

SwachhSeva an AI Assisted Mobile System for Civic Issue Reporting and Intelligent Complaint Management

Rashmi C. G.¹; Mrunali G. M.²; Meghana H. Huvanur³;
Punya P. Y.⁴; Pawanraj S. P.⁵

^{1,2,3,4}Department of Computer Science and Engineering, East West Institute of Technology, Bengaluru.

⁵Faculty of Computer Science and Engineering, East West Institute of Technology, Bengaluru.

Publication Date: 2026/10/02

Abstract: There is a lot of work to do in cities across India when it comes to fixing problems like potholes, garbage, broken streetlights and blocked drains. Now most of the work is done by hand. This can be slow. It is hard to keep track of the problems. It is hard to say who is responsible. It is hard to let people know what is happening. To fix these problems this paper introduces SwachhSeva. SwachhSeva is a way to report and manage city problems. It connects people with workers through a mobile app and a website. The system has an app for people. It uses React Native. There is a website for workers. It uses React. There is an end that uses Flask. People can send pictures and their location when they report a problem. Workers can see the reports. They can sort them. They can give them to the person. They can fix them. A special model called MobileNetV2 helps sort the problems. It uses a method called transfer learning. If the model is not sure the problem goes to a person instead. The system checks for repeated reports. It uses a formula called Haversine. This helps people see if others have already reported the problem. They can upvote of sending the same report again. A system gives scores to problems that need more attention. It looks at how many people upvoted. It looks at how old the problem's. It looks at deadlines. This helps workers know which problems to fix first. The system also sends problems to the department. It keeps track of deadlines. A map shows where the problems are. This helps bosses see where the most problems are. It helps them decide where to send help. Testing the model worked well on the data used.. The results depend on how much data there is. SwachhSeva is a way to handle city problems. It is not expensive. It can be better, with real data and better maps.

Keywords: Civic Complaint Management, MobileNetV2, Transfer Learning, Image Classification, Duplicate Detection, Geolocation, SLA Management, Citizen Grievance Redressal, Human-in-the-Loop AI, Priority Scoring.

How to Cite: Rashmi C. G.; Mrunali G. M.; Meghana H. Huvanur; Punya P. Y.; Pawanraj S. P. (2026) SwachhSeva an AI Assisted Mobile System for Civic Issue Reporting and Intelligent Complaint Management. *International Journal of Innovative Science and Research Technology*, 11(9), 2454-2465. <https://doi.org/10.38124/ijisrt/26sep1065>

I. INTRODUCTION

The rapid growth of populations has brought many civic problems that touch the daily lives of people. Potholes, piles of garbage, broken streetlamps, blocked drains, and worn footpaths are common in cities. There is a need for quick help from local officials. These problems may seem small on their own. When they keep happening, they can harm safety, cleanliness, traffic, and the overall feel of city life. So, finding and fixing these problems fast is crucial for a safe and well - run city.

Old ways of handling complaints rely on handwritten reports and paperwork. Citizens might have to go to an office, call a number, or fill out a web form. That makes it hard for people to give locations, follow their complaint, or

know which department will deal with it. For officials sorting through complaints, putting them in the right category, assigning them, and watching them finish can be slow. As complaints grow, the work takes longer, creating duplicates. Urgent matters may get lost.

Smartphones are everywhere. They offer a chance to improve how complaints are handled. Today's phones can take photos get GPS coordinates, and send data online. A mobile app for reporting problems lets citizens attach a picture and the exact spot. That gives city workers useful information compared to a plain text note and helps them spot and fix issues faster.

Adding intelligence can make the system even better by cutting down on manual sorting. Image techniques can

analyze the photos people send and identify which type of problem it shows. In this study, we use a MobileNetV2 network, trained on existing data, to classify civic photos into set categories. Automatic classification helps organize complaints and send them to the team. The computer can still get it wrong, especially if the picture is blurry or does not match a known category. So, the system also lets a human review cases with confidence or no prediction.

Another problem is duplicates in complaints. Many people might report the pothole, trash heap, or broken light separately. While community reporting is good, repeating the issue can overload staff and clutter records. SwachhSeva solves this with location - based detection. The app checks a complaint's GPS against nearby open complaints and flags possible duplicates. Citizens can then upvote an existing report instead of making a new one.

Prioritizing complaints matters because city teams often get reports all at once but have limited time. Not every complaint can be handled immediately. A complaint that many people support or that has stayed open longer than promised may need attention. SwachhSeva, therefore, scores complaints using a weighted system that looks at upvotes, how old the complaint is, and whether it meets the service - level agreement. This extra tool helps officers spot complaints that must be addressed soon.

Location data can also reveal patterns beyond single complaints. If many issues spread across a city, showing them as isolated pins can hide trends. SwachhSeva adds a heatmap to the admin dashboard, coloring areas with reported problems. This view lets managers spot recurring trouble areas and plan where to send crews. The heatmap provides a view of city problems instead of just individual marks.

To tackle these challenges, we present SwachhSeva, a system that links citizens and city officers through a mobile and web platform. The app, on phones, is built with React Native; the server runs a Flask REST API; data lives in PostgreSQL; and the admin page is made with React. SwachhSeva pulls together image classification, GPS, duplicate detection, community voting, priority scoring, routing by department, service - level tracking, and heatmaps into one workflow.

The goal of SwachhSeva is to cut back on handwritten complaints while boosting openness, accountability, and speed in work. By mixing AI, mobile reporting, and map analysis, SwachhSeva offers a way to help city teams handle complaints and lets citizens see how their reports move forward. SwachhSeva is built to grow, so we can add data, new problem types, and better maps and alerts in the future.

The main contributions of this work are as follows:

We extend real-time geospatial duplicate detection into a recurring-hotspot detection mechanism that analyzes a locations complaint history, including already-resolved complaints to distinguish systemic infrastructure problems

from isolated incidents. A capability absent from the civic complaint management systems reviewed in this study.

We introduce a citizen dispute and reopening mechanism that is directly coupled to the priority-scoring system automatically elevating a complaints priority when a citizen contests its resolution closing an accountability loop that remains one-directional in systems.

We design a citizen engagement scoring mechanism that rewards outcome-validated participation. Reports that are actually resolved and upvotes on complaints later confirmed genuine. Than raw submission or upvote volume discouraging low-quality or spam reporting.

We propose a linearly weighted priority-scoring formula in place of an opaque confidence-based or learned ranking mechanism allowing municipal officers to audit and understand exactly why one complaint is prioritized over another.

We integrate image-based classification, geospatial duplicate detection, prioritization, department routing with SLA tracking, recurring-pattern analysis and citizen accountability mechanisms into a single end-, to-end civic complaint management workflow addressing a gap identified in the reviewed literature, where existing systems typically implement only one or two of these components in isolation.

II. LITERATURE REVIEW

The rise of civic complaint management systems has drawn more attention as smart cities grow, as people use mobile devices, as artificial intelligence improves, and as maps become more detailed. I have watched how these systems grow in my city. Traditional ways of handling complaints usually ask a citizen to write down a problem. Then city workers decide what it is and who fixes it. Recent studies try to lessen this work by adding mobile apps, automatic picture checks, map tags, online dashboards, and data-driven ranking. These ideas support building a better civic complaint management system.

Many researchers have looked at ways to report civic problems. I have used these apps to report a streetlight. A phone app lets a citizen point out an issue from the spot where it happens. GPS puts the coordinates on the complaint, giving city workers more precise data than a handwritten address. Photos can be attached so the problem can be seen visually. These tools make reporting easier and cut the effort needed to file a complaint. Still, if the system only focuses on registration, city officers still have to do a lot of manual work to sort, check, assign, and decide how urgent each complaint is.

Computer vision has been added to complaint management systems to reduce the need for hand sorting. I have seen the difference when a photo automatically flags a pothole. Image classifiers and object detectors can find types of road or environmental issues from pictures sent by citizens. Studies that use learning, like convolutional neural

networks and detection algorithms, have shown that the system can spot potholes, trash, road cracks, and other visible problems. Automatic classification saves time for officers who would otherwise look at every complaint one by one. It also provides a consistent way to label each report.

Deep learning models need training data and may not work well when the pictures look very different from the training set. I understand that a model can miss a picture that's too dark. Light, camera angle, picture quality, background items, and how a problem looks can all affect the model's accuracy. In life, citizen photos are not taken in controlled labs. Therefore, a civic complaint management system should not assume every automated guess is right. A confidence score plus a human check gives a way to proceed when the model is unsure.

Transfer learning helps when the specific dataset is small. I know that transfer learning saves time for developers. Instead of building a deep network from scratch, a model that has already learned from a large image set can be fine - tuned for this task. Lightweight models like MobileNetV2 are ideal when computing power is limited. Earlier work shows transfer learning works well for image tasks, so it fits civic issue classification when there are few labeled pictures.

Geographic data is another part of the civic complaint management system. I see how GPS helps the city staff. GPS reporting ties each complaint to a spot, letting authorities see where problems cluster. Location data can assign complaints to the right department, spot reports, and find repeat problem areas. Just storing coordinates does not provide useful insight. Visual tools are needed to turn points into information that helps decision-makers.

Geospatial heatmaps are one visual tool. I have noticed how heatmaps show spots. Instead of showing every complaint as a marker, a heatmap highlights areas with complaints. This helps officials spot civic issue hotspots and see where problems repeat. Heatmaps also aid planning and resource allocation by showing complaint density. In management , this is especially useful when many complaints are spread across different parts of a city.

Detecting complaints is another challenge that recent civic reporting research has tackled. In a community - based reporting setting, many citizens may file reports for the same issue. If each report is kept as a complaint, the system can grow with duplicate entries, and officers may spend extra time on the same problem. Using proximity helps spot potential duplicates. By comparing the location of a complaint with those of open complaints, the system can flag reports that may refer to the same physical problem. Community tools such as upvoting can then show the public's concern level without creating independent records.

Citizen participation is also a part of digital civicplatforms. Features like complaint tracking, notifications, upvoting , and status updates encourage

citizens to monitor problems. Community support signals how important an issue is. However, citizen input alone does not ensure that authorities will act on complaints in order. Administrative prioritization tools are needed so that officers can manage complaints when the number of reports exceeds resources.

Many systems have added dashboards to provide municipal staff with a single place to view complaints. Web - based dashboards display complaint details, locations, categories, status, and other data in a structured layout. Such interfaces improve monitoring compared with systems that spread complaint information across channels. A dashboard becomes more useful when it offers decision - support information beyond merely showing stored records. Priority indicators, overdue alerts, geographic visualizations, and officer performance statistics help turn the dashboard into a management tool.

Managing complaints with service - level agreements keeps accountability high. Different civic problems require expected resolution times. For example, a blocked drain or a garbage pile requires action differently than some infrastructure repairs. Assigning a deadline to each complaint category allows the system to check whether complaints close within the expected window. When a complaint stays open past its deadline, it becomes overdue and is given higher priority or escalated for further attention. This structured approach monitors the performance of the complaint - resolution process.

Research shows that tools such as reporting, deep learning , GPS tagging, and dashboards each add value. Many of these tools are built as separate pieces or focus on only one part of complaint management. A complete system can benefit from combining them into a single workflow: citizen reporting, automated classification, verification, department assignment, prioritization, monitoring, resolution , and spatial analysis.

SwachhSeva is built with this integrated approach. SwachhSeva pairs a citizen - facing app with an administrative web dashboard and a Flask - based backend. Photos submitted by citizens go through a MobileNetV2 model trained with transfer learning. If the model cannot predict with confidence, SwachhSeva sends the complaint for manual review instead of relying on an uncertain prediction. SwachhSeva also uses information to find potential duplicates and offers an upvote button so citizens can support existing reports.

SwachhSeva also adds a priority that considers community upvotes, complaint age, and service - level agreement status. This means SwachhSeva takes into account a complaint's category, urgency, and community impact. Department - based routing reduces effort by sending complaints directly to the appropriate authorities. SwachhSeva provides status updates throughout the resolution process, giving both citizens and administrators clear visibility.

SwachhSeva adds a heatmap to show where many complaints cluster. SwachhSeva turns complaint records into a spatial map that highlights problem hotspots. This lets administrators move beyond complaints to observe broader patterns. Such spatial data helps identify hotspots, plan field operations, and allocate resources effectively.

The review shows a gap: no system combines image classification, duplicate detection, community priority, routing, service - level tracking, and heatmaps. SwachhSeva fills this gap by integrating all these functions into one platform. SwachhSeva does not aim to replace judgment but to support municipal staff by reducing repetitive work, organizing complaints, and presenting data clearly .

The literature indicates that AI, mobile technology , GPS, and web tools can improve complaint management. SwachhSeva's main contribution is merging these technologies into an end - to - end workflow while still allowing human review for uncertain AI predictions. This foundation can grow with data sets , more issue categories, real - world municipal data, advanced geospatial analysis, and automated notifications.

III. METHODOLOGY

The proposed SwachhSeva system is developed as an integrated civic complaint management platform that combines image-based issue classification, location-based duplicate detection, automatic department assignment, service-level agreement tracking, priority calculation, and administrative monitoring. The methodology follows a sequence of stages beginning with complaint submission and image processing, followed by classification and verification, geospatial processing, complaint routing, prioritization, and final resolution. The major stages involved in the development of the system are described below.

➤ System Design and Data Flow

SwachhSeva follows a client-server architecture in which citizens interact with the system through a mobile application, while municipal officers and administrators use a web-based dashboard. The citizen submits an image of a civic issue along with its geographical location and an optional description. The request is transmitted to the Flask-based REST API, where authentication, complaint processing, image classification, duplicate detection, and routing operations are performed.

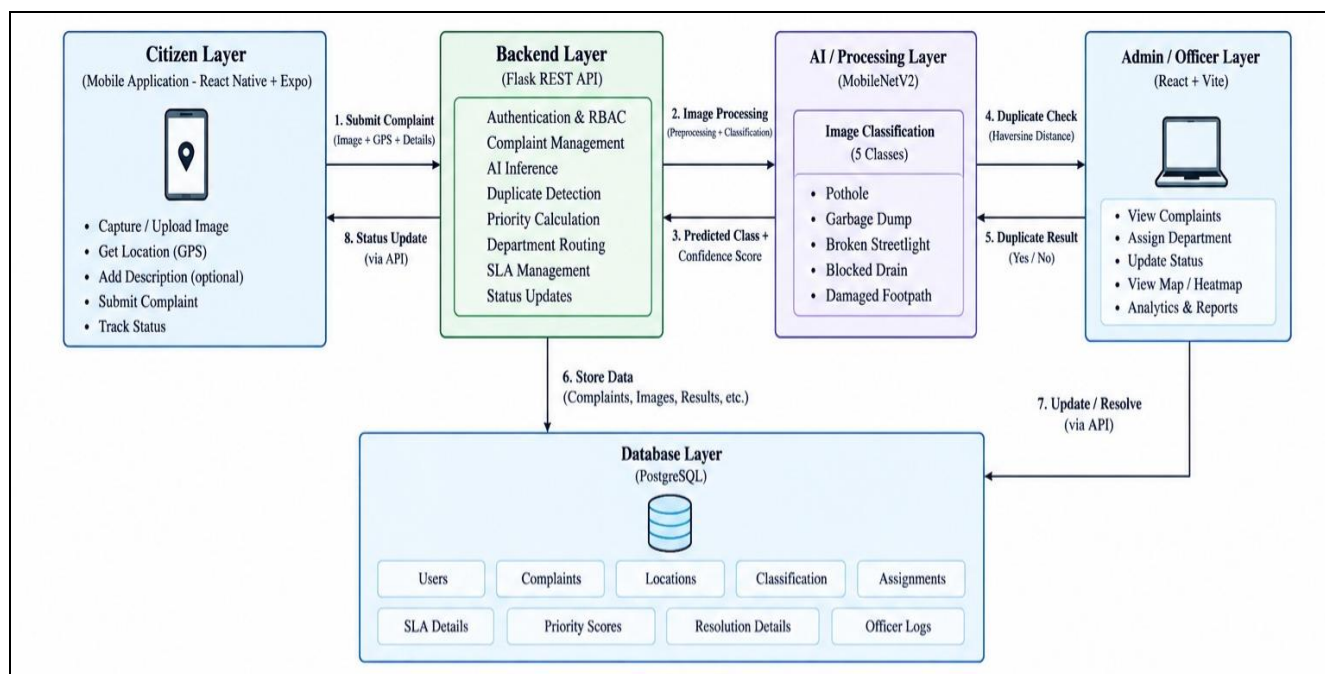


Fig 1 System Design and Data Flow of SwachhSeva.

The processed complaint information is stored in a PostgreSQL database. Based on the detected issue category and prediction confidence, the complaint is either processed automatically or sent for manual verification. Verified complaints are assigned to the appropriate department and provided with an SLA deadline. Officers can subsequently update the complaint status and upload resolution photographs, while citizens can monitor the progress of their submitted complaints.

➤ Dataset Collection and Preparation

A curated dataset of civic issue images is used to train the image classification model. The dataset contains five categories corresponding to the major civic problems handled by the system: pothole, garbage dump, broken streetlight, blocked drain, and damaged footpath.

The collected images are organized according to their respective classes before being used for model training. To maintain a reliable evaluation process, the dataset is divided into training, validation, and testing subsets using a stratified split. Approximately 70% of the images are used for training, 15% for validation, and 15% for testing.

The training data is augmented using operations such as horizontal flipping and brightness and contrast variations. These transformations help the model learn from different visual conditions while avoiding unnecessary changes to the validation and testing data.

➤ Image Preprocessing

Before classification, each submitted image is converted into a suitable format for the trained neural network. The input image is resized to 224×224 pixels and represented using three colour channels.

Image preprocessing ensures that the input dimensions remain consistent with the MobileNetV2 model. The processed image is then supplied to the classification network for feature extraction and prediction.

➤ MobileNetV2-Based Civic Issue Classification

SwachhSeva uses MobileNetV2, a lightweight convolutional neural network, for automatic classification of civic issue images. The model is selected to provide an efficient classification mechanism suitable for integration with a web-based backend.

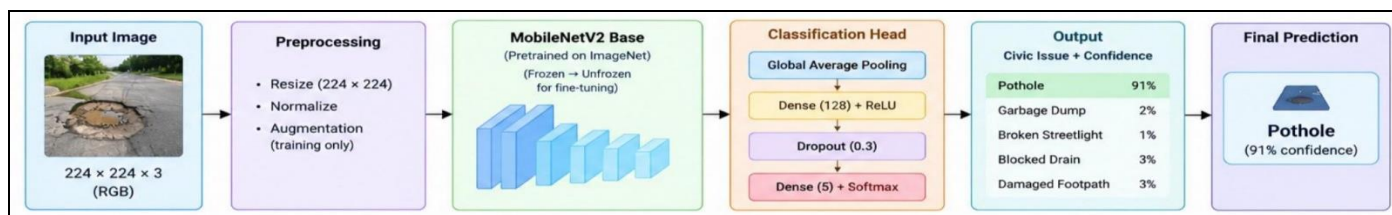


Fig 2 MobileNetV2 Model Architecture Figure

➤ Transfer Learning and Model Training

Transfer learning is used to adapt the pretrained MobileNetV2 network to civic issue classification. Initially, the pretrained base layers are kept frozen while the newly added classification layers are trained on the prepared civic issue dataset.

After the initial training stage, selected layers of the pretrained network are partially fine-tuned using the civic issue images. The model is trained using the Adam optimizer and categorical cross-entropy loss.

The categorical cross-entropy loss is defined as:

$$L = -\sum_{i=1}^5 y_i \log(P_i)$$

Where (y_i) represents the actual class label and (P_i) represents the predicted probability for class (i) .

The trained model is evaluated using accuracy, precision, recall, F1-score, and a confusion matrix to

The pretrained MobileNetV2 feature-extraction layers are combined with a custom classification head. The extracted features are passed through Global Average Pooling, followed by a fully connected layer containing 128 neurons with the Rectified Linear Unit (ReLU) activation function. A dropout layer with a rate of 0.3 is then applied to reduce overfitting. Finally, a five-neuron output layer with Softmax activation produces probabilities corresponding to the five civic issue categories.

The ReLU activation function is expressed as:

$$f(x) = \max(0, x)$$

Where (x) represents the input to the activation function.

For the five-class output layer, Softmax converts the output values into class probabilities:

Where (z_i) is the output value for class (i) , and (P_i) represents the probability assigned to that class.

The predicted civic issue is selected using:

$$\hat{y} = \arg \max_i (P_i)$$

Where $(\hat{y})$ represents the predicted class.

understand its classification performance across the five categories.

➤ Confidence-Based Human Verification

To reduce the possibility of incorrect automatic classification, SwachhSeva uses the prediction confidence produced by the classification model. The confidence value is calculated as the highest probability among the five predicted classes:

$$C = \max(P_1, P_2, P_3, P_4, P_5)$$

A confidence threshold of 60% is used for automatic processing. When the confidence is equal to or greater than 60%, the predicted category is accepted for further complaint processing. When the confidence is below 60%, the complaint is moved to Pending Review for manual verification.

If a trained classification model is unavailable, the complaint is also handled through the manual-review process rather than generating an unreliable prediction.

➤ Geospatial Duplicate Detection

To prevent multiple citizens from submitting separate complaints for the same nearby civic issue, the system performs a location-based duplicate check. When a new complaint is submitted, its geographical coordinates are compared with the coordinates of existing open complaints.

The Haversine formula is used to calculate the geographical distance between two locations. First, the intermediate value is calculated as:

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right) \tag{1}$$

The angular distance is then calculated as:

$$c = 2\arctan2(\sqrt{a}, \sqrt{1-a}) \tag{2}$$

The distance between the two locations is obtained using:

$$d = R \cdot c$$

Where (R) is the Earth's radius, (ϕ_1, ϕ_2) are the latitudes, and $(\Delta\lambda)$ represents the difference in longitude.

For SwachhSeva, open complaints located within 100 m are considered potential duplicates. Citizens can support an existing complaint through upvoting instead of creating another complaint for the same issue.

➤ Department Routing and SLA Management

After the complaint category is verified, the system automatically associates the complaint with the corresponding municipal department. Each issue category is assigned a predefined service-level agreement (SLA).

The implemented routing is as follows:

Table 1 The Implemented Routing

Civic Issue	Department	SLA
Pothole	Roads and Infrastructure	5 days
Garbage Dump	Solid Waste Management	2 days
Broken Streetlight	Electrical Department	3 days
Blocked Drain	Drainage and Sewerage	3 days
Damaged Footpath	Civil Works Division	7 days

The SLA deadline is calculated from the complaint creation time:

$$T_{SLA} = T_{created} + SLA_{days}$$

Where $(T_{created})$ represents the complaint creation time and (SLA_{days}) represents the permitted resolution period.

The complaint progresses through stages such as Submitted, Pending Review, Assigned, In Progress, and Resolved. Complaints that exceed their SLA without resolution are identified as overdue.

➤ Complaint Priority and Escalation

To help officers identify complaints requiring greater attention, SwachhSeva calculates a weighted priority score. The score considers community support through upvotes, complaint age, and the number of days the complaint remains overdue.

The implemented priority calculation is:

$$Priority = (U-1) \times 10 + D \times 15 + A \times 2 \tag{3}$$

Where (U) is the number of upvotes, (D) is the number of days overdue, and (A) is the age of the complaint in days.

The weighting gives additional importance to complaints that have remained unresolved beyond their SLA

period. Complaints that cross the defined escalation condition are highlighted for administrative attention.

➤ Complaint Resolution and System Validation

Once a complaint is assigned to an officer, its status can be updated through the administrative dashboard. The officer can provide resolution details and upload an after-resolution photograph. The citizen can then view the updated complaint status through the mobile application.

The administrative dashboard also provides complaint statistics, geographical visualization, and officer performance information for monitoring the resolution process.

The classification model is evaluated using standard classification metrics. Accuracy is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

Precision is calculated as:

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

Recall is calculated as:

$$Recall = \frac{TP}{TP+FN}$$

The F1-score is calculated using:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

Where (TP) , (TN) , (FP) , and (FN) represent true positive, true negative, false positive, and false negative classifications, respectively.

These evaluation measures are used to assess the classification model, while the end-to-end system is validated

➤ System Architecture Diagram

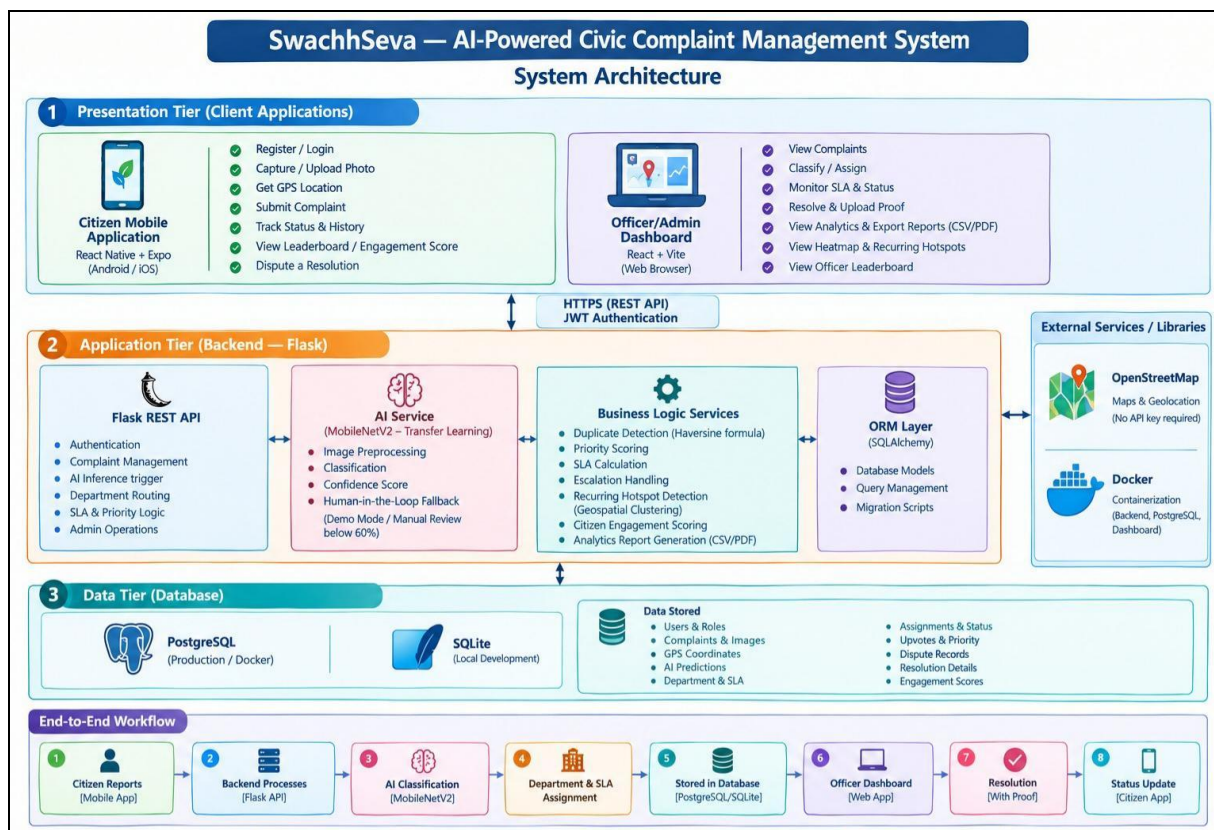


Fig 3 System Architecture Diagram

➤ Citizen Mobile Application:

The application is developed using React Native with Expo and allows citizens to register, log in, capture or upload images of civic issues, and provide GPS-based locations with optional descriptions. Citizens can track their complaints and check for nearby existing complaints to reduce duplicate submissions. Existing complaints can also be supported through upvotes.

➤ Flask REST API Backend:

The Flask-based REST API acts as the central processing layer. It handles authentication, complaint submission, image processing, classification, duplicate detection, department routing, and status management. Role-based access is provided for citizens, officers, and administrators, with password hashing and JWT-based authentication for security.

➤ AI Classification Module:

The system uses a MobileNetV2 convolutional neural network with transfer learning to classify images into five categories: pothole, garbage dump, broken streetlight, blocked drain, and damaged footpath. A confidence threshold is used

to send low-confidence predictions or cases without a trained model for manual officer review.

➤ Geospatial and Complaint Processing:

The Haversine distance formula is used to identify nearby existing complaints and reduce duplicate reports. The same location-based analysis helps identify recurring civic problems. The system also applies priority scoring based on community confirmations, complaint age, and SLA-overdue duration.

➤ Database and Administrative Dashboard:

Complaint and user information, classification results, locations, and status details are stored in a relational database. The administrative dashboard, developed using React with Vite, provides complaint management, map and heatmap visualization, analytics, and reporting. Classified complaints are routed to the appropriate department with an SLA deadline, while officers can monitor and update their resolution status.

The overall architecture follows:

Citizen Mobile Application → Flask REST API → AI Classification / Geospatial Processing → Department Routing and SLA Management → Database → Administrative Dashboard → Resolution → Citizen

IV. RESULTS AND DISCUSSION

The developed SwachhSeva system was evaluated based on its ability to classify civic issues, process citizen complaints, identify potential duplicate complaints, route complaints to the appropriate departments, and support

monitoring through the administrative dashboard. The evaluation considered both the performance of the MobileNetV2-based classification model and the functionality of the complete complaint management workflow.

➤ Civic Issue Classification Results

The MobileNetV2 model was trained using transfer learning on a curated civic-issue dataset containing five classes: pothole, garbage dump, broken streetlight, blocked drain, and damaged footpath. Following a stratified 70/15/15 split, the model was evaluated on a held-out test set of 20 images (4 per class) that was not used during training or validation.

Table 2 Classification Performance of the MobileNetV2 Model (Test Set, n = 20)

Metric	Value
Accuracy	100.0%
Precision (macro-average)	100.0%
Recall (macro-average)	100.0%
F1-Score(macro-average)	100.0%

Per-class results were identical across all five categories (precision = recall = F1 = 1.00, support = 4 for each class), and the weighted average matched the macro average exactly, consistent with the perfectly balanced test set.

While these results demonstrate that the transfer-learning approach successfully distinguished between the five selected civic-issue categories on the available data, this result should be interpreted cautiously rather than as evidence of production-level robustness. **The test set size (20 images total, 4 per class) is small**, and the perfect score is at least partly attributable to this small sample size combined with the visual distinctiveness of the five chosen categories, rather than proof that the model would generalise equally well to a larger,

more visually varied, real-world image stream. A substantially larger and more diverse dataset — spanning different lighting conditions, camera angles, occlusion levels, and regional infrastructure variation — would be required before this accuracy figure could be treated as representative of real-world deployment performance. This limitation is discussed further in Section V.9.

➤ Confusion Matrix Analysis

A confusion matrix was generated to examine the class-wise predictions of the trained model on the test set (Fig. 2). Every test image was classified into its correct category, producing a perfect diagonal matrix with no off-diagonal (misclassified) entries.

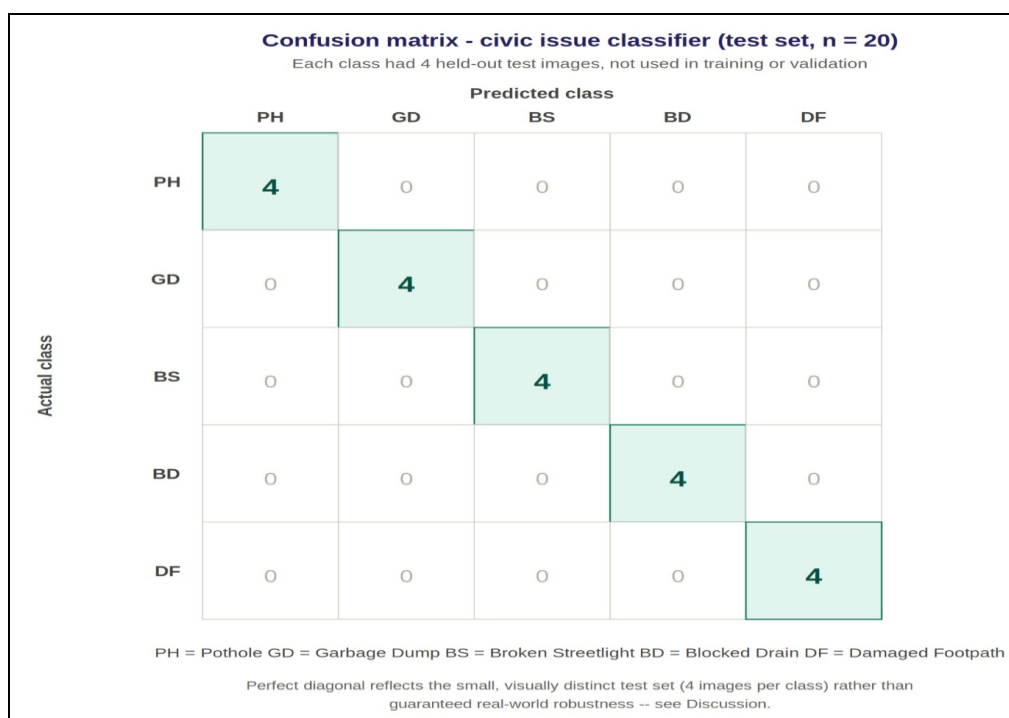


Fig 4 Confusion Matrix of the MobileNetV2 Civic-Issue Classification Model (Test Set, n = 20).

The absence of any inter-class confusion — for instance, between visually similar categories such as blocked drains and garbage dumps — is a positive result for this dataset, but as noted above, should be read in the context of the dataset's limited size rather than as a claim of zero confusability in general.

➤ *Confidence-Based Classification*

The system uses a confidence threshold of **60%** to control automatic classification. Predictions with confidence equal to or greater than 60% are accepted for automatic department/SLA routing, while predictions below the threshold — or requests made when no trained model is loaded (Demo Mode) — are routed to manual officer review instead.

This mechanism was verified functionally during system testing: submissions in Demo Mode were correctly and explicitly flagged as 'Pending Review' rather than assigned a fabricated classification, and once a trained model was loaded, `/api/health` correctly reported `ai_model_loaded: true` with subsequent submissions receiving real confidence-scored predictions. This human-in-the-loop mechanism reduces the

risk of an uncertain or unavailable model silently misrouting a complaint.

➤ *Duplicate Complaint Detection*

The geospatial duplicate-detection mechanism compares a newly submitted complaint's location against existing open complaints using the Haversine distance formula, flagging matches within a **100 m radius**. This was verified with a deliberate test case: a complaint submitted approximately 24.7 m from an existing open report was correctly identified as a nearby match and offered to the citizen for confirmation ("upvote") rather than automatically treated as a duplicate submission.

This functionality addresses repeated reporting of the same civic problem: rather than creating a redundant complaint, citizens can confirm an existing nearby report, incrementing its `upvote_count` and giving officers a quantifiable signal of community-perceived urgency.

➤ *Department Routing and SLA Management*

The system automatically associates classified complaints with the appropriate department and a predefined Service-Level Agreement (SLA) duration.

Table 3 Department Routing and SLA

Civic Issue	Assigned Department	SLA
Pothole	Road and Infrastructure	5 days
Garbage Dump	SolidWasteManagement	2 days
BrokenStreetlight	Electrical Department	3 days
Blocked Drain	Drainage and Sewerage	3 days
DamagedFootpath	Civil Works Division	7 days

All five mappings were verified functionally: classifying a test complaint into each category correctly produced the corresponding department assignment and SLA deadline (submission timestamp + configured SLA days).

➤ *Priority, Escalation and Dispute Results*

The priority mechanism combines community upvotes, complaint age, overdue duration, and dispute frequency into a single, transparent priority score:

$$\text{Priority} = (U - 1) \times 10 + D \times 15 + A \times 2 + C \times 25$$

Where U is the up vote count, D is the number of days overdue, A is the complaint age in days, and C is the number of times the complaint has been disputed after resolution.

This formula was verified against known inputs: a complaint with 2 up votes, 4 days overdue, and same-day age produced a priority score of exactly 70 — matching $(2-1) \times 10 + 4 \times 15 + 0 \times 2 = 70$ — and a separately tested disputed complaint (0 extra up votes, 0 days overdue, same-day age, 1 dispute) produced a score of exactly 25, matching $0 + 0 + 0 + 1 \times 25 = 25$.

A complaint is additionally marked escalated once it passes 2 days beyond its SLA deadline while unresolved. The

citizen-facing dispute mechanism allows a citizen to reopen a resolved complaint with a stated reason if the resolution does not hold; this was validated end-to-end, correctly transitioning the complaint's status from 'RESOLVED' to 'REOPENED' and increasing its priority score accordingly, ensuring disputed resolutions surface near the top of an officer's priority-sorted queue.

➤ *Recurring Hotspot Detection*

To distinguish isolated incidents from systemic infrastructure problems, complaints are grouped by issue type and location (using the same Haversine-based proximity logic, applied across a location's entire history including already-resolved complaints) via a connected-components clustering approach. A cluster of 3 or more same-category reports within 100 m is flagged as a recurring hotspot, with severity graded by cluster size (Moderate: 3–4, High: 5–6, Critical: 7+).

This was validated on real accumulated system data: a cluster of 6 pothole reports at a single location, spanning submissions from 30 August to 9 September, was correctly flagged as a High-severity hotspot, with the system additionally surfacing that one of the six reports had been disputed after a prior resolution — a concrete signal that the location's underlying infrastructure problem had not been permanently addressed by earlier patchwork fixes.

➤ Citizen Engagement and Administrative Dashboard Results

A citizen engagement score, $\text{Score} = 5n_{\{\text{submitted}\}} + 15n_{\{\text{resolved}\}} + 3n_{\{\text{accurate}\}}$, was implemented to reward outcome-validated participation over raw submission volume. This was verified against two citizens with differing activity patterns, both producing scores that matched the formula exactly.

The administrative dashboard provides a centralized interface for monitoring and managing submitted complaints, including complaint listing and filtering, a geographic map and density heatmap, department/status/category statistics, an officer performance leaderboard, a citizen engagement leaderboard, and CSV/PDF analytics export with optional date-range filtering — all validated functionally against a live backend during development.

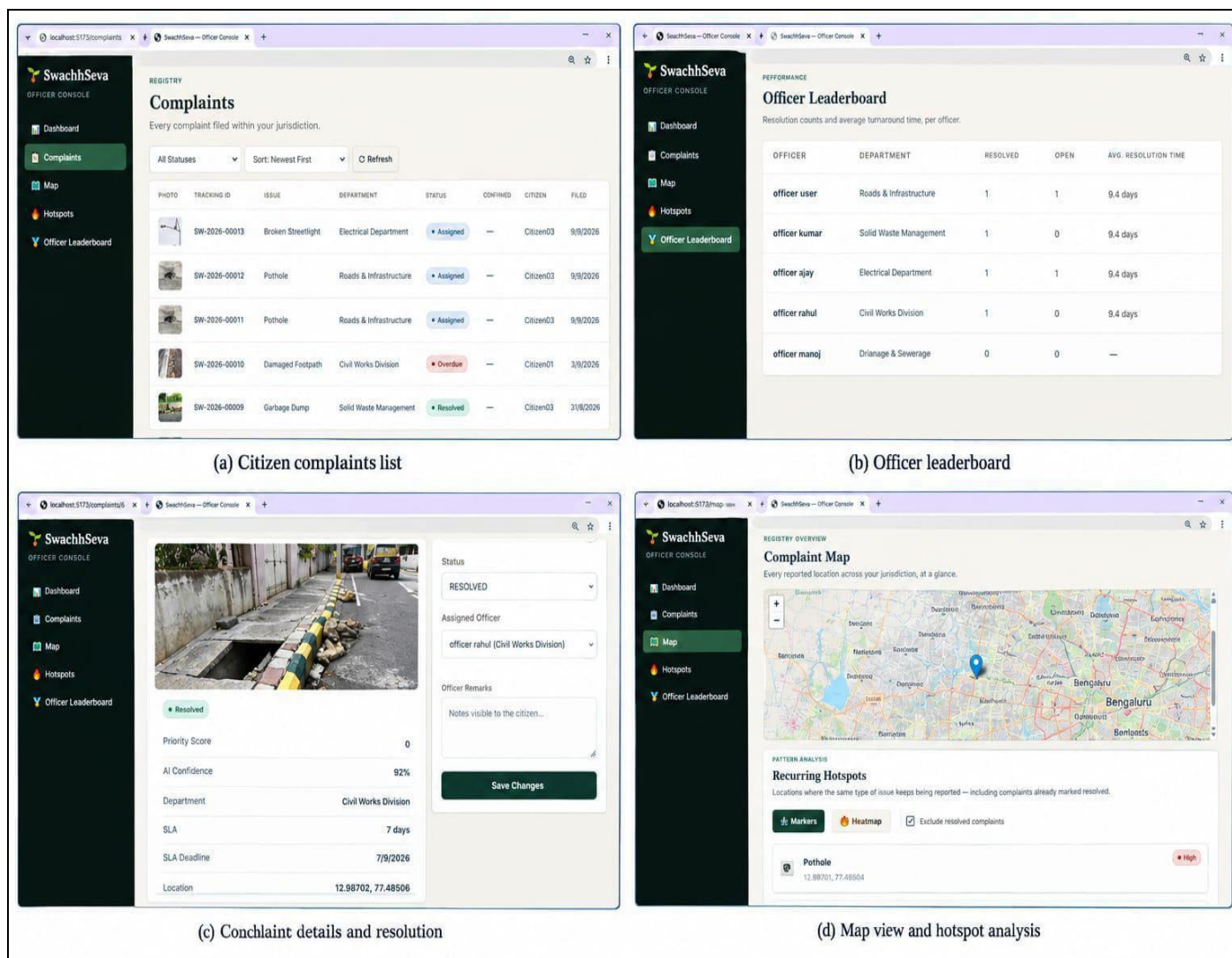


Fig 5 Administrative Dashboard for Complaint Monitoring and Management.

➤ End-to-End System Validation

The complete system was validated by following the complaint lifecycle from submission to resolution across all implemented roles (citizen, officer, admin). A typical complaint passes through image and location submission, backend processing, AI classification or manual verification, duplicate checking, department routing, SLA and priority calculation, officer processing, resolution, and — where applicable — citizen dispute and reopening.

Testing also surfaced and resolved a cross-platform defect: a file-path handling difference between POSIX and Windows filesystems initially prevented resolution photos (stored in a subfolder) from being served correctly on a Windows deployment, while root-level complaint photos were

unaffected. This was traced to `os.path.normpath()` converting forward slashes to backslashes in a way that broke the underlying file-serving lookup, and was resolved by validating the URL-style path directly instead. This illustrates the practical value of testing beyond a single development platform.

➤ Discussion

The results show that SwachhSeva combines machine-learning-based classification with transparent, rule-based complaint-management mechanisms. The MobileNetV2 model provided perfect classification accuracy on the available test set, while the 60% confidence threshold ensures uncertain predictions are not processed automatically. However, given the limited size of the current dataset (20 test

images), this accuracy figure should be understood as a demonstration of the pipeline's correctness rather than a robust estimate of real-world classification performance; a substantially larger and more visually diverse dataset would be needed before deployment-grade accuracy claims could be made.

The geospatial duplicate-detection and upvoting mechanism addresses repeated reporting of the same issue, while the recurring-hotspot detection extends this same geospatial logic across a location's full history to surface systemic infrastructure problems rather than treating each report in isolation. Department routing and SLA tracking provide a structured mechanism for assigning responsibility and monitoring resolution timelines, and the transparent priority-scoring formula — deliberately linear and auditable rather than a learned ranking model — allows officers to understand exactly why one complaint outranks another. The citizen dispute mechanism closes a feedback loop that would otherwise be one-directional: without it, a resolution marked complete has no mechanism for a citizen to contest it even when the underlying problem persists.

Overall, the system demonstrates that an integrated pipeline combining AI-assisted classification, geospatial analysis, transparent prioritization, and citizen feedback mechanisms can provide a structured, low-cost approach to civic complaint management. The primary limitation identified through this evaluation is dataset scale rather than system design: expanding the training and test datasets — particularly for currently underrepresented categories such as blocked drains — remains the most direct path to strengthening the classification component's real-world reliability.

V. CONCLUSIONS

SwachhSeva presents an AI-powered approach for improving the management of civic complaints in urban areas. The system integrates a citizen mobile application, Flask-based REST API, MobileNetV2 image classification model, geospatial duplicate detection, complaint prioritization, automatic department routing, and an administrative dashboard into a single workflow. Citizens can report civic issues using photographs and GPS-based locations, while municipal officers can monitor, assign, and manage complaints through the web dashboard.

The use of MobileNetV2 with transfer learning provides automated classification of common civic issues, while the confidence-based manual review mechanism helps prevent unreliable classifications when the model is uncertain or unavailable. Haversine-based duplicate detection reduces repeated complaints, and priority scoring, SLA tracking, and departmental routing support more organized complaint handling. The system also incorporates citizen participation through confirmations, dispute and reopen mechanisms, and engagement recognition.

Overall, SwachhSeva provides a practical and low-cost framework for improving the efficiency, transparency, and accountability of civic complaint management. The proposed system can be applied to municipal corporations, smart-city initiatives, and infrastructure maintenance departments. Its future improvement can focus on larger and more diverse real-world datasets, improved AI models, and wider deployment to support reliable civic issue management at municipal scale.

REFERENCES

- [1]. K. S. N. Gowtham, V. Munduri, M. V. Deepak, M. P. L. P. D. Sravanthi, M. A. P. Varma, and S. Bonthu, "A Machine Learning Integrated Framework for Automated Civic Complaint Validation and Prioritization in Smart City Governance," in *Proc. 4th Int. Conf. on Self Sustainable Artificial Intelligence Systems (ICSSAS)*, 2026, p.1071, doi: 10.1109/ICSSAS68835.2026.11559493.
- [2]. P. K. S., S. Srividhya, S. Vishakh, and V. B. R., "CivicEye: AI and Blockchain-Powered Platform for Transparent Civic Issue Reporting and Resolution," in *2026 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 2026, doi: 10.1109/IITCEE67948.2026.11394376.
- [3]. M. S. A., S. K., and N. K., "Smart Complaint Routing Systems for Public Issues Using Multimodal Artificial Intelligence," in *2026 IEEE International Conference for Convergence in Computing Technology (I3CTCON)*, Lonavala, India, 2026, doi: 10.1109/I3CTCON68242.2026.11508062.
- [4]. V. Bhatia, A. K. Pathak, A. K. Singh, S. Raghuwanshi, P. Kumar, and O. Krejcar, "Real-time pothole detection using YOLO models: An efficient and cost-effective solution for infrastructure monitoring," *International Journal of Transportation Science and Technology*, 2026, doi: 10.1016/j.ijtst.2025.12.004.
- [5]. S. R. Kuthyar, S. Roopashree, V. Rasika, S. Manjesh, R. Girimaji, R. D. Arjun, and S. Priyadarshini, "An Intelligent Pothole Detection and Alerting System using Mobile Sensors and Deep Learning," in *2021 IEEE 18th India Council International Conference (INDICON)*, Guwahati, India, 2021, pp. 1–6, doi: 10.1109/INDICON52576.2021.9691661.
- [6]. G. Babu R., C. Chellaswamy, M. Surya Bhupal Rao, M. Saravanan, E. Kanchana, and J. Shalini, "Deep Learning Based Pothole Detection and Reporting System," in *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, 2020, pp. 143–148, doi: 10.1109/ICSSS49621.2020.9202061.
- [7]. H. Abdu and M. H. M. Noor, "A Survey on Waste Detection and Classification Using Deep Learning," *IEEE Access*, vol. 10, pp. 128151–128165, 2022, doi: 10.1109/ACCESS.2022.3226682.

- [8]. H. Ma, Y. Ye, J. Dong, and Y. Bo, “An Intelligent Garbage Classification System Using a Lightweight Network MobileNetV2,” in *2022 7th International Conference on Signal and Image Processing (ICSIP)*, Suzhou, China, 2022, pp. 531–535, doi: 10.1109/ICSIP55141.2022.9886985.
- [9]. K. K. Sai, D. D. V. Kumar, A. Sahrudhay, and K. Dharavath, “Pothole Detection Using Deep Learning,” in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, Belagavi, Karnataka, India, 2023, pp. 1–6, doi: 10.1109/INCOFT60753.2023.10425461.
- [10]. S. Bej, S. Roy, D. Daw, A. Paul, S. Saha, S. Maity, and N. Ghosh, “SmartPave: An Advanced IoT-Based System for Real-Time Pothole Detection, Tracking, and Maintenance,” in *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)*, Bhubaneswar, India, 2023, pp. 532–537, doi: 10.1109/APSIT58554.2023.10201720.
- [11]. S. Alshammari and S. Song, “3Pod: Federated Learning-Based 3 Dimensional Pothole Detection for Smart Transportation,” in *2022 IEEE International Smart Cities Conference (ISC2)*, Pafos, Cyprus, 2022, pp. 1–7, doi: 10.1109/ISC255366.2022.9922195.
- [12]. I. Isha, S. Bhardwaj, S. Kumari, U. Srivastva, et al., “Analysing Urban Grievances on X (Formerly Twitter) Using NLP: A Framework for Civic Action,” in *2025 IEEE 7th International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2025, doi: 10.1109/ICCCA66364.2025.11325672.
- [13]. B. Kabilan, R. Sairam, and M. Praveen, “Enhanced CNN Architecture for Accurate Waste Classification in Smart Cities,” in Proc. 2024 Int. Conf. IoT Based Control Networks and Intell. Syst. (ICICNIS), Bengaluru, India, 2024, pp. 538–543, doi: 10.1109/ICICNIS64247.2024.10823383.
- [14]. D. Jyothirmai, S. Sai Charan Reddy, N. Sai Karthikeya, S. Dinesh Reddy, P. Jagadeesh, and R. Pitchai, “Pothole Detection and Enhanced Road Safety Using Machine Learning,” in Proc. 2024 5th Int. Conf. Electron. Sustainable Commun. Syst. (ICESC), Coimbatore, India, 2024, pp. 1392–1396, doi: 10.1109/ICESC60852.2024.10690147.