International Journal of Computer Sciences and Engineering
Vol.**11**, Issue.**12**, pp.**26-31**, December **2023**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

JCSE
ISSN: 2347-2693 (E)

Research Paper

Pothole Detection and Reporting System Implementation Using Yolov8 and TensorFlow.js

Nidhi Ruhil[1], Devansh Sahni[2*], Anushaka[3], Anurag Wadhwa[4], Anjali Sharma[5]

[1,2,3,4,5]Dept. of Computer Science and Engineering, Dr Akhilesh Das Gupta Institute of Professional Studies (Previously ADGITM), New Delhi, India

**Corresponding Author: devanshsahni@gmail.com*

Received: 08/Nov/2023; **Accepted**: 11/Dec/2023; **Published**: 31/Dec/2023. **DOI**: https://doi.org/10.26438/ijcse/v11i12.2631

Abstract: Potholes present a substantial hazard to both road safety and the structural integrity of vehicles. This paper introduces a novel approach to pothole detection leveraging YOLOv8, an object detection algorithm, and TensorFlow.js. The proposed system aims to detect potholes accurately and swiftly by analysing live video feeds. The trained model exhibits promising performance metrics in pothole detection, with the bounding box precision at 0.822 and the mean Average Precision (mAP) value of 0.847, highlighting the model's robustness in accurately localizing potholes. The proposed pothole detection system presents a promising solution for proactive road maintenance and safety enhancement. Its efficiency in real-time detection, combined with the adaptability of TensorFlow.js, holds the potential for widespread implementation, contributing to mitigating road hazards and infrastructure maintenance. The use of Tensorflow.js allows JavaScript developers to work with YOLOv8 reducing the dependency on Python for this purpose. The Pothole Detection and Reporting System with YOLOv8 and Tensorflow.js provides quite promising results.

Keywords: Object detection, YOLOv8, TensorFlow.js, Road safety, Pothole detection.

1. Introduction

Road infrastructure forms the backbone of transportation networks, serving as vital conduits for societal mobility and economic activity. However, the continuous wear and tear experienced by roads results in a persistent challenge manifested prominently through potholes. These seemingly small surface irregularities carry substantial implications, disrupting the functionality of roadways and posing multifaceted risks. 5,000+ deaths were caused by road accidents due to potholes between 2018 and 2020. In 2021, potholes were one of the major reasons for over 3600 accidents in India.

The field of pothole detection has seen extensive studies employing manual detections, sensor-based systems, YOLO models, and other object detection techniques. However, these investigations predominantly emphasize object identification, lacking effective user-friendly reporting features. Current models and applications suffer from limited user interaction, hindering the smooth reporting of identified potholes. This research presents a pioneering direction by integrating YOLOv8 into a web-based interface using TensorFlow.js. This novel approach aims not only to detect potholes in real time but also to streamline the reporting process for users. By focusing on both detection and user-friendly reporting, this study aims to bridge the gap between detection models and simplified reporting for improved road safety and maintenance.

1.1 Proposed Approach

In response to this challenge, our innovative approach involves crafting a user-centric web application to detect potholes on roads, employing the power of YOLOv8 and TensorFlow.js. This unique integration enables immediate pothole detection within web browsers, eliminating the necessity for users to download and install a separate app. Our solution is swift, precise, and cost-effective, utilizing smartphone cameras and GPS for seamless real-time pothole identification. We aim to establish a seamless system for detecting potholes from live video frames, facilitating efficient reporting and proactive road maintenance.

1.2 Motivation

Motivation behind the project lies in addressing issues of road infrastructure and road safety due to the presence of potholes. This paper aims to address the pressing need for an accessible and effective pothole detection and reporting system, emphasizing the significance of a web-based application in streamlining this process. The selection of TensorFlow.js and YOLOv8 stems from their distinct advantages. TensorFlow.js ensures seamless accessibility through web-based deployment, eliminating the necessity for separate installations and aligning perfectly with the project's aim of inclusivity. YOLOv8, recognized for its precision and rapid object detection capabilities, serves as an ideal choice to power the backend of this web application, enabling real-time pothole identification.

The remaining sections of the paper are structured as outlined below: Section 1 encompasses the introduction which presents an overview of road infrastructure challenges and introduces the proposed solution for real-time pothole detection. Section 2 includes the related work which delves into previous studies on pothole detection methodologies, exploring traditional and machine learning-based approaches. Section 3 contains the measures undertaken for dataset selection, preprocessing, and augmentation techniques. It also contains the essential steps for building the system's design including frontend interface, backend setup, image processing and database integration. Section 4 describes the results and discussion including the performance metrics, real-time video stream processing, and geolocation integration and discusses the system advantages, challenges and future enhancements. Section 5 summarizes the research work and outlines potential future directions.

2. Related Work

Previous studies have extensively explored pothole detection methodologies, ranging from traditional image processing techniques to advanced machine learning models.

Detection of potholes has seen advancements primarily through models like YOLO and Faster R-CNN, proving effective in identifying these road imperfections in images and video frames. One study [5] proposed an economical sensor-driven method utilizing CNN; however, it struggled to detect potholes accurately under varying light conditions. Other strategies [6, 7] rely on sensor-based techniques. For instance, research [6] employed smartphone-based accelerometers and GPS for road damage identification, achieving high accuracy but not specifically targeting potholes. Similarly, Piao and Aihara [7] implemented a sensor-based system utilizing accelerometers and GPS, yet faced challenges due to its physical sensor-based approach compared to digital methodologies.

The examination in [10] delves into detecting potholes using vision-based (images), vibration-based (accelerometer data), and 3D reconstruction (stereo-vision), outlining their advantages—like 3D reconstruction's precision and vibration-based methods' cost-effectiveness and drawbacks, such as vulnerability to lighting disparities in vision-based techniques. The study in [9] investigates various methods for detecting potholes and assessing their effectiveness: vibration-based, 3D reconstruction, and vision-based approaches. It highlights the precision of vision-based techniques, discussing their advantages and constraints. The paper explores smartphone-based detection, laser scanning, and RGB analysis, outlining progress and hurdles in identifying road potholes.

The research in [11] suggests utilizing mobile devices that come with G-sensors and GPS sensors to gather accelerometer data and location details. The process includes standardizing accelerometer data by employing Euler angles, identifying potholes using different methods such as Z-THRESH, Z-DIFF, STDEV(Z), and G-ZERO, and pinpointing exact pothole locations through spatial interpolation using GPS data.

The approach described in reference [8] explored IoT-enabled pothole detection and reporting systems utilizing Arduino microcontrollers and sensors within a vehicle setup. This system aimed to identify and report potholes and road obstacles by activating sensors and cameras based on the vehicle's motion, subsequently transmitting relevant data to a server. However, the sensor-based method exhibits shortcomings in accurately identifying and precisely locating potholes when compared to the capabilities of YOLOv8. Moreover, Arduino-based systems are constrained in terms of scalability and flexibility, limiting their use across diverse devices due to specific hardware requirements.

The study in [3] proposes a hands-free detection of potholes by observing the accelerometer readings on the phone while the user is travelling. Despite its advantages such as real-time processing and minimal storage needs, this method has inherent drawbacks. It may produce inaccuracies by identifying road joints as potholes, and it might not detect potholes located in the centre of the road without vehicular impact.

Research Paper [4] uses the older version of the YOLO model for detecting potholes with less accuracy than the YOLOv8 model results.It allows the user to click images and submit them to detect potholes. There is no real-time detection. [6] It uses an app for the detection of potholes. However, this poses challenges of low user adoption due to specific use cases, potential storage constraints, and ongoing maintenance burdens like updates and compatibility. A web-based solution can bypass these issues by offering immediate accessibility, eliminating app downloads, and ensuring broader device compatibility without storage limitations.

3. Experimental Method

3.1. Dataset:

In this study, the dataset used for pothole detection is sourced from Kaggle, specifically the "Pothole Detection Dataset", provided by Raj Dalsaniya. This dataset offers a comprehensive collection of images where potholes are annotated in YOLO v7 PyTorch format. The images are captured at different scales, backgrounds, and weather conditions. Additionally, the dataset contains images of water-filled potholes, ensuring that the model is trained to recognize a variety of scenarios.

The dataset consists of a total of 2105 images depicting pothole-ridden road surfaces. Each image in the dataset was resized to 640x640 (Stretch) along with that data augmentation was performed to increase the dataset's size and diversity, which proved beneficial given the uncertainty in weather conditions, camera mounting, and image quality variability.

3.2. Training on Colab and Preparation:

Collaboratory, or Colab, is a creation of Google Research, offering a Linux-based environment with a user interface modelled on the Jupyter Notebook service. Colab provides complimentary access to substantial computing resources, including Graphical Processing Units (GPUs). It is widely employed for developing deep learning applications, leveraging popular libraries like TensorFlow, PyTorch, and OpenCV.

For our deep learning project, we conducted training and validation on Google Colab using a Tesla T4 GPU. The Tesla T4 GPU is recognized for its significant speed advantage over Central Processing Units (CPUs) in the context of deep learning applications. Following the training phase, we exported the model in TensorFlow.js (tfjs) format, as it was intended for utilization in TensorFlow.js.

The figure below shows the detected potholes along with their confidence scores written above the bounding boxes. For this case, the threshold confidence score was 0.45.

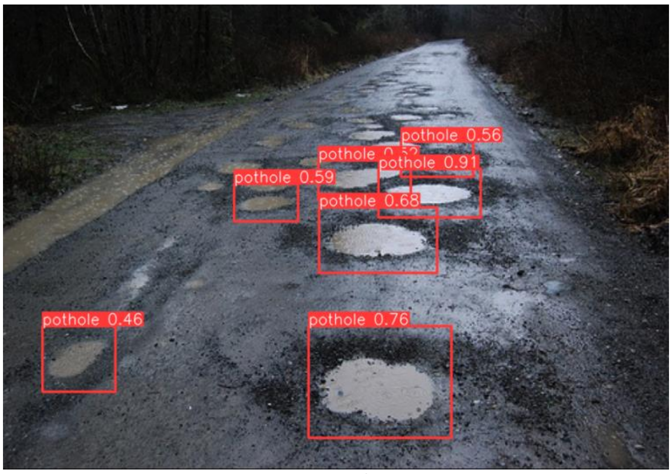

Figure 1. Detected potholes from the trained model

3.3. User Interface and Interaction:

The front end, developed using React, provides a user-friendly interface enabling individuals to utilize their devices' cameras or webcams for capturing the frames. Utilizing navigator.mediaDevices for camera access and navigator.geolocation for obtaining latitude and longitude, the system becomes a dynamic portal for continuous frame capture. Here, two canvases are created, one visible and one hidden canvas, adjusting their dimensions to match the video stream. The function captures the current video frame, drawn onto the hidden canvas, and converts it to a JPEG image. This image is encoded in base64 format for seamless transmission. The encoded image along with geographical coordinates, is sent to a server endpoint for further processing. The server responds with information about detected potholes, and if any are found, the visible canvas is updated to display red bounding boxes around the identified potholes. These bounding boxes are drawn based on the coordinates received from the server.
Upon successful detection of the pothole, the capturing and transmission of frames are halted.

3.4. Preprocessing:

Commencing the intricate process with frames encoded in the base64 format, the preliminary steps encompass decoding and resizing to adhere to the model's preferred input dimensions of 640x640 (Stretch). This deliberate augmentation serves to significantly elevate the overall performance of the model. Following the resizing phase, the frames undergo a meticulous transformation into tensors, a pivotal stage aimed at ensuring perfect alignment with the model's distinct input criteria. To achieve a state of harmonious compatibility, a careful normalization process is executed, methodically preparing the frames for seamless integration into the sophisticated architecture of the YOLOv8 model.

3.5. Post Processing:

The output from the model undergoes two crucial steps of Confidence score filtering and Intersection-over-Union (IOU) application. The output is filtered to retain only those bounding boxes that surpass a threshold confidence score of 50%. This step ensures that only highly probable potholes are considered for further analysis. Further, the Intersection over Union (IoU) is calculated by determining the overlap between two bounding boxes. First, the coordinates of the intersection rectangle are found by identifying the maximum values of the top-left corner (x1, y1) and the minimum values of the bottom-right corner (x2, y2) between the two boxes. The area of this intersection rectangle is computed. The IoU is then obtained by dividing the area of the intersection by the sum of the areas of the two bounding boxes minus the intersection area. This ratio provides a measure of the relative overlap between the two bounding boxes, with a higher IoU indicating greater agreement and overlap. If the overlap exceeds a certain threshold of 70%, the boxes are considered to represent the same pothole. This suppression mechanism ensures that only distinct and non-intersecting potholes are included in the final count. Finally, the function returns the selected and filtered bounding boxes as an array to the front end which is used for real-time detection of potholes.

3.6. Database Integration:

To ensure enduring and substantive influence, this project seamlessly integrates MongoDB for the streamlined storage and retrieval of identified pothole coordinates, encompassing both latitude and longitude details, along with a comprehensive count of detected potholes. This pivotal feature not only expedites the prompt reporting of potholes in real time but also establishes a robust foundation for prospective initiatives. By creating a repository of vital information, including pothole locations and their frequency, the system contributes significantly to future road reconstruction endeavours and strategic planning.

4. Results and Discussion

4.1. Performance Metrics

The following metrics were used for evaluating the model:

Mean Average Precision(mAP): It is a widely accepted performance metric for object detection models. mAP is calculated by taking the mean of Average Precision (AP) for each of the 'n' classes. AP for each class 'k' is determined by calculating the area under the precision-recall curve. mAP provides a single score that considers Recall, Precision, and Intersection over Union (IoU), eliminating bias in performance measurement.

Processing Time: Processing time is a crucial metric for assessing the speed at which the model classifies an input image. It encompasses the total time taken by the model for pre-processing, inference, loss calculation, and postprocessing of an image. Swift decision-making is essential to detect potholes early, making the reporting and repairing better and swift.

F1-Confidence Score Analysis: The depicted curve illustrates the model's optimal F1 score for pothole detection, indicating peak performance at a confidence level of approximately 0.5. This signifies the model's heightened effectiveness in pothole identification when maintaining a moderate confidence threshold. Deviations towards excessively low or high confidence levels may result in missed potholes or erroneous detections. Additionally, the graph highlights an average F1 score of 0.81 across all classes when the confidence level is precisely 0.499. This analysis underscores the critical relationship between confidence levels and the model's precision in pothole detection which is shown in figure 2 below.

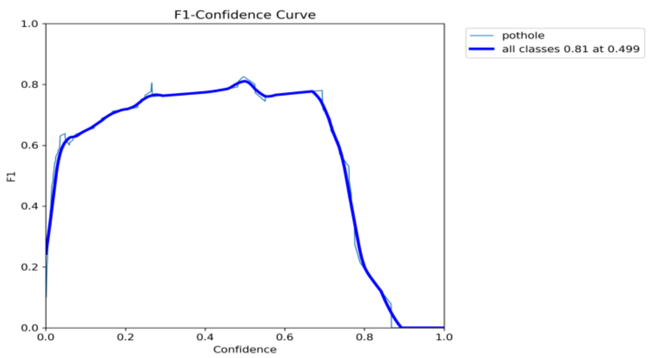

Figure 2. F1 score of the model

4.2. Results

Model Performance Metrics: The proposed pothole detection system leverages the YOLOv8s.pt model trained on a dataset from Kaggle comprising 2107 pothole images. The model demonstrated commendable performance, as evidenced by the following metrics in Table 1:

Table 1. Model metrics

Precision (P)	0.822
Recall (R)	0.76
mAP(mean Average Precision)	0.847
mAP50-95	0.372

Precision(P), with a value of 0.822, represents the accuracy of the model in correctly identifying potholes among the instances it predicts. A higher precision indicates a lower rate of false positives, signifying the model's proficiency in precise pothole detection. Recall(R), denoted by a value of 0.76, signifies the model's ability to capture and identify all instances of potholes present in the dataset. A higher recall value suggests a lower rate of false negatives, indicating the model's effectiveness in comprehensive pothole recognition. mAP (mean Average Precision) with a value of 0.847, mAP provides an overall assessment of the model's precision across different confidence thresholds. This metric considers the precision-recall trade-off and provides a comprehensive evaluation of the model's performance. mAP50-95, with a value of 0.372, specifies the mean Average Precision over a specific confidence interval (from 50% to 95%). It offers a nuanced evaluation, considering a range of confidence levels, providing insights into the model's robustness under varying thresholds.

These metrics collectively depict the model's accuracy, completeness, and overall performance in detecting and recognizing potholes, crucial for validating and understanding the model's capabilities.

Inference Speed: The expeditious inference speed of our model assumes paramount importance in real-time applications. Demonstrating commendable efficiency, our system achieved an average processing time of 0.2ms for preprocessing, 4.3ms for inference, and 0.9ms for post-processing per image. This accelerated processing capability plays a pivotal role in swift and accurate pothole detection, significantly contributing to the reduction of overall system response time.

Non-Max Suppression: This technique is used to check overlapping boundary boxes. To check if two boxes overlap or not, we calculate the Intersection Over Union (IOU) and compare it with the Threshold value set by us.

Box A and Box B are two distinct potholes with their respective diagonal coordinates. Below we calculate the length and height of the intersecting box to find the intersection area.

L represents Length and H represents Height

Box A: ($x_1^{(A)}$, $y_1^{(A)}$), ($x_2^{(A)}$, $y_2^{(A)}$)
Box B: ($x_1^{(B)}$, $y_1^{(B)}$), ($x_2^{(B)}$, $y_2^{(B)}$)

$$L = min(x2(A), x2(B)) - max(x1(A), x1(B)) \quad (1)$$

$$H = min(y2(A), y2(B)) - max(y1(A), y1(B)) \quad (2)$$

$$Intersection\ Area(IA) = L * H \quad (3)$$

$$UnionArea = AreaBoxA + AreaBoxB - IA \quad (4)$$

$$IOU = (Intersection\ Area) / (Union\ Area) \quad (5)$$

If IOU is greater than or equal to the Threshold (0.7 in our case), we take only one box with the higher probability. If it is smaller than the Threshold, then we take both boxes.

Real-time Video Stream Processing: The system seamlessly processes real-time video streams from user webcams. Frames are continuously transmitted to the backend, undergo preprocessing, and are fed into the YOLOv8 model for pothole detection. The efficiency of our approach is further highlighted by its ability to operate on diverse devices. The smooth transmission of frames was key to making our system more efficient and friendly. The technique utilized in this system guarantees the transmission of high-quality frames to the model, enhancing processing capabilities.

Geolocation Integration: To augment the system's functionality, the incorporation of geolocation coordinates has been implemented. This integration facilitates the extraction of precise location data from user devices, accompanying each frame. The inclusion of latitude and longitude coordinates in the system not only ensures accurate detection and reporting of pothole locations but also establishes a seamless link for real-time communication with municipal authorities. The recorded coordinates are stored in the database, paving the way for a responsive approach to maintenance, tailored to the specific locations of identified potholes. This strategic integration marks a significant advancement in optimizing location specific repair strategies.

4.3. Discussion

Data Limitations for Indian Roads: The existing dataset for training and validating the model on Indian roads, specifically concerning potholes, is notably restricted. This constraint results in models having less exposure to the unique conditions of Indian roads, known for their high traffic volume and distinct characteristics compared to roads in other countries. With a more extensive and high-quality dataset, the potential exists to develop a model finely tuned to the intricacies of Indian roads, ultimately leading to elevated precision and detection rates in practical, real-world situations.

Advantages and Future Applications: The commendable success achieved by our system in real-time pothole detection unfolds a multitude of advantages. The prompt response time for authorities ensures swift pothole repairs, contributing to a substantial reduction in road accidents and vehicle damage. The utilization of TensorFlow.js not only facilitates the deployment of web-based applications but also enhances accessibility and user-friendliness, eliminating the necessity for additional software installations and lowering pothole detection costs. While acknowledging challenges such as scalability and privacy concerns, the potential applications extend to diverse settings, including roads, highways, and densely populated urban areas.

Challenges and Future Enhancements: Challenges and Future Enhancements: The ongoing evolution of our system involves addressing challenges related to scaling, particularly in deploying road cameras extensively and navigating privacy concerns. Future enhancements will delve into innovative solutions, ensuring the adaptability of the system for broader deployment and aligning with the dynamic requirements of evolving urban landscapes.

In conclusion, the compelling results presented in this study underscore the effectiveness of our pothole detection system. This success not only establishes a solid foundation but also paves the way for future advancements and applications within the realm of intelligent transportation systems.

5. Conclusion and Future Scope

This research has presented a robust and efficient pothole detection system built upon the foundation of the YOLOv8 object detection algorithm and facilitated by TensorFlow.js for seamless deployment. The investigation focused on addressing the pressing need for proactive road maintenance by employing advanced computer vision techniques to detect and localize potholes in real time. The experimental results showcased the system's commendable performance, characterized by a recall rate of 0.76, the mean average precision metrics, notably mAP@50 of 0.847, along with a bounding box precision of 0.822, demonstrated the model's ability to precisely detect and localize potholes with varying degrees of stringency. The significance of these findings lies in the potential practical applications of the developed system. Its efficiency in real-time pothole detection, coupled with the portability and versatility offered by TensorFlow.js, presents opportunities for widespread deployment. This system could revolutionize road monitoring strategies, facilitating timely repairs and preemptive maintenance to ensure safer and more resilient infrastructure.

The future scope for the Pothole Detection and Reporting System includes integrating an API for providing the coordinates of the detected potholes ensuring an optimized and pothole-free route, and updating and retraining the model at certain periods to adapt to changing conditions of road. This system looks forward to working on building a Road Quality Index based on factors like potholes, surface conditions, and overall infrastructural health. The system can be developed to handle diverse datasets, considering different road types, materials and construction standards across cities. This ensures that the machine learning model can be trained and fine-tuned effectively for different scenarios.

Conflict of Interest:
None

Funding Source:
None

Authors' Contributions
The authors collectively engaged in a comprehensive review of relevant literature and analyzed various research papers about the subject matter to formulate the foundation for this study. Furthermore, the participation of all authors in the thorough review and editing process of the manuscript,

ultimately providing their approval for the final version of the document.

Acknowledgements:
None

References

[1] Anas Al Shaghouri, Rami Alkhatib, Samir Berjaoui, "Real-Time Pothole Detection Using Deep Learning," **2021.** https://doi.org/10.48550/arXiv.2107.06356

[2] O. M. Khare, S. Gandhi, A. M. Rahalkar, S. Mane, "YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes," **2023.**

[3] S. R. Kuthyar, R. S., V. Rasika, S. Manjesh, R. Girimaji, R. Davis Arjun, P. S., "An Intelligent Pothole Detection and Alerting System using Mobile Sensors and Deep Learning," **2021.**

[4] R. Hiremath, K. Malshikare, M. Mahajan, "A Smart App for Pothole Detection Using Yolo Model," 2021.

[5] A. Kumar, S. Kumar, "Road quality management using mobile sensing," International Conference on Innovations in Information Embedded and Communication Systems, **2018.**

[6] A. Mendis, G. Starzadnis, G. K. Anoris, "Real-time pothole detection using Android smartphones and accelerometers," 12th International Conference on ITS Telecommunications (ITST), pp. 668-672, 2019.

[7] B. Piao, K. Aihara, "Detecting the road surface condition by using mobile crowd sensing with a drive recorder," IEEE 20th International Conference on Intelligent Transportation Systems, **2019.**

[8] Collinson Colin M. Agbesi, Ebenezer K. Gavua, Seth Okyere-Dankwa, Kwame Anim Appiah, Kofi Adu-Manu Sarpong, “Pothole Detection, Reporting and Management using Internet of Things: Prospects and Challenges”, International Journal of Emerging Science and Engineering (IJESE), **2017.**

[9] Jetashri R. Gandhi, U. K. Jaliya, D. G. Thakore, “A Review Paper on Pothole Detection Methods”, International Journal of Computer Sciences and Engineering (IJCSE), **2019**

[10] Young-Mok Kim, Young-Gil Kim, Seung-Yong Son, Soo-Yeon Lim, Bong-Yeol Choi and Doo-Hyun Choi, “Review of Recent Automated Pothole-Detection Methods”, Applied Sciences, **2022.**

[11] Hsiu-Wen Wang, Chi-Hua Chen,Ding-Yuan Cheng, Chun-Hao Lin and Chi-Chun Lo, “A Real-Time Pothole Detection Approach for Intelligent Transportation System”, ResearchGate, **2015.**

AUTHORS’ PROFILE

Ms. Nidhi Ruhil is currently working as an Assistant Professor at the Dr. Akhilesh Das Gupta Institute of Professional Studies (previously ADGITM), which is affiliated with Guru Gobind Singh Indraprastha University in Delhi, India. She is employed in the Computer Science & Engineering department. A+ rated bachelor's degree in computer science and engineering from Maharishi Dayanand University in Rohtak was given in 2012. She graduated from ITM University in Gurugram with a master's degree in computer science & engineering with an A+ in 2014. She has presented at numerous international conferences and had books and research articles published in renowned magazines. Her areas of interest include networking, IoT, artificial intelligence, and information and communication technology.

Mr. Devansh Sahni is a dedicated B.Tech. student majoring in Computer Science and Engineering at Guru Gobind Singh Indraprastha University, anticipated to graduate in 2024. A profound interest and commitment to technology and web development marks his academic journey. Devansh's focus on original and innovative projects reflects his passion for contributing valuable insights to the field. With a keen eye for impactful work, he seeks to make meaningful contributions that have the potential to benefit others.

Ms. Anushaka, a devoted B.Tech in CSE student at Dr. Akhilesh Das Institute of Professional Studies, affiliated with Guru Gobind Singh Indraprastha University, passionately explores the dynamic realm of computer science, with a particular focus on web development. Her academic journey reflects a keen enthusiasm for staying abreast of new technological trends and embracing the latest advancements in the field. Known for her natural adaptability to diverse environments, Anushaka excels in collaborative group projects, showcasing both teamwork and versatility in navigating complex challenges

Mr. Anurag Wadhwa, currently pursuing Bachelors of Technology in Computer Science and Engineering at Guru Gobind Singh Indraprastha University, Delhi, India, is a passionate web developer. He is dedicated to mastering the art of technology, combining academic rigour with practical application. Eager to delve deeper into the world of software engineering, Anurag demonstrates a keen interest in leveraging. His commitment to staying updated with the latest advancements in programming languages and frameworks underscores his determination to excel in the dynamic field of web development.

Ms. Anjali Sharma is presently dedicated to her pursuit of a Bachelor's in Computer Science Engineering at Dr. Akhilesh Das Gupta Institute of Professional Studies, affiliated with Guru Gobind Singh Indraprastha University in Delhi. With a profound interest in web development and technology, she harbours a deep passion for delving into the ever-evolving domains of innovation within the computing sphere. Anjali remains committed to harnessing her expertise and insights to drive meaningful contributions to the ongoing advancements in this dynamic field.

