International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Intelligent Traffic Management for Congestion Control using YOLO Algorithmhm

Mrs.Susmitha BC
Assistant Professor
Department of Electronics and Communications
ACS College of Engineering, Bangalore,India
bc.susmitha@gmail.com

Nithin Nicholas
UG Scholar
Department of Electronics and Communications
ACS College of Engineering, Bangalore,India
nithinnicholas17@gmail.com

Tejas J
UG Scholar
Department of Electronics and Communications
ACS College of Engineering, Bangalore,India
tejasyadav5483@gmail.com

Tejashwini B
UG Scholar
Department of Electronics and Communications
ACS College of Engineering, Bangalore,India
tejashwini.b5696@gmail.com

Ujwal Gowda V
UG Scholar
Department of Electronics and Communications
ACS College of Engineering, Bangalore,India
ujwalviji2000@gmail.com

***Abstract*—The number of cars on roads keeps growing daily; old ways of managing traffic just can't keep up anymore. Those outdated setups run okay if there aren't many vehicles around. But once things get crowded - particularly if one lane backs up more than the opposite side - fixed light schedules fall apart, causing extended waits plus sluggish movement. To tackle this problem, our idea is a smart traffic light setup that swaps timed changes for instant adjustments based on current road flow. Instead of preset schedules, signals shift when needed, guided by constant visual monitoring of cars. Using fast image recognition, it keeps track of how many vehicles are around and how backed up they get. With the help of YOLO, a tool famous for spotting objects quick and right it reads camera footage in real time to measure crowding and movement. With YOLO, the system spots cars instantly - then adjusts light timing based on live info. Heavy lanes get priority so jams clear faster, yet emptier roads don't sit idle for no reason. By shifting signals smartly, it cuts delays, eases crowding, lowers wait times, boosts safety, and keeps traffic moving smoother. This smart traffic system shows how today's AI tools can change city travel making jams easier to handle through quicker, smoother responses that adapt on the fly.**

Keywords*— *Open CV, YOLO Algorithm, Real-Time Traffic Management, Image Processing, Machine Learning.

I. INTRODUCTION

In modern times, folks usually go for driving their personal cars instead of hopping on buses or sharing rides. Because of this trend, there's been a quick rise in how many private cars are out on streets. Thanks to that surge, problems keep popping up - traffic jams stand out as a major headache. Even if people will keep driving their own cars, there's still a way to handle traffic better so roads get less jammed. A lot of today's projects are upgrading old transit setups turning them into smarter versions that respond faster. One example is called Intelligent Transport System, or ITS for short. Inside this setup, engineers built tools that watch traffic signals live, changing how long lights stay green depending on how many cars wait at each road end. Counting those vehicles doesn't rely on just one method - it uses several kinds of sensors instead. We aim to build a small setup showing actual road scenarios while tracking and handling vehicle movement at the same time. For this task, we rely on a ready-made YOLO system to spot objects. Instead of starting from scratch, we take advantage of YOLO's ability to identify cars using frame differences between moving parts and static areas. On top of that, OpenCV cleans up visual clutter in each picture so detection works better. CCTV units set up for watching areas can grab video from roads. After that, those visuals go into a ready-made system. Every roadside section gets split into pieces with the same size for snapping pics and checking them. The number of cars spotted per piece heads over to a small computer device. Depending on how many vehicles show up, correct light timings get picked per side. At first, it sees if the number of cars stays nearly steady through all video clips. When that's true, lights keep changing on their usual fixed schedule. But when car numbers shift too much from normal levels, adjustments happen instantly timing changes on the fly. This keeps vehicles moving more freely without jams.

II. Literature Survey

Development of a Machine Learning-Based Model for an Intelligent Ambulance Detection System using CNN project brings a smart way to spot ambulances so they don't get stuck in traffic jams. It runs on a small computer called Raspberry Pi, paired with a camera that grabs video as it happens. Instead of relying on standard methods, it uses a lightweight AI model - MobileNet-SSD - to recognize emergency vehicles quickly. Once the system sees an ambulance, it flips the traffic light to green without waiting. That change happens through relays tied to the Pi's output pins. So, the vehicle moves through smoothly, no hold-ups. A collection of 200 ambulance pictures came from video clips - each labeled with Labellmg. After that, those labels

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

changed format into TFRecords so TensorFlow could use them smoothly. Using an SSD setup, the system runs as fast as 14 frames per second. That speed works well on small devices, especially when cameras aren't high-res. This setup works well without costing much, while staying dependable enough to back up urgent rescue efforts. Because it gets ambulances through gridlock faster, this approach might cut delays dramatically - helping protect people in danger.

Real-Time Traffic Monitoring and Ambulance Prioritization Using YOLOv9 and Deep Learning project brings a smarter way to watch traffic and help ambulances move faster by using YOLOv9 along with deep reinforcement learning. As city roads get more crowded, rescue vehicles usually get stuck in jams. Instead of waiting, the setup checks camera feeds nonstop through YOLOv9, spotting cars and trucks across different lanes without mistakes. When it finds an ambulance, it instantly marks that road section as urgent - changing stoplights ahead so the path clears out fast. The setup uses Deep Reinforcement Learning, so it picks up trends from past plus live traffic to tweak light schedules on the fly. Instead of just counting cars, it tracks how packed lanes are and how smoothly traffic moves, giving controllers clear insights. Tests prove wait times drop by about a quarter while emergency units arrive 30 to 40 percent quicker. In short, this smart system boosts safety, eases jams, and helps ambulances or fire trucks get through faster.

III. Proposed System

The suggested setup tackles today's traffic problems by adjusting lights according to live data. Instead of relying on fixed timers, it responds as things change on the ground. For spotting cars, an already trained YOLO model picks them out and tallies how many there are. Once counted, that number helps decide when signals should switch. It works with nearly every camera brand - even basic security ones you can buy cheap. Still, for best results, one Pi camera handles shots from all four directions. That's because it connects smoothly with the Laptop board without hiccups. The pictures taken go straight to the YOLO system so it can spot vehicles - after that, counting begins, just like in the diagram. Instead of switching gear, the same camera handles every side of the crossroad one after another.

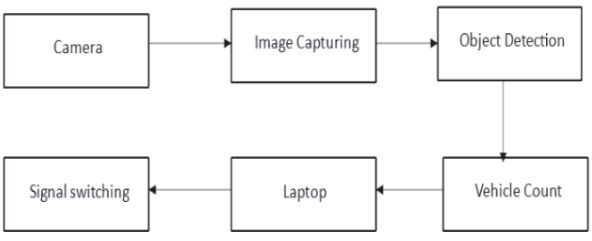

Fig 1: Proposed System

The suggested setup tackles today's traffic problems by adjusting lights according to live data. Instead of relying on fixed timers, it responds as things change on the ground. For spotting cars, an already trained YOLO model picks them out and tallies how many there are. Once counted, that number helps decide when signals should switch. It works with nearly every camera brand - even basic security ones you can buy cheap. Still, for best results, one Pi camera handles shots from all four directions. That's because it connects smoothly with the Laptop 3 board without hiccups. The pictures taken go straight to the YOLO system so it can spot vehicles - after that, counting begins, just like in the diagram. Instead of switching gear, the same camera handles every side of the crossroad one after another.

Image Acquisition: An image's usually seen as a flat grid labeled f(x, y), with x, y marking spots on a plane. At each spot, the number tells how bright or dark that part looks. Turning this smooth layout into something digital means slicing those positions into separate steps. Each location gets a fixed value instead of flowing smoothly. Digital pictures are made up of tiny bits, every one called a pixel. Here, the pictures used come from eye scans pulled from the STARE or DRIVE collections. Formation of an Image: A picture f(x, y) needs specific rules to exist. Because the numbers show real energy from a source, they can't be zero or endless - instead, they've got to stay within limits while still carrying actual value.

So, here's the range that works:$0 < f(x, y)$ - that's way smaller than infinity

Image Pre-Processing

Image Resizing / Scaling:

Image scaling pops up nearly everywhere you see digital pictures - like turning raw camera data into full-colour images or making photos bigger. Every time a picture shifts from one size grid to another, that's scaling at work. You need it when adjusting pixel counts so things fit what your device or software needs. Switching sizes can go very differently depending on the method used - even if starting with the exact same image.

RGB to Grayscale Conversion:

People see colours using special eye cells named cones, tuned to light waves like red, green, or blue. These come in three kinds, each reacting to one main shade. Together, they mix signals so we can spot nearly every hue around us. Since colour pics keep separate levels for red, green, B info, turning them into gray tones makes things lighter. A simple way is taking the mean across every channel instead (R added to G, then divided by 3 alongside B) split into three parts equally Still, because people notice green light more clearly, using a mix that gives extra weight to green usually works better for making gray pictures look right.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

IV. ardware Requirements

The following are the hardware components used in Intelligent traffic management for congestion control using YOLO

Arduino UNO: The Arduino UNO runs the whole setup, linking every part of the smart traffic system together. When the YOLO module sends car info, it turns that into instant signal adjustments. Instead of fixed timers, lights change according to how busy each road is. With plenty of ports available, connecting screens or detectors takes little effort. The Arduino's easy coding setup speeds up building, trying out ideas, or tweaking things. Instead of jumping sharply, colours shift gradually from red to yellow then green. Acting like the brain, it manages step-by-step actions, pauses, along with sending signals back and forth. Here, it keeps every output running steady without hiccups. Low cost, adaptability, plus beginner-friendly design give it an edge for smart light experiments.

Power supply regulator: The power supply regulator keeps voltage steady so every part runs safely. Yet it blocks spikes that might harm delicate bits such as the Arduino, lights, or display screen. Instead of wild inputs, it delivers smooth output by changing rough or high volts into reliable ones. Because of this, the setup can work nonstop, crucial when watching traffic over time. With clean power, everything behaves right - no sudden stoppages happen. It keeps parts from getting too hot or hit by sudden surges. Power gets spread out fairly - thanks to the regulator - to the main chip, screen, maybe some LEDs. If things aren't controlled right, stuff can glitch up or give wrong results. Tougher performance comes from less strain on circuits; that's how it boosts trust in the build. Really, it's what holds the whole electrical setup together.

Webcam: The webcam acts as the main eye for the traffic setup. While it records live road scenes, it sends video chunks to spot cars using YOLO. With sharp visuals, the system can pick out vehicles while tracking jam levels. Since it streams nonstop, changes in movement are watched on the fly. Seeing everything on the road, the webcam works like a lookout. Instead of just sitting idle, it sends images straight to the YOLO model. This model checks each frame, spotting cars one by one while guessing how packed the street is. Unlike pricier tools, webcams cost little, fit almost anywhere, plus link up fast with regular computers. Because they deliver live video, traffic lights can shift timing right when roads get busy. In short, without this camera feeding fresh details, smart signal systems wouldn't work at all.

LCD display: The 16×2 LCD screen shows essential info in a clean, easy-to-read way. Data like car numbers, traffic crowding, or light state pops up on it. So people running tests can see how YOLO tracking shapes signal changes. As things run, live updates appear - helping users stay in sync. Instead of guessing, they get instant responses from the device. It links the tiny computer directly to whoever's watching. The small size and simple characters make reading quick plus clear. Power comes straight from the Arduino - no extra parts needed. Uses very little energy, so it can run nonstop without issues. Shows what's happening inside the system in a way you can actually see. All together, this makes the smart traffic model easier to use and understand.

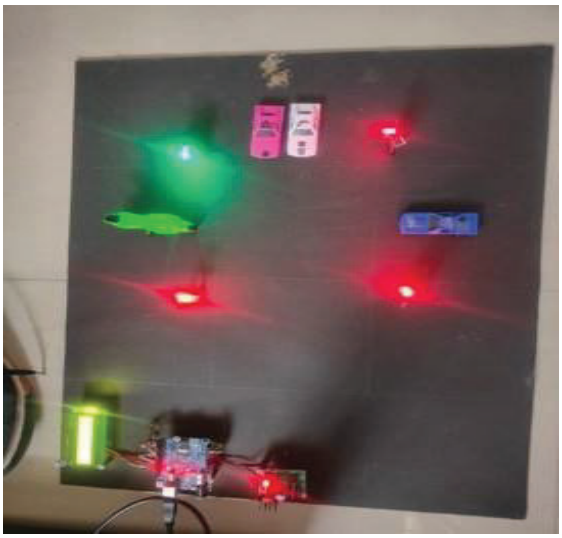

Fig 2:Hardware Components

V. Software Requirements

The open-source Arduino IDE makes coding and sending programs to Arduino boards way easier. It serves as the main hub for creating projects, letting people type, check, then push their code straight onto devices. Instead of being stuck on one system, this tool runs on Linux, macOS, and Windows - so folks can use what they already have. Because it works everywhere, more makers stick with it, whether tinkering at home or building serious gear. Arduino IDE works mainly with C or also C++ both common choices when building stuff for small computers inside devices. Arduino IDE links code with physical parts without hassle. So it helps build accurate light-switching routines using what YOLO spots. Since it's small, free, and works on most computers, tinkering feels smooth. Simple setup plus solid performance suits devices stuck inside machines. Altogether, this tool matters a lot when writing, checking, or running smart road-light rules.

OpenCV's an open-source toolkit made for working with visuals - like photos or moving footage. Instead of just linking pieces together, it grabs each video frame straight from your camera during live use. Because of how fast it runs, delays stay super low when checking data on the fly. On top of that, it cleans up blurry spots, reshapes pictures, pulls out key moments, also outlines vehicles spotted by YOLO. Thanks to built-in support for Python, hooking it up with smart detection systems feels smooth, almost like they were meant to work side-by-side. OpenCV boosts how well the system tracks changing traffic flow all the time. With OpenCV onboard, it outlines

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

cars, keeps track of numbers, while giving live visuals back. Fast processing means it runs without hiccups, even on regular devices. Widely used in labs and smart tech setups, it's trusted because it adapts easily and works reliably. For this build, it acts as the main base handling instant road checks and spotting jams**.**

VI. esult and Analysis

The intelligent traffic management system for congestion control using the YOLO algorithm developed in this project successfully demonstrates a practical and efficient solution for optimizing traffic flow through real-time image processing and machine learning techniques. The system integrates YOLO-based vehicle detection, OpenCV-based preprocessing, and a Raspberry Pi or laptop-based processing unit to dynamically regulate traffic signal timings based on real-time traffic density. One of the primary achievements of the project is the accurate detection and classification of vehicles using live video feeds captured through a camera.

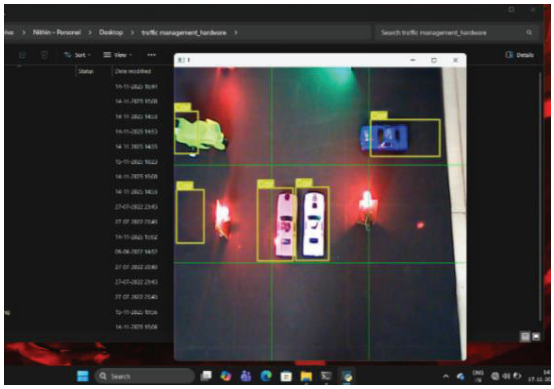

Fig.3: Vehicle detection

The YOLO-based model, supported by robust preprocessing steps such as smoothing, erosion, dilation, background modeling, and subtraction, consistently identified moving vehicles with high accuracy across multiple test scenarios. Bounding box visualization enabled clear visual confirmation of detections, while effective foreground extraction reduced false positives and improved classification reliability. Unit testing results confirmed that the vehicle detection module successfully passed all validation checks, demonstrating its robustness and suitability for real-time deployment. In addition to reliable detection, the system achieved precise vehicle counting, which is a critical requirement for estimating traffic density and enabling adaptive signal control. Frame-by-frame processing and region-of-interest segmentation allowed accurate counting across all four sides of the simulated junction, even under significantly varying traffic conditions. Multiple test cases validated the vehicle-counting subsystem, confirming its ability to generate dependable density data for decision-making. Based on the obtained vehicle density values, the system implemented effective dynamic signal switching, a key advancement over traditional fixed-time traffic control methods. The adaptive logic allocated longer green signal durations to highly congested lanes, minimized waiting time on low-density roads, and avoided inefficient signal allocation to empty lanes. Furthermore, the system intelligently switched between static and dynamic modes depending on whether vehicle counts crossed predefined threshold values, ensuring stability and responsiveness under normal and peak traffic conditions. The preprocessing pipeline played a crucial role in maintaining consistent system performance by generating cleaner foreground masks, eliminating noise, and stabilizing object isolation even in noisy frames or varying lighting environments. The seamless integration between the image processing modules and the hardware control logic further enhanced system reliability.

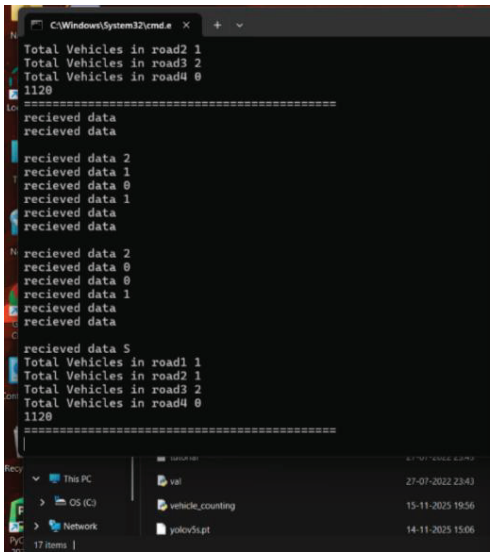

Fig.4: Vehicle Counting

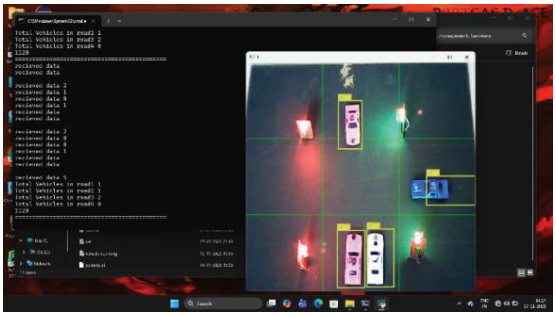

Fig.5: Final Result

Real-time communication between the detection, counting, and signal timing modules was achieved with minimal delay, enabling continuous operation and smooth signal transitions. Integration testing verified proper synchronization between software outputs and hardware actions, ensuring correct signal behavior in response to real-time traffic density variations. Testing outcomes demonstrated consistent system behavior across image capture, detection, counting, and synchronization tests, confirming the robustness of the complete system. Experimental analysis also revealed significant improvements in traffic flow efficiency, as dynamic signal switching reduced congestion during high-density scenarios while preventing unnecessary green signal time on sparsely populated lanes, thereby improving overall

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

junction throughput. An important outcome of the project is the confirmation that strong system performance can be achieved using minimal and affordable hardware, as the model operated effectively using standard webcams or CCTV footage without the need for additional sensors. This low-cost and scalable design enhances the feasibility of real-world deployment. Overall, the successful implementation and positive testing outcomes validate the effectiveness, reliability, and practical applicability of the proposed intelligent traffic management system, highlighting its potential integration into smart city infrastructure for enhanced traffic monitoring and congestion control.

VII. CONCLUSION

The creation of a smart traffic setup using YOLO is a solid move toward handling city traffic jams. Instead of relying on set schedules or simple sensors, it uses live car spotting along with self-adjusting signals for quicker, sharper responses. Since YOLO can spot various vehicles precisely and without lag, the system adapts fast when road patterns shift - keeping things moving better at peak times. This smart method cuts down idle time at crossroads

- while slashing fuel use and pollution, helping create greener cities. Thanks to deep learning, it gets better over time by studying old traffic flows, becoming sharper and more reliable. Since towns keep expanding and roads get busier, solutions like this can scale up easily without breaking the bank.

The setup works with many kinds of cameras - even low-cost ones so it can spread widely, especially where resources are tight. Because its design is adaptable, extra tools like spotting speeders, helping ambulances move faster, or watching out for people on foot can be added without hassle. Thanks to YOLO, tracking stays sharp whether it's raining, sunny, daytime, or dark, so things keep running smooth no matter the weather. Even though it works well, there are still problems - like blocked views, quick shifts in light, or slow processing that show we've got more work to do. Down the line, fixes might involve linking several cameras, using on-device computation, or upgraded YOLO versions that run faster and spot things more precisely. Intelligent traffic control powered by the YOLO method offers a solid way to cut jams, boost movement, while helping shape future-ready urban areas. This tech proves how artificial intelligence can turn regular road setups into sharp, adaptive grids that handle today's travel needs better.

VIII. REFERENCES

[1] Li Xun, alongside Nan Kaikai, followed by Liu Yao, together with Zuo Tao - "A Live Traffic Spotting Technique Using an Upgraded Kalman Filter." Featured in the 2018 Third Global Meet on Robot & Auto Engineering (ICRAE), held in Guangzhou, China, from Nov 17 to 19, 2018.

[2] Safoora Maqbool, along with Mehwish Khan and Jawaria Tahir, worked together with Abdul Jalil; then came Ahmad Ali plus Javed Ahmad. Their topic? Spotting vehicles, following their movement, also counting them. They showed this at an event - the 2018 IEEE ICSIP conference - held mid-July in Shenzhen, China.

[3] R. Krishnamoorthy along with Sethu Manickam explored automated traffic tracking via image vision. Their work appeared at ICICCT 2018 - the second global conference on creative communication and computing tech. It took place in Coimbatore, India. Dates were April 20th through 21st in 2018.

[4] Jess Tyron G. Nodado, then Hans Christian P. Morales alongside Ma. Angelica P. Abugan, followed by Jerick L. Olisea together with Angelo C. Aralar, plus Pocholo James M. Loresco presented a smart traffic light setup using computer vision; it included Android-based monitoring and control. Their work appeared in the TENCON 2018 – IEEE Region 10 Conference held in Jeju, South Korea.

[5] Sayan Mondal, along with Alan Yessenbayev and Jahya Burke, plus Nihar Wahal explored how neural object detection systems gather information. Their work was shared at NeurIPS 2018 - held in Montreal, Canada. The event marked its 32nd edition that y

[6] Sujin Jose Arul, along with Mithilesh B. S., Shreyas L., Sufiyan, Gopal Kaliyaperumal - plus Jayasheel Kumar K. - worked on modeling a smart traffic light setup for emergency vehicles; they used image processing methods. The study was shared during ICIPTM 2024.

[7] Elumalaivasan Poongavanam alongside Mahavishnu C., Prathik A., Kayalvizhi V., together with Rajesh R. worked on spotting emergency vehicles using a 2D deep learning model plus visual tech. They showed this research during ICETITE 2024.

[8] P. Kaladevi, along with Balamurugan M., Gokul D., plus Gopika S., worked on live traffic tracking and giving ambulances first pass using YOLOv9 paired with deep learning methods - shared during ICCSAI 2025.