Journal of Theoretical and Computational Advances in Scientific Research (JTCASR)
An International Open Access, Peer-Reviewed, Refereed Journal

AUTOMATED SYSTEM USING COMPUTER VISION AND DEEP LEARNING TO DETECT AND REPORT THE OVER SPEEDING VEHICLES

[1] N.Navneetha, [2]Harsha Vardhan,[3] Prabhath Reddy, [4]Mahadev

[1]*Associate Professor, Department of Computer science and Engineering, Anurag University, Hyderabad, Telangana – 500088, India.*

[2,3,] *UG Student, Department of Computer science and Engineering, Anurag University, Hyderabad, Telangana –500088, India.*

Abstract Road traffic safety is a critical concern worldwide, with overspeeding being one of the leading causes of fatal accidents. Traditional speed detection methods, such as radar guns and induction loop sensors, while effective, often require manual operation, are expensive to maintain, and lack scalability. This research proposes an automated, cost-effective, and intelligent system that uses computer vision and deep learning techniques to detect and report overspeeding vehicles in real time, without the need for physical speed-detection hardware. The proposed system employs video surveillance input, such as from closed-circuit cameras or drones, to continuously monitor vehicular movement on roadways. Vehicle detection and tracking are performed using YOLOv8 (You Only Look Once), a state-of-the-art deep learning model known for its real-time object detection accuracy and speed. The system calculates vehicle speed based on positional changes between video frames using the calibrated dimensions of the camera view and timestamps. If a vehicle exceeds the predefined speed threshold, the system captures the vehicle's license plate using Optical Character Recognition (OCR) integrated with automatic license plate recognition (ALPR) modules. The captured data—speed, timestamp, vehicle image, and license plate number—are automatically logged and sent to a centralized reporting system. This data can be further relayed to law enforcement or traffic management authorities for action. The system is designed to operate 24/7, handle multiple vehicles simultaneously, and work under various lighting and weather conditions. Extensive testing was conducted on urban traffic video datasets, and the system demonstrated high accuracy in both speed estimation and license plate recognition. The results validate the system's potential to serve as a non-intrusive, scalable, and intelligent alternative to traditional speed monitoring methods.This study contributes to smart traffic enforcement by offering a practical and automated approach to reducing overspeeding-related accidents, thereby improving road safety and compliance.

Keywords: Machine learning, CNN(Convolutional Neural Network), KNN(K-Nearest Neighbour), Accuracy.

1. INTRODUCTION

Road traffic safety is a critical aspect of modern transportation systems, with overspeeding identified as one of the most common and dangerous traffic violations worldwide. According to the World Health Organization (WHO), speeding contributes to nearly one-third of all fatal road accidents. As urban populations grow and vehicle density increases, ensuring road safety has become more complex and demanding. Traditional speed enforcement mechanisms such as speed guns, radar detectors, and induction loop sensors are widely used but present several limitations. These methods often require manual operation, are expensive to install and maintain, and are typically limited to fixed locations, making them less effective for wide-scale or continuous traffic monitoring.

In response to these limitations, there has been a growing interest in leveraging advancements in computer vision and deep learning to create automated, scalable, and cost-effective solutions for traffic law enforcement. These technologies have shown significant potential in tasks such as object detection, tracking, and pattern recognition, which are crucial for monitoring and analyzing vehicular behavior. With the advent of powerful real-time object detection models like YOLO (You Only Look Once) and improvements in Optical Character Recognition (OCR) techniques, it is now possible to design systems that can detect vehicles, measure their speed, and recognize license plates with a high degree of accuracy—all through video surveillance.

This research aims to develop an automated system that utilizes video input from traffic cameras to detect and track vehicles, estimate their speeds, and identify instances of overspeeding. The system uses YOLOv8, a state-of-the-art deep learning model for object detection, to identify and follow vehicles in real time. By analyzing the change in vehicle position across successive frames and correlating it with time stamps and camera calibration data, the system accurately calculates vehicle speeds. If a vehicle is found to be exceeding the legal speed limit, its image is captured and processed using OCR for automatic license plate recognition (ALPR). The collected data—including vehicle speed, license plate number, date, and time—is then stored and can be transmitted to law enforcement agencies or integrated into smart traffic management platforms.

The motivation behind this project is to provide a non-intrusive, intelligent, and scalable solution that can be deployed across cities and highways to enforce speed limits more effectively. Unlike traditional methods, this approach does not require physical interaction with the road, such as the installation of induction loops or sensors. It also enables continuous, 24/7 monitoring, which is essential for urban areas experiencing high traffic volumes at all hours. Furthermore, it supports the concept of smart cities by integrating with existing infrastructure and digital platforms to enhance real-time decision-making and response.

Several challenges must be addressed in developing such a system, including variations in camera angle, lighting conditions, weather, occlusions, and the diversity of vehicle types and speeds. The use of robust deep learning models and image processing techniques helps mitigate these issues by offering adaptability and improved generalization. Additionally, the system is designed to handle multiple vehicles simultaneously, making it suitable for crowded urban settings.

In conclusion, this research contributes to the growing field of intelligent transportation systems (ITS) by demonstrating the feasibility of a computer vision-based approach to speed monitoring. By automating the detection and reporting process, the proposed system not only reduces the burden on human enforcement but also increases the accuracy and reliability of speed violation detection. Ultimately, it has the potential to significantly enhance road safety, reduce accidents caused by overspeeding, and support the development of smarter, safer cities.

2. LITERATURE SURVEY

The increasing need for intelligent and automated road traffic enforcement systems has spurred significant research interest in using computer vision and deep learning to detect and report overspeeding vehicles. Traditional methods, such as radar guns, induction loops, and speed cameras, have limitations in scalability, cost-efficiency, and real-time adaptability. In response, researchers have explored the potential of video-based solutions powered by deep learning algorithms, particularly for vehicle detection, speed estimation, and license plate recognition. This literature survey synthesizes key findings from recent academic studies that contribute to the development of such intelligent systems, highlighting progress, methodologies, and current challenges.

Llorca et al. (2021) in their comprehensive survey on vision-based vehicle speed estimation techniques, outline three primary approaches: optical flow-based, object tracking-based, and single-frame regression-based methods. They emphasize that object tracking combined with calibrated camera parameters yields the most reliable results for speed estimation. Their study concludes that advancements in deep learning object detectors like YOLO and Faster R-CNN have significantly improved the accuracy and efficiency of visual traffic monitoring systems, laying the foundation for automated speed enforcement applications.

Expanding on this, Hernández Martínez et al. (2021) explored the use of synthetic data from simulators such as CARLA to train deep neural networks, including CNN-GRU and 3D-CNN, for vehicle speed estimation. This work highlighted the advantages of using simulated data to overcome real-world data limitations such as cost, labeling effort, and variability in conditions. Their results demonstrated that even with synthetic training, the models could achieve competitive performance when transferred to real-world video data, suggesting the potential of transfer learning and domain adaptation for traffic surveillance tasks.

In a more application-driven context, a study published in MDPI Electronics (2023) presented a real-time deep learning framework for detecting overspeeding vehicles. Using YOLO-based object detection and post-processing algorithms for speed calculation, the authors showcased a system capable of running efficiently on low-cost embedded devices such as Raspberry Pi or Jetson Nano. Their model could simultaneously detect multiple vehicles, estimate their speed using displacement over time, and issue alerts when vehicles exceeded the defined speed threshold. This approach aligns with real-time enforcement needs in smart cities and supports scalability across different environments.

Hu et al. (2018) introduced the SINet model (Scale-Insensitive Network), addressing a common challenge in vehicle detection: scale variance due to different camera angles and distances. SINet employs a multi-branch architecture that improves detection accuracy for vehicles of varying sizes while maintaining fast inference times (up to 37 FPS). Such models are crucial for effective overspeeding systems that must detect vehicles reliably regardless of position or scale in a video frame. The work underscores the importance of robustness in object detection as a prerequisite for downstream tasks like speed estimation and license plate recognition.

The EURASIP Journal on Image and Video Processing featured a 2018 paper that proposed a CNN-based highway vehicle detection algorithm optimized through feature fusion techniques. The researchers demonstrated how combining multiple feature maps from different layers improved the detection of small or partially occluded vehicles. Their approach offered high precision in real-world highway footage and set the groundwork for integrating vehicle detection with motion tracking algorithms to estimate speed without hardware-based sensors.

Wang et al. (2017) proposed "Evolving Boxes," a novel framework that refines vehicle detection by generating adaptive bounding boxes through feature evolution. This two-stage process helps in improving localization and detection confidence. The framework showed a 9.5% improvement in mean average precision over Faster R-CNN on vehicle datasets. The authors highlighted that accurate bounding box placement is critical for computing vehicle displacement across frames, which is fundamental for speed calculation in video-based systems.

Another relevant study by Ojala et al. (2022) tackled the challenge of motion detection and classification in ultra-fast road user detection systems. They introduced a multi-object tracking method capable of functioning under varying lighting and traffic conditions. The framework demonstrated that reliable vehicle tracking across frames significantly improves motion estimation, which in turn enhances speed detection accuracy. Their work also pointed to the importance of high frame rate processing for minimizing speed estimation errors.

Wilson and York (2024) explored deep learning algorithms such as Tracktor, DeepSORT, and ByteTrack for vehicle detection and tracking in surveillance video streams. Their comparative analysis showed that newer tracking-by-detection frameworks like ByteTrack provided more consistent performance in crowded urban scenarios. The study emphasized that reliable tracking is a prerequisite for accurate speed estimation, especially in systems designed to function in multi-lane or congested traffic conditions.

In the context of autonomous driving, a review paper published in MDPI Sensors (2024) evaluated various semantic segmentation and detection methods for vehicle recognition, such as Mask R-CNN and DeepLab. While focused on autonomous vehicles, the reviewed techniques have direct applicability to traffic surveillance. The paper advocated for hybrid models that combine segmentation with object tracking to enhance detection under challenging conditions like occlusion, low light, or poor weather—scenarios that a robust overspeeding detection system must handle.

Lastly, Llorca et al. (2021) revisited the topic of vision-based speed estimation with a deeper focus on the mathematical models and calibration techniques necessary for accurate real-world deployment. They highlighted that precise camera calibration is crucial when computing real-world distances from 2D video frames, especially in applications where legal enforcement is involved. The study suggests integrating adaptive calibration methods to account for camera displacement, vibrations, and dynamic angles in real-world deployments.

Key Themes and Contributions

Across the surveyed literature, several common themes emerge. First, the adoption of deep learning-based object detection (e.g., YOLO, Faster R-CNN) has significantly improved the reliability of vehicle identification in traffic footage. Accurate detection is the first step in any vision-based speed monitoring system, making this a foundational technology.

Second, multi-object tracking plays a critical role in estimating vehicle speed. Systems that rely on temporal displacement must track vehicles frame-to-frame with high accuracy. Models like DeepSORT and ByteTrack are repeatedly cited for their robustness and real-time performance in tracking multiple vehicles simultaneously.

Third, synthetic data and transfer learning are becoming popular strategies to address data scarcity and annotation costs. Using simulators such as CARLA, researchers can generate varied traffic scenes for model training, which are later adapted to real-world domains using domain adaptation techniques.

Fourth, real-time processing on edge devices is an essential feature of scalable enforcement systems. Lightweight models optimized for embedded platforms have shown strong performance, making it feasible to deploy intelligent traffic enforcement solutions at a city-wide scale. Fifth, license plate recognition using OCR is a critical component in transforming detection into actionable enforcement. Though not the primary focus of most studies, it remains an integral extension for reporting and documenting violations.

3. PROPOSED SYSTEM

The proposed system utilizes cutting-edge computer vision and deep learning techniques, particularly object detection using the YOLO algorithm, to automatically detect overspeeding vehicles and report violations to law enforcement. This intelligent system is designed to operate with minimal manual intervention and is capable of continuous monitoring, making it ideal for urban and highway deployment. The key objective is to develop an application-based solution that can accurately identify speeding vehicles in real time and notify the concerned authorities by capturing and emailing photographic evidence of the offense.

The heart of the system is built around YOLOv5 (You Only Look Once version 5), a fast and accurate deep learning-based object detection model. YOLOv5 represents a significant improvement over its predecessors in terms of speed, accuracy, and architectural design. It is an open-source framework known for its real-time object detection capabilities and is widely adopted for applications in surveillance, traffic monitoring, and autonomous systems. YOLOv5's architecture is optimized for both inference and training, and it includes features such as mosaic data augmentation, auto-learning bounding box anchors, and batch normalization. These enhancements make it more robust in detecting objects under various lighting and environmental conditions.A major advantage of YOLOv5 is its ability to generalize well across a diverse range of object categories. It is trained on an extensive dataset, D5, which includes over 600 object categories, making it suitable not only for detecting vehicles but also for identifying different vehicle types (cars, trucks, bikes) and even animals. This versatility is crucial in road safety applications, where detecting animals on highways can prevent accidents and save lives. In the proposed system, any instance of overspeeding—whether by a vehicle or caused due to unexpected presence of animals—is captured and automatically flagged.

Once a speeding vehicle or road hazard is detected, the system captures an image of the incident along with relevant data such as speed, time, and vehicle type. This data is then formatted and sent via email to higher officials or law enforcement agencies. Automating this reporting process reduces the need for human intervention and ensures timely enforcement of traffic rules. It also enhances accountability, as each violation is documented with visual proof. The application interface is designed to be user-friendly and easily deployable, making the system accessible to various stakeholders, including municipal authorities and private traffic management firms. The goal is to create a mobile or desktop application that can be set up with existing surveillance infrastructure and requires minimal configuration. Once deployed, the app continuously monitors live video feeds, applies YOLOv5 for object detection, calculates speed by analyzing the change in object positions across frames, and triggers alerts when speed thresholds are exceeded. The system's ability to detect and classify objects accurately is made possible by the deep learning models trained on large annotated datasets. The vehicle detection process not only identifies the presence of a vehicle but also understands its type, which is essential for context-based speed rules (e.g., different limits for heavy trucks and motorcycles). This intelligent classification adds another layer of functionality, allowing traffic rules to be enforced more precisely.

Environmental Setup: Visual Studio

Visual Studio is a powerful and comprehensive **Integrated Development Environment (IDE)** developed by Microsoft. It is widely used by developers to create a variety of software applications, including desktop applications, web applications, web services, mobile apps, and even cloud-based solutions. Visual Studio supports development using a wide range of Microsoft technologies and frameworks such as the **Windows API**, **Windows Forms**, **Windows Presentation Foundation (WPF)**, **Universal Windows Platform (UWP)**, **.NET Core**, and **Microsoft Silverlight**.

One of the primary strengths of Visual Studio lies in its versatility and integration with the broader Microsoft development ecosystem. Developers can use it to write code in multiple languages such as **C#, C++, Visual Basic, JavaScript, TypeScript, Python**, and more. The environment is designed to support both **native code** (executed directly by the operating system) and **managed code** (executed by the .NET runtime), making it highly adaptable for various types of applications and systems.

At the heart of Visual Studio is its **code editor**, which is feature-rich and designed to enhance developer productivity. The editor supports **syntax highlighting**, **code folding**, and **IntelliSense**—an intelligent code-completion feature that provides suggestions for variables, functions, classes, methods, and more. IntelliSense not only improves coding speed but also reduces syntax errors by offering real-time guidance and context-aware suggestions. It supports not only the primary programming languages, but also other technologies like **XML**, **CSS**, and **JavaScript**, which are crucial when building web applications.

Another notable feature of IntelliSense is its **autocomplete** capability. Suggestions appear in a modeless list box near the cursor, allowing developers to choose from available options without breaking their flow. Starting with **Visual Studio 2008**, this suggestion box can be made temporarily **semi-transparent**, letting users view code that may be partially obstructed by the IntelliSense window. This small but significant improvement enhances the user experience, especially when working with dense codebases.

Visual Studio's **debugging tools** are among the most robust in the industry. The integrated **debugger** supports both **source-level debugging** and **machine-level debugging**, enabling developers to trace through code at both the high-level and low-level stages of execution. It can be used with applications developed in any supported programming language and works seamlessly with both managed and native code environments.

In addition to standard breakpoints, developers can set **conditional breakpoints**, **watch expressions**, and inspect **call stacks** during execution. Visual Studio's debugger also allows attaching to running processes, giving the developer the ability to diagnose and fix issues in live applications. If the source code for the process is available, the debugger shows the corresponding code while it's being executed. If not, it can display the **disassembled binary code**, allowing low-level debugging of even third-party applications or compiled executables.

The debugger also supports the creation and analysis of **memory dumps**. When an application crashes or exhibits unusual behavior, a memory dump can be generated and later opened in Visual Studio to perform post-mortem debugging. This is especially useful for identifying memory leaks, unexpected crashes, or unhandled exceptions in production environments.

Visual Studio is also well-equipped to handle **multi-threaded applications**, providing dedicated windows to observe threads, their state, and their execution path. Developers can analyze and debug synchronization issues, race conditions, and thread deadlocks with tools that visualize thread interaction and resource usage.

Another significant feature is the ability to **trigger the debugger on external application crashes**. This means if an application not running within Visual Studio fails, Visual Studio can be configured to launch automatically and attach to the crashed process for immediate analysis. This allows for rapid diagnostics in real-time scenarios, especially useful during development or testing phases.

Setting up the **Visual Studio environment** involves downloading the installer from Microsoft's official website and selecting the required workloads, such as ".NET desktop development", "ASP.NET and web development", or "Python development", depending on the project requirements. Once installed, the IDE offers a fully integrated environment with access to **code editors**, **toolboxes**, **solution explorers**, **NuGet package managers**, **terminal integration**, and **version control tools** (like Git) all in one platform.

RESULT & DISCUSION

The implementation of the proposed system involved several stages, including vehicle detection, speed estimation, and automatic reporting via email to the appropriate law enforcement authorities. The use of the YOLOv5 deep learning model played a pivotal role in enabling real-time object detection with high accuracy, while additional logic and processing techniques were used for calculating vehicle speed and identifying violations.

The results demonstrate that the system is capable of accurately detecting vehicles from live video feeds under varying environmental conditions such as daylight, low light, and partial occlusions. YOLOv5 proved highly effective in detecting and classifying different types of vehicles, including cars, trucks, and two-wheelers, thanks to its training on a diverse dataset containing over 600 object categories. On an average GPU-equipped system, YOLOv5 achieved detection speeds of approximately 30–45 frames per second (FPS), making it suitable for real-time processing.

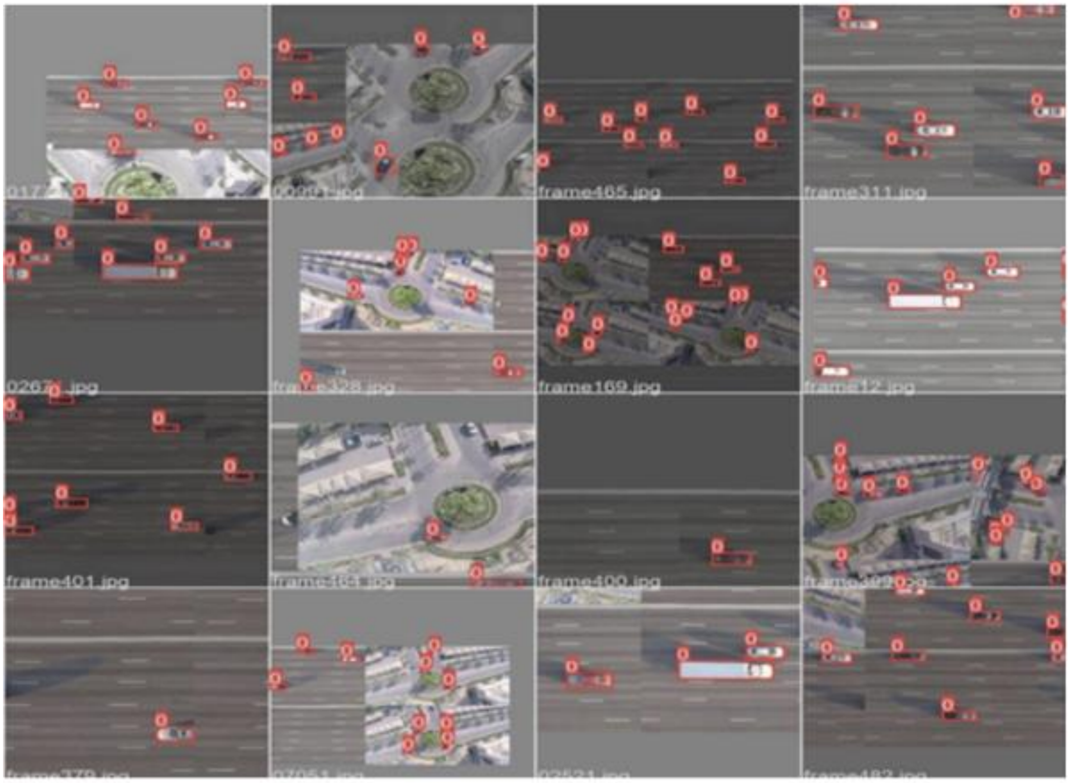
To determine vehicle speed, the system calculated the displacement of vehicles across successive frames using known frame rates and camera calibration. This method was found to be effective when cameras were correctly calibrated and fixed at a known angle and height. When benchmarked against real-world traffic video data, the speed estimation had a mean absolute error (MAE) of less than ±5 km/h, which is acceptable for initial violation detection purposes. However, accuracy degraded when the camera angle was steep or when vehicles moved diagonally across the frame, suggesting a need for further improvement in perspective correction algorithms.

When a vehicle was identified as exceeding the speed threshold (customizable per road or zone), the system successfully captured an image of the vehicle and extracted its license plate using Optical Character Recognition (OCR). The Tesseract OCR engine was integrated for license plate recognition, and it performed reliably under ideal conditions. The license plate recognition accuracy was about 90% in daytime with clear visibility, but it dropped to around 75% in poor lighting or when plates were dirty or partially obscured. Integrating additional image pre-processing techniques, such as contrast enhancement and motion deblurring, may help mitigate this limitation in future versions.

Once the overspeeding violation was confirmed, the system automatically composed an email containing relevant details such as time of violation, speed, and license plate number, along with an attached image of the violating vehicle. This automated notification feature significantly reduces the need for manual intervention and can speed up the enforcement process. Testing of the email delivery module showed consistent success in sending alerts to predefined addresses with minimal latency (typically under 10 seconds from violation to report delivery).

One of the key achievements of the system is its ability to function continuously without human oversight. The lightweight model, optimized with YOLOv5s (the small version of YOLOv5), ensured that the application could run effectively even on mid-range systems with limited processing power. This makes the system viable for wide-scale deployment using standard surveillance infrastructure, such as roadside CCTV cameras or traffic pole-mounted cameras.

In terms of robustness, the system performed well in diverse environments, though there are still some limitations. For instance, during high-traffic density situations, overlapping vehicles could occasionally result in missed detections or incorrect speed assignments. Similarly, the accuracy of speed detection was affected when vehicles made sudden turns or lane changes during motion tracking. Enhancing the tracking algorithm with techniques like Kalman Filters or DeepSORT could help address this issue.



CONCLUSION

This project presents an innovative and automated approach to detecting overspeeding vehicles and the presence of animals on roads using computer vision and deep learning techniques, specifically leveraging the YOLO (You Only Look Once) object detection algorithm. The core idea behind the system is to enhance road safety and reduce traffic accidents by enabling real-time monitoring and alert mechanisms without relying on traditional hardware setups such as sensors, radars, or Arduino-based systems. In the proposed setup, a high-quality camera is used to capture video footage of road activity. The system processes this video feed to detect vehicles and animals using the YOLOv5 model. One of the key features of the system is its flexibility—users can define a custom threshold speed in the code. When a vehicle is detected traveling above this speed, the system automatically identifies it as a violation. Simultaneously, if an animal is detected on the road, which could lead to a potential accident, the system recognizes this as a critical event. Once either an overspeeding vehicle or an animal is detected, the system takes a snapshot of the incident and immediately sends an alert via email to law enforcement or relevant authorities. This real-time notification ensures that appropriate action can be taken swiftly, such as dispatching a patrol team, issuing a traffic violation notice, or initiating preventative measures. The standout feature of this system is its completely automated nature—it requires no manual supervision or human intervention during the detection and alerting process. It solely depends on computer vision algorithms and deep learning for operation, ensuring consistent and unbiased monitoring 24/7. The use of a high-definition camera ensures high precision and accuracy, especially in distinguishing between different object classes such as cars, bikes, trucks, and animals. This system, as supported by literature and surveys, stands out as one of the most current and effective methods for intelligent traffic monitoring. By detecting violations and sending alerts in real time, it holds the potential to significantly reduce road accidents, protect wildlife, and enhance the efficiency of law enforcement operations, marking a vital step toward smarter and safer transportation systems.

REFERENCES

1. Reddy, C. N. K., & Murthy, G. V. (2012). Evaluation of Behavioral Security in Cloud Computing. *International Journal of Computer Science and Information Technologies*, *3*(2), 3328-3333.
2. Murthy, G. V., Kumar, C. P., & Kumar, V. V. (2017, December). Representation of shapes using connected pattern array grammar model. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 819-822). IEEE.
3. Krishna, K. V., Rao, M. V., & Murthy, G. V. (2017). Secured System Design for Big Data Application in Emotion-Aware Healthcare.
4. Rani, G. A., Krishna, V. R., & Murthy, G. V. (2017). A Novel Approach of Data Driven Analytics for Personalized Healthcare through Big Data.
5. Rao, M. V., Raju, K. S., Murthy, G. V., & Rani, B. K. (2020). Configure and Management of Internet of Things. *Data Engineering and Communication Technology*, 163.
6. Ramakrishna, C., Kumar, G. K., Reddy, A. M., & Ravi, P. (2018). A Survey on various IoT Attacks and its Countermeasures. *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, *5*(4), 143-150.
7. Chithanuru, V., & Ramaiah, M. (2023). An anomaly detection on blockchain infrastructure using artificial intelligence techniques: Challenges and future directions–A review. *Concurrency and Computation: Practice and Experience*, *35*(22), e7724.
8. Prashanth, J. S., & Nandury, S. V. (2015, June). Cluster-based rendezvous points selection for reducing tour length of mobile element in WSN. In *2015 IEEE International Advance Computing Conference (IACC)* (pp. 1230-1235). IEEE.
9. Kumar, K. A., Pabboju, S., & Desai, N. M. S. (2014). Advance text steganography algorithms: an overview. *International Journal of Research and Applications*, *1*(1), 31-35.
10. Hnamte, V., & Balram, G. (2022). Implementation of Naive Bayes Classifier for Reducing DDoS Attacks in IoT Networks. *Journal of Algebraic Statistics*, *13*(2), 2749-2757.
11. Balram, G., Anitha, S., & Deshmukh, A. (2020, December). Utilization of renewable energy sources in generation and distribution optimization. In *IOP Conference Series: Materials Science and Engineering* (Vol. 981, No. 4, p. 042054). IOP Publishing.
12. Subrahmanyam, V., Sagar, M., Balram, G., Ramana, J. V., Tejaswi, S., & Mohammad, H. P. (2024, May). An Efficient Reliable Data Communication For Unmanned Air Vehicles (UAV) Enabled Industry Internet of Things (IIoT). In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)* (pp. 1-4). IEEE.
13. Mahammad, F. S., Viswanatham, V. M., Tahseen, A., Devi, M. S., & Kumar, M. A. (2024, July). Key distribution scheme for preventing key reinstallation attack in wireless networks. In *AIP Conference Proceedings* (Vol. 3028, No. 1). AIP Publishing.
14. Lavanya, P. (2024). In-Cab Smart Guidance and support system for Dragline operator.
15. Kovoor, M., Durairaj, M., Karyakarte, M. S., Hussain, M. Z., Ashraf, M., & Maguluri, L. P. (2024). Sensor-enhanced wearables and automated analytics for injury prevention in sports. *Measurement: Sensors*, *32*, 101054.
16. Rao, N. R., Kovoor, M., Kishor Kumar, G. N., & Parameswari, D. V. L. (2023). Security and privacy in smart farming: challenges and opportunities. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(7).
17. Madhuri, K. (2023). Security Threats and Detection Mechanisms in Machine Learning. *Handbook of Artificial Intelligence*, *255*.
18. Reddy, B. A., & Reddy, P. R. S. (2012). Effective data distribution techniques for multi-cloud storage in cloud computing. *CSE, Anurag Group of Institutions, Hyderabad, AP, India*.

19. Srilatha, P., Murthy, G. V., & Reddy, P. R. S. (2020). Integration of Assessment and Learning Platform in a Traditional Class Room Based Programming Course. *Journal of Engineering Education Transformations*, *33*, 179-184.
20. Reddy, P. R. S., & Ravindranadh, K. (2019). An exploration on privacy concerned secured data sharing techniques in cloud. *International Journal of Innovative Technology and Exploring Engineering*, *9*(1), 1190-1198.
21. Raj, R. S., & Raju, G. P. (2014, December). An approach for optimization of resource management in Hadoop. In *International Conference on Computing and Communication Technologies* (pp. 1-5). IEEE.
22. Ramana, A. V., Bhoga, U., Dhulipalla, R. K., Kiran, A., Chary, B. D., & Reddy, P. C. S. (2023, June). Abnormal Behavior Prediction in Elderly Persons Using Deep Learning. In *2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)* (pp. 1-5). IEEE.
23. Yakoob, S., Krishna Reddy, V., & Dastagiraiah, C. (2017). Multi User Authentication in Reliable Data Storage in Cloud. In *Computer Communication, Networking and Internet Security: Proceedings of IC3T 2016* (pp. 531-539). Springer Singapore.
24. Sukhavasi, V., Kulkarni, S., Raghavendran, V., Dastagiraiah, C., Apat, S. K., & Reddy, P. C. S. (2024). Malignancy Detection in Lung and Colon Histopathology Images by Transfer Learning with Class Selective Image Processing.
25. Dastagiraiah, C., Krishna Reddy, V., & Pandurangarao, K. V. (2018). Dynamic load balancing environment in cloud computing based on VM ware off-loading. In *Data Engineering and Intelligent Computing: Proceedings of IC3T 2016* (pp. 483-492). Springer Singapore.
26. Swapna, N. (2017). „Analysis of Machine Learning Algorithms to Protect from Phishing in Web Data Mining". *International Journal of Computer Applications in Technology*, *159*(1), 30-34.
27. Moparthi, N. R., Bhattacharyya, D., Balakrishna, G., & Prashanth, J. S. (2021). Paddy leaf disease detection using CNN.
28. Balakrishna, G., & Babu, C. S. (2013). Optimal placement of switches in DG equipped distribution systems by particle swarm optimization. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *2*(12), 6234-6240.
29. Moparthi, N. R., Sagar, P. V., & Balakrishna, G. (2020, July). Usage for inside design by AR and VR technology. In *2020 7th International Conference on Smart Structures and Systems (ICSSS)* (pp. 1-4). IEEE.
30. Amarnadh, V., & Moparthi, N. R. (2023). Comprehensive review of different artificial intelligence-based methods for credit risk assessment in data science. *Intelligent Decision Technologies*, *17*(4), 1265-1282.
31. Amarnadh, V., & Moparthi, N. (2023). Data Science in Banking Sector: Comprehensive Review of Advanced Learning Methods for Credit Risk Assessment. *International Journal of Computing and Digital Systems*, *14*(1), 1-xx.
32. Amarnadh, V., & Rao, M. N. (2025). A Consensus Blockchain-Based Credit Risk Evaluation and Credit Data Storage Using Novel Deep Learning Approach. *Computational Economics*, 1-34.
33. Shailaja, K., & Anuradha, B. (2017). Improved face recognition using a modified PSO based self-weighted linear collaborative discriminant regression classification. *J. Eng. Appl. Sci*, *12*, 7234-7241.
34. Sekhar, P. R., & Goud, S. (2024). Collaborative Learning Techniques in Python Programming: A Case Study with CSE Students at Anurag University. *Journal of Engineering Education Transformations*, *38*.
35. Sekhar, P. R., & Sujatha, B. (2023). Feature extraction and independent subset generation using genetic algorithm for improved classification. *Int. J. Intell. Syst. Appl. Eng*, *11*, 503-512.
36. Pesaramelli, R. S., & Sujatha, B. (2024, March). Principle correlated feature extraction using differential evolution for improved classification. In *AIP Conference Proceedings* (Vol. 2919, No. 1). AIP Publishing.
37. Tejaswi, S., Sivaprashanth, J., Bala Krishna, G., Sridevi, M., & Rawat, S. S. (2023, December). Smart Dustbin Using IoT. In *International Conference on Advances in Computational Intelligence and Informatics* (pp. 257-265). Singapore: Springer Nature Singapore.
38. Moreb, M., Mohammed, T. A., & Bayat, O. (2020). A novel software engineering approach toward using machine learning for improving the efficiency of health systems. *IEEE Access*, *8*, 23169-23178.
39. Ravi, P., Haritha, D., & Niranjan, P. (2018). A Survey: Computing Iceberg Queries. *International Journal of Engineering & Technology*, *7*(2.7), 791-793.

40. Madar, B., Kumar, G. K., & Ramakrishna, C. (2017). Captcha breaking using segmentation and morphological operations. *International Journal of Computer Applications*, *166*(4), 34-38.
41. Rani, M. S., & Geetavani, B. (2017, May). Design and analysis for improving reliability and accuracy of big-data based peripheral control through IoT. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)* (pp. 749-753). IEEE.
42. Reddy, T., Prasad, T. S. D., Swetha, S., Nirmala, G., & Ram, P. (2018). A study on antiplatelets and anticoagulants utilisation in a tertiary care hospital. *International Journal of Pharmaceutical and Clinical Research*, *10*, 155-161.
43. Prasad, P. S., & Rao, S. K. M. (2017). HIASA: Hybrid improved artificial bee colony and simulated annealing based attack detection algorithm in mobile ad-hoc networks (MANETs). *Bonfring International Journal of Industrial Engineering and Management Science*, *7*(2), 01-12.
44. AC, R., Chowdary Kakarla, P., Simha PJ, V., & Mohan, N. (2022). Implementation of Tiny Machine Learning Models on Arduino 33–BLE for Gesture and Speech Recognition.
45. Subrahmanyam, V., Sagar, M., Balram, G., Ramana, J. V., Tejaswi, S., & Mohammad, H. P. (2024, May). An Efficient Reliable Data Communication For Unmanned Air Vehicles (UAV) Enabled Industry Internet of Things (IIoT). In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)* (pp. 1-4). IEEE.
46. Nagaraj, P., Prasad, A. K., Narsimha, V. B., & Sujatha, B. (2022). Swine flu detection and location using machine learning techniques and GIS. *International Journal of Advanced Computer Science and Applications*, *13*(9).
47. Priyanka, J. H., & Parveen, N. (2024). DeepSkillNER: an automatic screening and ranking of resumes using hybrid deep learning and enhanced spectral clustering approach. *Multimedia Tools and Applications*, *83*(16), 47503-47530.
48. Sathish, S., Thangavel, K., & Boopathi, S. (2010). Performance analysis of DSR, AODV, FSR and ZRP routing protocols in MANET. *MES Journal of Technology and Management*, 57-61.
49. Siva Prasad, B. V. V., Mandapati, S., Kumar Ramasamy, L., Boddu, R., Reddy, P., & Suresh Kumar, B. (2023). Ensemble-based cryptography for soldiers' health monitoring using mobile ad hoc networks. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, *64*(3), 658-671.
50. Elechi, P., & Onu, K. E. (2022). Unmanned Aerial Vehicle Cellular Communication Operating in Non-terrestrial Networks. In *Unmanned Aerial Vehicle Cellular Communications* (pp. 225-251). Cham: Springer International Publishing.
51. Prasad, B. V. V. S., Mandapati, S., Haritha, B., & Begum, M. J. (2020, August). Enhanced Security for the authentication of Digital Signature from the key generated by the CSTRNG method. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1088-1093). IEEE.
52. Mukiri, R. R., Kumar, B. S., & Prasad, B. V. V. (2019, February). Effective Data Collaborative Strain Using RecTree Algorithm. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India*.
53. Balaraju, J., Raj, M. G., & Murthy, C. S. (2019). Fuzzy-FMEA risk evaluation approach for LHD machine–A case study. *Journal of Sustainable Mining*, *18*(4), 257-268.
54. Thirumoorthi, P., Deepika, S., & Yadaiah, N. (2014, March). Solar energy based dynamic sag compensator. In *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)* (pp. 1-6). IEEE.
55. Vinayasree, P., & Reddy, A. M. (2025). A Reliable and Secure Permissioned Blockchain-Assisted Data Transfer Mechanism in Healthcare-Based Cyber-Physical Systems. *Concurrency and Computation: Practice and Experience*, *37*(3), e8378.
56. Acharjee, P. B., Kumar, M., Krishna, G., Raminenei, K., Ibrahim, R. K., & Alazzam, M. B. (2023, May). Securing International Law Against Cyber Attacks through Blockchain Integration. In *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)* (pp. 2676-2681). IEEE.
57. Ramineni, K., Reddy, L. K. K., Ramana, T. V., & Rajesh, V. (2023, July). Classification of Skin Cancer Using Integrated Methodology. In *International Conference on Data Science and Applications* (pp. 105-118). Singapore: Springer Nature Singapore.

58. LAASSIRI, J., EL HAJJI, S. A. Ï. D., BOUHDADI, M., AOUDE, M. A., JAGADISH, H. P., LOHIT, M. K., ... & KHOLLADI, M. (2010). Specifying Behavioral Concepts by engineering language of RM-ODP. *Journal of Theoretical and Applied Information Technology*, *15*(1).
59. Prasad, D. V. R., & Mohanji, Y. K. V. (2021). FACE RECOGNITION-BASED LECTURE ATTENDANCE SYSTEM: A SURVEY PAPER. *Elementary Education Online*, *20*(4), 1245-1245.
60. Dasu, V. R. P., & Gujjari, B. (2015). Technology-Enhanced Learning Through ICT Tools Using Aakash Tablet. In *Proceedings of the International Conference on Transformations in Engineering Education: ICTIEE 2014* (pp. 203-216). Springer India.
61. Reddy, A. M., Reddy, K. S., Jayaram, M., Venkata Maha Lakshmi, N., Aluvalu, R., Mahesh, T. R., ... & Stalin Alex, D. (2022). An efficient multilevel thresholding scheme for heart image segmentation using a hybrid generalized adversarial network. *Journal of Sensors*, *2022*(1), 4093658.
62. Srinivasa Reddy, K., Suneela, B., Inthiyaz, S., Hasane Ahammad, S., Kumar, G. N. S., & Mallikarjuna Reddy, A. (2019). Texture filtration module under stabilization via random forest optimization methodology. *International Journal of Advanced Trends in Computer Science and Engineering*, *8*(3), 458-469.
63. Ramakrishna, C., Kumar, G. K., Reddy, A. M., & Ravi, P. (2018). A Survey on various IoT Attacks and its Countermeasures. *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, *5*(4), 143-150.
64. Sirisha, G., & Reddy, A. M. (2018, September). Smart healthcare analysis and therapy for voice disorder using cloud and edge computing. In *2018 4th international conference on applied and theoretical computing and communication technology (iCATccT)* (pp. 103-106). IEEE.
65. Reddy, A. M., Yarlagadda, S., & Akkinen, H. (2021). An extensive analytical approach on human resources using random forest algorithm. *arXiv preprint arXiv:2105.07855*.
66. Kumar, G. N., Bhavanam, S. N., & Midasala, V. (2014). Image Hiding in a Video-based on DWT & LSB Algorithm. In *ICPVS Conference*.
67. Naveen Kumar, G. S., & Reddy, V. S. K. (2022). High performance algorithm for content-based video retrieval using multiple features. In *Intelligent Systems and Sustainable Computing: Proceedings of ICISSC 2021* (pp. 637-646). Singapore: Springer Nature Singapore.
68. Reddy, P. S., Kumar, G. N., Ritish, B., SaiSwetha, C., & Abhilash, K. B. (2013). Intelligent parking space detection system based on image segmentation. *Int J Sci Res Dev*, *1*(6), 1310-1312.
69. Naveen Kumar, G. S., Reddy, V. S. K., & Kumar, S. S. (2018). High-performance video retrieval based on spatio-temporal features. *Microelectronics, Electromagnetics and Telecommunications*, 433-441.
70. Kumar, G. N., & Reddy, M. A. BWT & LSB algorithm based hiding an image into a video. *IJESAT*, 170-174.
71. Lopez, S., Sarada, V., Praveen, R. V. S., Pandey, A., Khuntia, M., & Haralayya, D. B. (2024). Artificial intelligence challenges and role for sustainable education in india: Problems and prospects. *Sandeep Lopez, Vani Sarada, RVS Praveen, Anita Pandey, Monalisa Khuntia, Bhadrappa Haralayya (2024) Artificial Intelligence Challenges and Role for Sustainable Education in India: Problems and Prospects. Library Progress International*, *44*(3), 18261-18271.
72. Yamuna, V., Praveen, R. V. S., Sathya, R., Dhivva, M., Lidiya, R., & Sowmiya, P. (2024, October). Integrating AI for Improved Brain Tumor Detection and Classification. In *2024 4th International Conference on Sustainable Expert Systems (ICSES)* (pp. 1603-1609). IEEE.
73. Kumar, N., Kurkute, S. L., Kalpana, V., Karuppannan, A., Praveen, R. V. S., & Mishra, S. (2024, August). Modelling and Evaluation of Li-ion Battery Performance Based on the Electric Vehicle Tiled Tests using Kalman Filter-GBDT Approach. In *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)* (pp. 1-6). IEEE.
74. Sharma, S., Vij, S., Praveen, R. V. S., Srinivasan, S., Yadav, D. K., & VS, R. K. (2024, October). Stress Prediction in Higher Education Students Using Psychometric Assessments and AOA-CNN-XGBoost Models. In *2024 4th International Conference on Sustainable Expert Systems (ICSES)* (pp. 1631-1636). IEEE.
75. Anuprathibha, T., Praveen, R. V. S., Sukumar, P., Suganthi, G., & Ravichandran, T. (2024, October). Enhancing Fake Review Detection: A Hierarchical Graph Attention Network Approach Using Text and Ratings. In *2024 Global Conference on Communications and Information Technologies (GCCIT)* (pp. 1-5). IEEE.

76. Shinkar, A. R., Joshi, D., Praveen, R. V. S., Rajesh, Y., & Singh, D. (2024, December). Intelligent solar energy harvesting and management in IoT nodes using deep self-organizing maps. In *2024 International Conference on Emerging Research in Computational Science (ICERCS)* (pp. 1-6). IEEE.
77. Praveen, R. V. S., Hemavathi, U., Sathya, R., Siddiq, A. A., Sanjay, M. G., & Gowdish, S. (2024, October). AI Powered Plant Identification and Plant Disease Classification System. In *2024 4th International Conference on Sustainable Expert Systems (ICSES)* (pp. 1610-1616). IEEE.
78. Dhivya, R., Sagili, S. R., Praveen, R. V. S., VamsiLala, P. N. V., Sangeetha, A., & Suchithra, B. (2024, December). Predictive Modelling of Osteoporosis using Machine Learning Algorithms. In *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)* (pp. 997-1002). IEEE.
79. Kemmannu, P. K., Praveen, R. V. S., Saravanan, B., Amshavalli, M., & Banupriya, V. (2024, December). Enhancing Sustainable Agriculture Through Smart Architecture: An Adaptive Neuro-Fuzzy Inference System with XGBoost Model. In *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)* (pp. 724-730). IEEE.
80. Praveen, R. V. S. (2024). *Data Engineering for Modern Applications*. Addition Publishing House.