESP32 receives signals from the Raspberry Pi or laptop that indicate whether an oncoming vehicle is present.Data Interpretation: Based on this signal, the ESP32 determines if the headlights should switch to high or low beam.

Stable Communication: The ESP32 supports various communication protocols, including Wi-Fi and serial communication, which ensure reliable data transmission even in mobile or changing conditions.

Data Flow and Feedback Loop:

The ESP32 transmits commands immediately following vehicle detection, ensuring quick adjustments to the headlights to maintain real-time accuracy, the ESP32 continuously relays updated data from the Object Detection Module to the Control Module, forming a feedback loop that adjusts the headlights dynamically as new vehicles enter or leave the camera's field of view.

C. Control Module (Arduino)

Role: The Control Module, managed by an Arduino microcontroller, is responsible for switching the vehicle's headlights between high and low beams based on instructions from the Communication Module.

Signal Interpretation: The Arduino receives processed data from the ESP32, instructing it to activate the correct headlight setting.

Beam Control: When the "low beam" command is received, the Arduino activates the low beam headlights to reduce glare for oncoming drivers. When no vehicle is detected, the Arduino maintains or switches to the high beam, ensuring maximum visibility for the driver.

Real-Time Responsiveness:

With its lightweight, responsive design, the Arduino allows the headlights to toggle between high and low beams with minimal delay, keeping pace with real-time object detection updates.

Reliability and Efficiency: Designed for repeated and rapid switching, the Arduino's reliability and low power consumption make it ideal for frequent headlight adjustments in real-world driving conditions.

Component Interactions:

The interactions between these modules follow an organized and efficient workflow that ensures seamless operation:

Detection and Signal Generation: The Object Detection Module identifies oncoming vehicles and sends a signal to the Communication Module.

Command Relay: The Communication Module (ESP32) processes this detection signal and relays the appropriate command to the Control Module.

Headlight Adjustment: The Control Module (Arduino) receives the command from the ESP32 and adjusts the headlights as needed, switching to low beam when an oncoming vehicle is detected and reverting to high beam once the vehicle has passed.

IV. SYSTEM ARCHITECTURE

A. Input Layer: Data Acquisition and Object Detection

Camera: A camera, positioned on the vehicle or connected to a laptop in simulation mode, continuously captures live video of the road ahead. This video feed forms the foundation of the detection process, as it provides the YOLO algorithm with the visual data necessary for identifying oncoming vehicles.

YOLO Object Detection Module: The YOLO (You Only Look Once) algorithm, known for its high-speed, single-pass object detection, is deployed on a processing device such as a Raspberry Pi in real-time vehicle implementation, or on a laptop in a simulated environment. YOLO analyzes each frame from the camera feed to detect vehicles within the driver's path. It identifies vehicles by detecting shapes, movements, and patterns, and then outputs the position and confidence level for each detected vehicle. If an oncoming vehicle is within a predefined range, YOLO generates a detection signal, prompting the system to adjust the headlights.

B. Processing and Communication Layer

ESP32 Microcontroller (Data Processing and Communication Bridge): Acting as a bridge between the YOLO detection module and the headlight control system, the ESP32 microcontroller interprets the detection signals from YOLO, determines the appropriate action, and relays the command to the control module (Arduino). This layer is essential for reducing latency and ensuring real-time response, as the ESP32's high-speed communication capabilities enable the seamless transfer of commands. When YOLO identifies an oncoming vehicle, the ESP32 interprets the detection signal and sends a "low beam" command to the Control Module. Conversely, if no vehicle is detected, it sends a "high beam" signal, allowing the system to maximize driver visibility without glare for other drivers.

Data Flow and Signal Relay: The ESP32 manages a continuous feedback loop by consistently updating the Control Module based on YOLO's detection output, creating an uninterrupted data flow from input to output. This ensures that the headlight system adapts dynamically as vehicles appear and disappear from the camera's field of view, maintaining effective and safe headlight operation.

C. Output Layer: Headlight Control System

Arduino Microcontroller (Headlight Control): The Control Module, managed by an Arduino microcontroller, is responsible for executing the headlight adjustments. Upon receiving commands from the ESP32, the Arduino activates the high or low beam settings according to the system's real-time detection status. When a "low beam" command is received, the Arduino triggers the low beam headlights, reducing glare for the oncoming driver. If no vehicle is detected, the Arduino activates or maintains the high beam to ensure maximum road illumination for the driver.

Headlight Adjustment Mechanism: The Arduino's quick response capabilities allow it to toggle smoothly between high and low beams, keeping pace with real-time updates from the Object Detection Module. Its durability and low power requirements also make it suitable for handling frequent headlight adjustments in various driving conditions, from busy roads to low-visibility environments.

D.System Workflow Overview:

Detection and Command Generation: The camera continuously captures live footage of the road ahead, which YOLO analyzes to detect oncoming vehicles. If a vehicle is detected within a specified range, YOLO generates a detection signal.

Command Relay: The ESP32 processes the detection signal, interprets it, and transmits the appropriate command (either "low beam" or "high beam") to the Control Module (Arduino).

Headlight Adjustment: The Arduino receives the command from the ESP32 and adjusts the headlights accordingly, lowering the headlights to low beam when an oncoming vehicle is detected, and returning to high beam when the vehicle has passed.

E.Feedback Loop for Continuous Adjustment:

A key feature of this architecture is its feedback loop, which enables continuous monitoring and dynamic headlight adjustment based on changing traffic conditions. The Input Layer, Processing and Communication Layer, and Output Layer interact in a closed loop, allowing the system to make rapid headlight adjustments as the road environment changes in real time. This closed-loop design allows for enhanced driver safety and comfort, ensuring that the headlights automatically adapt to prevent glare for oncoming traffic while maintaining optimal visibility for the driver.

V.EXECUTION

The execution of the proposed automatic headlight dipping system involves two primary implementations: a real-time setup for on-road functionality and a simulation-based setup for controlled testing and demonstration. Each approach leverages YOLO, ESP32, and Arduino to achieve automatic headlight adjustment, adapting to oncoming vehicles.

In the real-time implementation, designed for use on a vehicle, the system setup includes mounting a camera on the vehicle to capture a live video feed, which is processed by a Raspberry Pi running YOLO. This setup allows YOLO to continuously analyze each video frame in real-time, detecting oncoming vehicles based on defined criteria. Once YOLO identifies a vehicle within a specified range, it generates a detection signal that is relayed to the ESP32, which serves as a communication bridge. The ESP32 interprets this signal and transmits a command to the Arduino to adjust the headlights to low beam. When no vehicle is detected, the ESP32 directs the Arduino to activate or maintain the high beam, providing optimal visibility for the driver. Testing the real-time setup on the road ensures that YOLO's detection parameters and ESP32's signal interpretation work seamlessly under real-world conditions. Calibration is conducted to refine YOLO's sensitivity, detection range, and headlight response times, ensuring quick, reliable adjustments under different lighting and weather conditions.

In the simulation-based implementation, the system is designed to replicate real-time functionality in a controlled environment, ideal for iterative testing and demonstration. In this setup, a laptop running YOLO processes a video feed—either pre-recorded or from a webcam—to simulate the detection of oncoming vehicles. The ESP32, connected to the laptop, receives signals from YOLO and transmits them to the Arduino, which controls two LEDs representing high and low beam headlights. When YOLO detects a simulated vehicle, the ESP32 transmits a "low beam" command to the Arduino, activating the low-beam LED. In the absence of detected vehicles, the system defaults to the high-beam LED, demonstrating how the system would operate in real road conditions. This setup allows for a range of test scenarios, adjusting detection thresholds and LED response times to ensure consistent performance. The simulated environment provides a safe, repeatable setting to observe and validate the system's logic, with real-time transitions between high and low beam states that mirror on-road behavior.

Both implementations demonstrate the system's effectiveness in different contexts, whether on the road or in simulation. Together, they show the adaptability and robustness of the proposed headlight dipping system, where the real-time setup highlights on-road functionality, and the simulation enables thorough testing and validation.

VI.CONCLUSION

In summary, this project demonstrates an effective automatic headlight dipping system that leverages the YOLO algorithm, ESP32, and Arduino to enhance driving safety by automating headlight adjustments. Through real-time object detection, the system accurately identifies oncoming vehicles and adjusts the headlights to reduce glare for others while maintaining clear visibility for the driver. The modular design, with dedicated components for detection, communication, and control, provides quick, reliable performance across various conditions, while the simulation setup allows thorough testing and refinement before real-world use. This system offers a practical solution for safer nighttime driving, with real-time adaptability that makes it suitable for integration into modern vehicle systems, contributing to the advancement of smart automotive technology.

VII.REFERENCE

1. C. Susana Martinez, S.L.Macknik, D.H.Hubel The role of fixational eye movements in visual perception. Nature Reviews Neuroscience. 2004; 5: 229–240P.

2. Ryota Kanai, Yukiyasu Kamitani, Universiteit Utrecht. Time-locked perceptual fading Induced by visual Transients.

3. S. Aishwarya. Bright Headlights a major cause of accidents. The Hindu, Online Edn; 2006.

4. C. Guttman. High intensity headlights could cause road accidents by dazzling oncoming drivers. Eurotimes; 2003.

5. J. J. Fazzalaro. Limitations on Headlight brightness. OLD Research Report. 2003; 87(1): 113–117p.

6. S.T.Chrysler, P.J.Carlson, H.Gene Hawkins. Imapcts of Retro reflectivity on sign management. 2003.

7. Lighting the future standard and high performance automotive halogen bulbs-Hella.

8. A.Majumder, S.Irani. Contrast enhancement of images using human contrast sensitivity.

9. A.B.Watson, Temporal sensitivity. Vision RPS. 1969; 9: 947–952p.

10. R . Shapley , E.Kaplan, K.Purpura. Contrast sensitivity and light adaptation in photoreceptors or in the retinal network; 1993.