

Automated Public Complaints Understanding: Managing Multimodal Inputs to Improve Authority-Level Decision Support

by

Sabiha Akter (222014017)

Moitree Mazumder (222014011)

Marjanul Jannat Prapti (222014082)

Kazi Azizul Haque (221014088)

*Capstone project report (CSE 4098C) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Shah Jafor Sadeek Quaderi



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SPRING 2026

© Sabiha Akter, Moitree Mazumder, Marjanul Jannat Prapti, and Kazi Azizul Haque

All rights reserved

DECLARATION

Project Title	Automated Public Complaints Understanding: Managing Multimodal Inputs to Improve Authority-Level Decision Support
Authors	<i>Sabiha Akter, Moitree Mazumder, Marjanul Jannat Prapti, and Kazi Azizul Haque</i>
Student IDs	222014017, 222014011, 222014082 and 221014088
Supervisor	Shah Jafor Sadeek Quaderi

We declare that this capstone project report entitled *Automated Public Complaints Understanding: Managing Multimodal Inputs to Improve Authority-Level Decision Support* is the result of our own work except as cited in the references. The capstone project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Sabiha Akter
222014017

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Moitree Mazumder
222014011

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Marjanul Jannat Prapti
222014082

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Kazi Azizul Haque
221014088

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: June 08, 2026



Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

APPROVAL OF THE COMMITTEE

This is to certify that the capstone project report entitled **Automated Public Complaints Understanding: Managing Multimodal Inputs to Improve Authority-Level Decision Support**, submitted by **Sabiha Akter** (222014017), **Moitree Mazumder** (222014011), **Marjanul Jannat Prapti** (222014082) and **Kazi Azizul Haque** (221014088) are undergraduate students of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Shah Jafor Sadeek Quaderi
Supervisor

Rezwan Al Islam Khan, PhD
Associate Professor

Nasir Uddin Ahmed
Lecturer

Place: Dhaka

Date: June 08, 2026

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our respected supervisor, Shah Jafor Sadeek Quaderi, for providing valuable guidance, continuous support, and constructive feedback throughout this capstone project. His guidance helped us improve both the technical development of the system and the quality of this report. We are also grateful to the Respected Teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh for their academic support, encouragement, and valuable learning throughout our undergraduate studies. We would like to thank our parents and family members for their unconditional love, patience, and continuous encouragement. Their sacrifices and support have been a constant source of motivation throughout this journey. Finally, we would like to thank our friends, classmates, and everyone who directly or indirectly helped us complete this capstone project successfully.

Sabiha Akter, Moitree Mazumder, Marjanul Jannat Prapti and Kazi Azizul Haque
University of Liberal Arts Bangladesh

Date: 08 June, 2026

To ...

*My family and to my own perseverance,
for the strength, patience, and determination that
carried me through this work.*

– Sabiha Akter

To ...

*My beloved parents, sister and friends
for their constant prayers, unconditional love and
support.*

– Moitree Mazumder

To ...

*My parents and loved ones,
for their guidance, motivation, and unconditional
support.*

– Marjanul Jannat Prapti

To ...

*My family,
for their love, sacrifices, and encouragement
throughout this journey.*

– Kazi Azizul Haque

ABSTRACT

Cities deal with many civic and environmental complaints including waste management, infrastructure issues, drainage problems and public utility issues. These reports are often unorganized, repetitive or incomplete, which makes it hard for authorities to respond effectively. Current systems depend on manual sorting and do not prioritize issues, causing delays and lowering citizen trust. To overcome this issues, this work suggest an AI-based framework CivicAI for understanding complaints automatically using multimodal inputs which uses Natural Language Processing to classify complaints texts, Computer Vision to analyze visual evidence from images, and Geolocation-based analysis to identify duplicate complaints and spatial hotspots. Transformer-Based Natural Language Processing was used for classification of complaints based on text, while YOLOv8n was implemented for detection of visual evidence. Shared-Space Late Fusion framework was employed to integrate predictions from text and images. In addition, the model allows summarizing complaints, calculating their prioritization level, detecting duplicates, recognizing hot spots, identifying escalations, and generating reliability scores using a dedicated dashboard for authorities. The RoBERTa text classifier achieved 86.61% accuracy, 0.8604% weighted-F1, and 0.6149 macro-F1. The YOLOv8n detector achieved 0.619 precision and 0.536 mAP@0.5. The findings show that multimodal complaint analysis can improve complaint understanding and provide practical support for faster, more organized, and evidence-based civic service response.

Keywords:Public complaints, Smart cities, Multimodal AI, Natural Language Processing (NLP), Computer vision, Geotagging, Decision support systems

Contents

Acknowledgements	vi
Dedication	vii
1 Introduction	1
1.1 Problem statement	2
1.2 Aim, objectives & Motivation	3
1.2.1 Aim	3
1.2.2 Objectives	3
1.2.3 Motivation	4
2 Literature Review	5
2.1 Natural Language Processing for Complaint Classification	5
2.2 Computer Vision for Evidence Detection in Images	6
2.3 Geospatial Analysis and Duplicate Detection	6
2.4 Complaint Prioritization and Dashboard Presentation	6
2.5 Key Findings and Identified Gaps	7
2.6 Analysis of Existing Researches	7
2.7 Model Description	11
2.7.1 DistilBERT	11
2.7.2 RoBERTa	11
2.7.3 SBERT	12
2.7.4 YOLO	12
3 Plans and Derivable	14
3.1 Budget Table	15
3.2 Tools and Libraries	16

4	Methodology	17
4.1	Complaint Text Dataset	19
4.2	Image Dataset	20
4.2.1	Text Dataset Cleaning and Preprocessing	22
4.2.1.1	Text Normalization	22
4.2.1.2	Filtering Noisy and Ambiguous Phrases	22
4.2.1.3	Removing Label Ambiguity	22
4.2.1.4	Merging Semantically Similar Categories	23
4.2.1.5	Minimum-Sample Filtering Strategy for Stratified K-fold Cross-validation	23
4.2.2	Image Data Preprocessing	25
4.2.3	Train/Test Split	25
4.2.3.1	Text Dataset	25
4.2.3.2	Image Dataset	25
4.2.4	Applying Cross-Validation to Evaluate Text Classifier Models	26
4.2.5	Unified YOLO Detection Dataset Construction For Image Data	27
4.2.6	YOLOv8 Model Training Procedure	27
4.2.7	Multimodal Fusion and Consistency Checking	28
4.2.7.1	Shared-space alignment of text and image evidence.	28
4.2.7.2	Fusion decision and safe fallback.	29
4.2.7.3	Weighted late fusion.	29
4.2.7.4	Consistency Checking	30
4.2.8	Decision Support Modules	30
4.2.8.1	Sentiment scoring with VADER	31
4.2.8.2	Urgency normalisation from sentiment	31
4.2.8.3	Priority ranking using Simple Additive Weighting (SAW)	32
4.2.8.4	Threshold-based escalation	33
4.2.8.5	Geolocation hotspot detection	34
4.2.8.6	Duplicate detection	34
4.2.8.7	Operational summary generation and dashboard output	35
4.2.9	Evaluation Metrics	35
4.2.10	Implementation Details	36

5	Design and Development	37
5.1	Functional Requirements	37
5.2	Non-Functional Requirements	38
5.3	System Architecture	39
5.4	Frontend Design and Development	41
5.4.1	Welcome Page	41
5.4.2	Citizen Reporting Interface	42
5.4.3	Authority Dashboard	44
5.5	Backend Design and Development	45
5.5.1	API Layer	45
5.5.2	AI Inference Pipeline	46
5.5.3	Database and Storage Integration	46
5.6	Website Features	47
5.7	Testing and Validation	48
5.8	Deployment and Hosting	49
6	Result and Discussion	50
6.1	Text Model Comparison	50
6.2	Final Selected Text Model (RoBERTa)	52
6.3	Visual Model Results (YOLOv8n) – Validation and Test Performance	53
6.4	Late Fusion and Decision Support Pipeline	64
6.4.1	Multimodal fusion behavior and consistency checking	65
6.4.2	Authority-Oriented Decision Outputs and Operational Support	67
6.4.2.1	Complaint Summarization for Rapid Review	69
6.4.2.2	Duplicate Detection and Workload Reduction	69
6.4.2.3	Hotspot Detection and Geospatial Decision Support	70
6.4.3	End-to-end Readiness and Dashboard Integration	71
6.5	Strengths, Limitations and Implications	72
6.6	Social, Legal and Ethical Issues	74
7	Conclusion	75
	References	76

List of Figures

3.1	Project Timeline Gantt Chart showing all phases from initiation through deployment.	15
4.1	CivicAI Methodology Diagram	18
4.2	Visualization of Top category distribution in text dataset	20
4.3	Overview of Images with class labels	21
5.1	CivicAI System Architecture	40
5.2	CivicAI Home Page	42
5.3	CivicAI Citizen Reporting Interface	43
5.4	CivicAI Authority Dashboard	44
6.1	Model detection of urban infrastructure issues, including damaged electric wires, broken road signs, and potholes.	57
6.2	Model detection of urban issues, including littering and fallen trees in public areas.	58
6.3	Model recognition of Images outside of dataset/real time.	59
6.4	Recall–Confidence curve of the YOLOv8n detector	59
6.5	Precision–Recall curve of the YOLOv8n detector	60
6.6	F1–Confidence curve of the YOLOv8n detector	61
6.7	Confusion matrix of the YOLOv8n detector	62
6.8	Per-class mAP50-95	63
6.9	Per-class Precision	64
6.10	Category review-source and AI assessment panel showing results.	65
6.11	Evidence panel showing the uploaded complaint image and the annotated detector output together with the extracted visual-evidence label and confidence score.	66

6.12 Complaint review interface showing the decision summary and priority status.	68
6.13 Analytics view showing repeated area-category patterns together with duplicate-linked complaints.	70
6.14 Hotspot map showing complaint concentration through a heatmap overlay, grouped area markers, and interactive pop-up labels for interpretation.	71

List of Tables

2.1	Contributions and Limitations Analysis in Complaints Management System	9
2.2	Preferred datasets,models and their performance in related Complaint Management Systems	10
3.1	Estimated Project Budget Breakdown	16
4.1	Overall Dataset Summary	21
4.2	Effect of Minimum Sample Threshold on Dataset Filtering	24
4.3	Per-class Dataset Integrity and Split Distribution	25
6.1	DistilBERT 9-fold Stratified Cross-validation Results	50
6.2	RoBERTa 9-fold Stratified Cross-validation Results	51
6.3	SBERT-based Classifier 9-fold Stratified Cross-validation Results . . .	52
6.4	RoBERTa Epoch-wise Validation Metrics (Internal Validation Split) .	53
6.5	Final RoBERTa Performance on the Locked 20% Test Set	53
6.6	YOLOv8n Training Losses Across Epochs	54
6.7	YOLOv8n Validation Performance (Best.pt, imgsz=416)	55
6.8	YOLOv8n Test Performance (Imgsz=416)	56

Chapter 1

Introduction

Governments provide essential services across diverse sectors, but traditional complaint channels are often slow, inefficient, and resource-constrained, causing many citizen grievances to go unaddressed. E-government and digital services offer a modern solution by streamlining administrative processes, reducing paperwork, and enabling faster, cost-effective complaint registration. These digital platforms improve accessibility, transparency, and follow-up on citizen complaints. Overall, e-government enhances accountability and strengthens public trust in government service delivery. Governments provide essential public services across multiple sectors, but when service quality falls short, citizens often submit complaints through traditional physical or representative channels [Madyatmadja et al. \(2023\)](#).

Smart cities, driven by big data and technology, enable citizens to conveniently report environmental issues through mobile phones and social networks, reflecting growing public awareness and participation [Jiao et al. \(2021\)](#). Advancing technology has led to web-based systems that automate complaint handling, with NLP integration improving the speed and accuracy of processing [Putra et al. \(2024\)](#). Platforms like FixMyStreet and Züri wie Neu were developed to let citizens report city infrastructure issues with descriptions, locations, and images, though reports are still manually categorized by authorities [Moya et al. \(2021\)](#). By using machine learning, governments can automatically detect and classify issues from citizen reports, easing manual workload, speeding responses, and improving public trust [Vrabie \(2025\)](#).

Artificial Intelligence has evolved from text-based systems to multimodal learning, integrating vision, audio, and language. This fusion overcomes the limitations of single-modality systems, enabling richer context understanding and more accurate

outputs [Arjunan \(2024\)](#). Multimodal AI combines different data types through machine learning to enhance understanding [Liu et al. \(2025\)](#). With the increasing use of smart city initiatives, there is a pressing need for intelligent complaint management systems that can automatically interpret and organize various inputs like text, images, and geolocation. Improvements in Natural Language Processing (NLP) and Computer Vision (CV) allow for better analysis of data generated by citizens. Meanwhile, geotagging helps group complaints by location to identify hotspots. By combining these technologies, an AI-driven framework can give authorities real-time decision support, reduce confusion, find duplicate reports, and prioritize issues based on their severity and urgency.

This research presents a multimodal AI system for understanding public complaints automatically. The system seeks to close the gap between citizen reporting and the decision-making process of authorities by offering a smart, interactive dashboard that shows actionable insights. This approach supports participatory governance and fosters the creation of sustainable, responsive, and efficient urban management practices.

1.1 Problem statement

Public authorities receive large numbers of civic complaints related to roads, waste management, drainage, utilities, public safety, and environmental issues. Many complaint-handling systems still depend on manual categorization, verification, routing, and prioritization, which becomes slow and inefficient as complaint volume increases [Madyatmadja et al. \(2023\)](#). However, even with the increased efficiency in handling such complaints digitally, the majority of such methods fail to incorporate the intelligent tool that could convert such unstructured data into action plans for governmental authorities, [Putra et al. \(2024\)](#). In addressing the central issue highlighted in this research, there is no single comprehensive AI-driven system to help understand all aspects of a complaint in terms of its text, visuals, and geographic coordinates. Current approaches separate these pieces of data and fail to recognize duplicates or repeating geographic concerns, verify complaints, and identify urgent ones. In previous research, it has been found that taking into consideration both textual information, images, time, and geographic information could help with improving understanding of complaints [Moya et al. \(2021\)](#). In

addition, using geospatial data would be useful for recognizing recurring problems [Ramalho et al. \(2020\)](#). As a result, a holistic system is required. Consequently, there is a need for a coordinated multimodal decision support system that can help in enhancing the quality of complaint classification, prioritizing, deduplication, hot-spot identification, and response from the authorities [Devanathan et al. \(2024\)](#).

1.2 Aim, objectives & Motivation

1.2.1 Aim

The aim of this project is to enhance public complaint management by supporting more efficient, transparent, and evidence-based civic service delivery. The project seeks to improve how citizen complaints are understood, organized, and used for authority-level decision-making, especially in cases where complaints contain unstructured descriptions, visual evidence, and location-based information. In a broader sense, this work aims to reduce delays in complaint handling, support better prioritization of public issues, identify recurring urban problems, and strengthen the connection between citizen reporting and timely authority response.

1.2.2 Objectives

Key Objectives of this research are:

- To design a complaints management system using multimodal inputs through artificial intelligence technology regarding the civil society and environmental sector
- To implement machine learning techniques based on natural language processing and computer vision for complaint understanding, categorization, and evidence analysis.
- To evaluate the effectiveness of the proposed system through experimental testing.

1.2.3 Motivation

Current research shows growing interest in multimodal complaint analysis, yet an important gap remains in the development of integrated systems for civic and environmental issue management. While previous studies have successfully applied Natural Language Processing to classify complaint texts [Madyatmadja et al. \(2023\)](#); [Jiao et al. \(2021\)](#) and computer vision to analyze complaint-related images [Vrabie \(2025\)](#); [Mamta et al. \(2024\)](#), these approaches are often developed separately. In addition, many existing systems do not adequately incorporate geolocation for complaint grouping, duplicate identification, and hotspot analysis [Ramalho et al. \(2020\)](#) nor do they provide an effective authority-facing dashboard for interactive review and decision support [Munot et al. \(2022\)](#). This project is motivated by the need to bring text, image, and geolocation analysis into a unified framework for public complaint understanding. By integrating these components, the system can reduce fragmentation in complaint handling, improve complaint review and prioritization, and support more efficient and informed public service response.

Chapter 2

Literature Review

The development of an intelligent system for civic complaint management draws on a growing body of research in Natural Language Processing (NLP), Computer Vision (CV), geospatial analysis, and decision-support systems. Existing studies provide important foundations for understanding how complaint text can be classified, how visual evidence can be analyzed, how location information can support grouping and hotspot detection, and how complaint data can be prioritized for authority review. This literature review brings together the key research areas that inform the technical design and methodological choices of the proposed system.

2.1 Natural Language Processing for Complaint Classification

A key part of this project is the automated sorting of textual complaints. Multiple studies show that machine learning and deep learning work well for this task. [Madyatmadja et al. \(2023\)](#) directly compared classifiers for e-gov complaints. They found that Support Vector Machines (SVM) and Random Forests (RF) reached high accuracy, setting a solid baseline for text classification. For a more detailed understanding, [Jiao et al. \(2021\)](#) used TF-IDF and sentiment scoring to effectively analyze Chinese environmental complaints. They highlighted the role of sentiment in understanding issue severity. Recent work demonstrated that transformer models like BERT can deliver excellent performance with low latency, making it a good fit for real-time systems. These studies help us choose BERT-based models for strong NLP categorization in this project.

2.2 Computer Vision for Evidence Detection in Images

The proposed system can analyze visual evidence from citizen-submitted photos. This sets it apart from systems that only use text. [Vrabie \(2025\)](#) showed the effectiveness of using a Faster R-CNN model to detect municipal problems in images, achieving 93 percent accuracy. This supports the use of deep learning for object detection. Additionally, [Mamta et al. \(2024\)](#) and [Devanathan et al. \(2024\)](#) promote a multimodal approach. Their research on the MMAspectX framework and MAACA model demonstrates that analyzing images and text together, such as using ResNet for visual features and BERT for text, works better than using either alone. This finding supports the integration philosophy of this project.

2.3 Geospatial Analysis and Duplicate Detection

A key challenge is locating and grouping complaints to find recurring issues. [Rammalho et al. \(2020\)](#) were the first to extract geospatial complaints from unstructured text, like TV news. They showed that geoparsing can work in this area. Their extraction success rate was moderate at 59.8 percent. This highlights the need for strong location processing. This project will tackle that by using direct geotags from user submissions. The idea of grouping and prioritizing is also clear in the work of [Peng et al. \(2022\)](#). Their "urban problem tree taxonomy" and social sensing method greatly reduced push failure rates. This shows the benefits of organizing complaint data effectively.

2.4 Complaint Prioritization and Dashboard Presentation

Merely categorizing complaints is not enough. Intelligent prioritization is key for effective resource allocation. [Vairetti et al. \(2024\)](#) tackled this by using a Spanish BERT model, BETO, for complaint prioritization. They achieved impressive accuracy at 92.1 percent and an AUC of 0.9785. Their success with an attention-based model strongly supports adding a similar prioritization layer based on severity, urgency, and frequency. For the front-end, [Munot et al. \(2022\)](#) developed an automated

classification dashboard for rural complaints. This study offers useful design ideas for creating an effective interactive dashboard for authorities.

2.5 Key Findings and Identified Gaps

While the existing literature is strong in addressing individual components like text, image, or location, few studies have successfully brought all three together in one complete process for civic issues. [Moya et al. \(2021\)](#) and [Devanathan et al. \(2024\)](#) found that using multiple types of data improves classification. However, their studies focused on urban reports and e-commerce videos, not a unified civic application. This project's design process, informed by foundational coursework in machine learning and software engineering, is motivated by this gap. It aims to combine the strengths of the cited works: the text classification by [Madyatmadja et al. \(2023\)](#) , image detection by [Vrabie \(2025\)](#) , geoparsing by [Ramalho et al. \(2020\)](#) , and prioritization by [Vairetti et al. \(2024\)](#) . The goal is to create a system specifically designed for effective civic issue management that is more powerful than its individual components.

2.6 Analysis of Existing Researches

Table 2.1 provides a review of academic papers about automated systems that analyze complaints and civic issues. The research is categorized by each paper's main contribution, such as new multimodal frameworks, datasets, or AI models, along with their recognized limitations. These limitations often point to future work that could improve accuracy, broaden scope, or include more effective techniques. The overall aim is to use AI and data mining to process feedback from sources like social media, urban reports, and videos for use in e-government, smart cities, and e-commerce.

While table 2.2 gives a technical summary of the models, performance metrics, and datasets used in different research papers on automated complaint and civic issue analysis. It highlights the specific AI models used, such as BERT, CNN, and Random Forest. The table includes their reported performance, like accuracy and

F1-score. It also details the datasets used for training and evaluation, including their source, size, and the types of data they contain. The table shows a trend of using more complex, multimodal models to achieve high performance on specialized datasets.

Table 2.1: Contributions and Limitations Analysis in Complaints Management System

Paper	Contribution	Limitations
Vrabie (2025)	End-to-end image processing pipeline for municipal issue reporting with speed improvements	Further speed optimization; expand issue types; improve robustness
Devanathan et al. (2024)	Tri-modal (text, audio, visual) analysis superiority for e-commerce complaint detection	Add rationale extraction; improve code-mixed video handling
Karren et al. (2024)	Contextual feedback boosts satisfaction, trust, and reuse in CUIs	Optimize feedback length; balance information vs. user overload
Mamta et al. (2024)	MAspectX dataset and MMAAspectX framework for aspect-based multimodal complaint mining	Add multiple causes; expand to more languages/domains
Putra et al. (2024)	Low-resource solution for rural complaint analysis using web data	Integrate advanced ML; improve unstructured data handling
Reddy (2024)	Tri-modal decision support system for claims assessment using Generative AI	Expand claim types; improve speed; explore Federated Learning
Vairetti et al. (2024)	Attention-based DL with MCDM for complaint prioritization	Extend to multiclass prioritization or ordinal regression
Madyatmadja et al. (2023)	First ML classifier comparison for Indonesian e-gov complaints	Lack of Indonesian text mining library; improve low-data categories
Peng et al. (2022)	Urban problem tree taxonomy for social sensing of complaints	Expand to more cities; implement incremental learning
Moya et al. (2021)	Multimodal features (text+images) improve urban report classification	Improve model accuracy
Jiao et al. (2021)	Framework for environmental complaint mining with keyword extraction and sentiment analysis	Larger corpus; add NER; build big data system
Kim et al. (2021)	Civic needs identification through big data analysis of 160k Seoul queries	Expand word relations; explore unsupervised methods
Madyatmadja et al. (2020)	Systematic review and technique suitability guide for e-gov complaint mining	Explore new and emerging techniques
Ramalho et al. (2020)	Brazilian TV news as source for geospatial complaint extraction feasibility	Improve geoparsing; compare topic modeling vs. ML classification
Kumar et al. (2019)	Civic Issue Graph datasets and no-label adaptation method using scene graphs	Extend to other domains; generate NL descriptions from graphs

Table 2.2: Preferred datasets, models and their performance in related Complaint Management Systems

Paper	Models	Performance	Dataset Info
Vrabie (2025)	Faster R-CNN, ResNet	93% accuracy; 7 sec/case (98.6% faster)	Municipal image data from Romania (2022). Labels: Pot-holes, Waste, Illegal Parking, Damaged Infrastructure
Devanathan et al. (2024)	MAACA (multimodal)	Complaint F1:87.44%; Aspect F1:86.54%	130 e-commerce YouTube videos. Text, Audio, Visual features
Mamta et al. (2024)	LLaVA, BERT, ResNet	Outperforms base-lines	MAAspectX corpus (4,966 annotated aspect terms)
Putra et al. (2024)	Cosine Similarity, TF-IDF	High precision/recall/F1	Web complaints from Pertumbukan Village
Madyatmadja et al. (2023)	kNN, RF, SVM, Adaboost	SVM accuracy: 89.2%	9,865 complaints from LAKSA app
Munot et al. (2022)	BERT	Accuracy=97%; Precision=0.83-1.0	Rural complaints text dataset
Peng et al. (2022)	SVM, CNN, LSTM, BERT	Push failure rate: 14.6% (from 73.8%)	12345-hotline data from Sanya, China
Moya et al. (2021)	Random Forest	Exp1: 68.5% accuracy (7 classes)	24,642 Zurich urban reports (2013–2020)
Jiao et al. (2021)	Jieba, TF-IDF, sentiment	Effective keyword extraction	2-year environmental complaints from Guangzhou
Kim et al. (2021)	Word2Vec+RF, Topic Model	70% accuracy	160,316 Seoul civic queries (2006–2017)
Madyatmadja et al. (2020)	NN, DT, SVM, TF-IDF	Classification techniques	Bandung e-complaint service data
Ramalho et al. (2020)	Geoparsing, NLP, topics	59.8% extraction success	1,007 Brazilian TV news videos
Kumar et al. (2019)	MotifNet, ADDA, Bi-LSTM	R@1=43.3; Precision=53%	485,927 complaints from FixMyStreet

2.7 Model Description

We performed model selection by evaluating three strong baseline models for classifying complaint texts using a consistent evaluation approach: DistilBERT, RoBERTa, and an SBERT (SentenceTransformer) embedding model paired with a lightweight neural classifier. All models were assessed using the same training data and metrics to ensure that any differences in performance were due to the models themselves, not variations in experimental setup.

2.7.1 DistilBERT

DistilBERT is a smaller and more efficient version of the BERT transformer model, created using knowledge distillation. It maintains a strong grasp of contextual language while greatly lowering the computational demands. By reducing the number of parameters and layers, DistilBERT allows for quicker training and inference, making it ideal for large-scale text analysis tasks and situations where resources are limited, like handling civic complaints and providing real-time decision support. DistilBERT facilitates effective edge deployment by reducing model size by around 40% while maintaining approximately 97% of BERT's accuracy with knowledge distillation [Dattasmita et al. \(2025\)](#). The balance between efficiency and performance makes DistilBERT a suitable baseline model for classifying large amounts of civic text, even though larger models like RoBERTa might achieve slightly better accuracy on more complex or nuanced language inputs.

2.7.2 RoBERTa

RoBERTa, which stands for Robustly Optimized BERT-Pretraining Approach, is an enhanced version of the BERT architecture aimed at better language representation. It is based on BERT but includes important changes in its pre-training process, like using larger datasets, implementing dynamic masking, and eliminating the Next Sentence Prediction (NSP) task. These transformer-based models like RoBERTa have achieved tremendous success in NLP Tasks due to their capacity to extract universal language representations from vast amounts of unlabeled text [Najeeb et al. \(2025\)](#). This strong optimization allows RoBERTa to effectively understand deep contextual details and long-range relationships. In real-world uses such

as analyzing civic complaints, it shows excellent results in classifying complaint texts because it can grasp contextual subtleties, informal language, and long-range dependencies. When compared to more efficient models like DistilBERT, which are more compact, RoBERTa generally provides better classification accuracy and more consistent confidence levels, making it a dependable option for decision-support systems that prioritize accuracy.

2.7.3 SBERT

SBERT is a tailored version of the BERT architecture that focuses on quickly producing independent, fixed-size sentence embeddings for tasks related to semantic similarity. Its Siamese network design allows for rapid text comparisons using cosine similarity, which makes it very scalable for uses such as clustering, searching, and identifying duplicates. SBERT uses a Siamese network architecture to generate fixed-size sentence-level embeddings and significantly outperforms BERT and other existing methods in producing meaningful sentence representations [Ortakci & Borhan \(2025\)](#). This efficiency is a significant benefit compared to models that need pairwise input. In real-world scenarios, SBERT shows a robust understanding of semantics, particularly in selecting evidence for fact verification; it showed a superior capability in capturing semantic similarity, with higher cosine similarity scores than TF-IDF methods [Gusdevi et al. \(2025\)](#). A well-known drawback of transformer models such as SBERT is their limit on the number of tokens for input texts. The design of SBERT strikes a balance between semantic depth and computational efficiency, making it a key component in text analysis workflows. Although precise fine-tuning may be necessary for detailed category differences, its pretrained embeddings offer a strong basis for various tasks, including identifying duplicate complaints and grouping themes in civic situations.

2.7.4 YOLO

In order to be useful in our system, there is a need for real-time identification of city problems using complaint images. According to [Diwan et al. \(2023\)](#), YOLO is appropriate for such tasks because it involves single-stage object detection, predicting the bounding box and class probabilities at one go. It is therefore fast for real-world applications. Balancing the foreground and background classes in YOLO

and handling small multi-scale objects for realistic images such as those in complaints is considered important tasks. This explains the basis for using the same for evaluation in terms of Average Precision for each class and the average Average Precision (mAP), as given by [Diwan et al. \(2023\)](#).

Chapter 3

Plans and Derivable

The project follows the sequential phases, starting with research and planning, which includes literature review and problem formulation. Figure 3.1 illustrates the project timeline from May 2025 to May 2026.

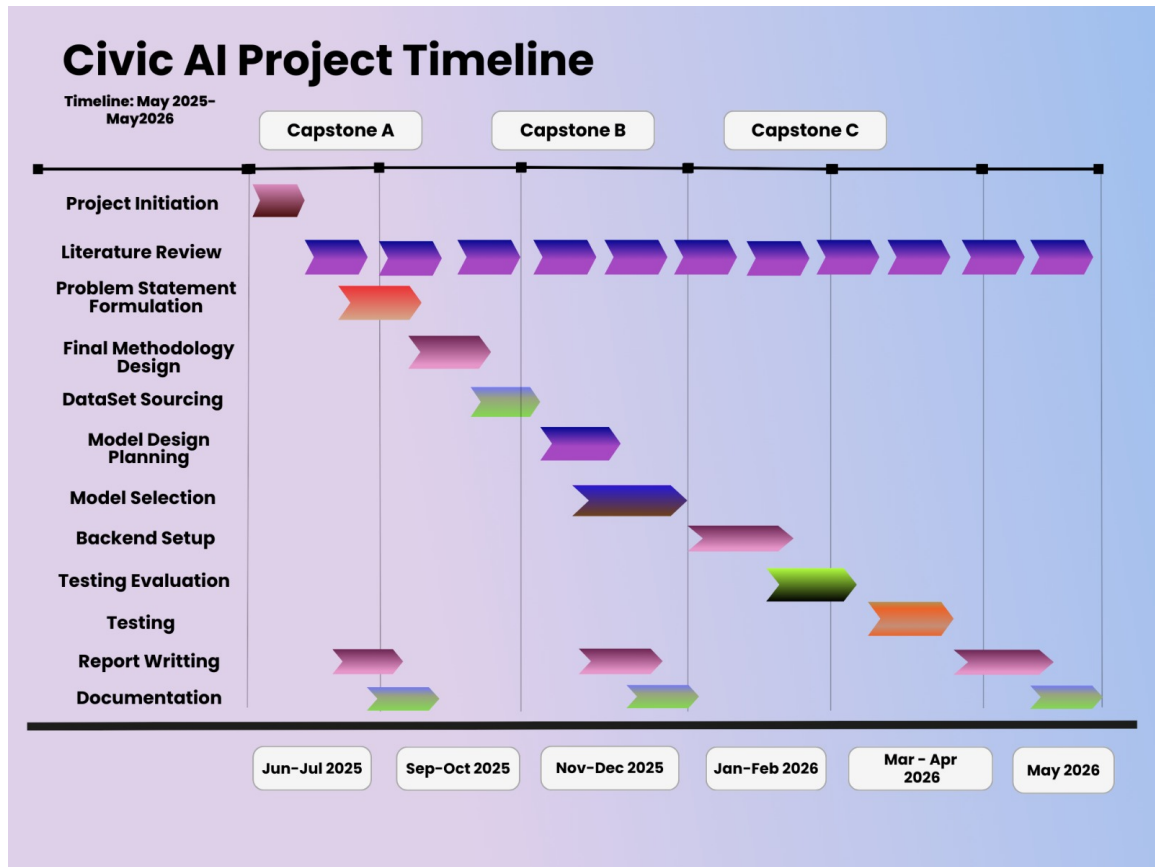


Figure 3.1: Project Timeline Gantt Chart showing all phases from initiation through deployment.

This will be followed by sourcing datasets, designing models, developing systems, testing, and finally, documentation.

3.1 Budget Table

Table 3.1 shows the estimated Budget Allocation for this Project Development and Research Activities:

Table 3.1: Estimated Project Budget Breakdown

No.	Category	Details	Cost (BDT)
1	Internet and Utilities	Approximate shared internet and electricity used during development and testing	24,000
2	Research Resources	Document preparation, referencing, and related software/tools	1,000
3	Documentation	Printing and report preparation	5000
4	Deployment Services	Frontend, backend, database, and email services used under free tiers	0
4	Domain Cost	Not purchased	0

From the table 3.1 we can see that the actual direct project cost was 30,000 BDT. This cost mainly covered Internet and utility expenses, Documentation costs, research-related resources. The main deployment services were run under free-tier plans.

3.2 Tools and Libraries

The deployment of the recommended civic complaint analysis system involved a development platform with capabilities that facilitate web application development, database management, and machine learning models. For the development of the system, testing, and executing the model, computers that are equipped with at least an Intel Core i5 processor and 8-16 GB of memory were used; for more computation-intensive tasks such as training, cloud-based platforms may be employed. The NLP functions in the AI modules of the system utilize PyTorch and the Hugging Face Transformer toolkit for the tasks classification of the complaints. The module for item recognition from the complaint images uses the Ultralytics YOLO model. While the backend services are built on the FastAPI platform, the complaints dataset, user details, and storage systems are handled by Supabase. The authorities' dashboard and the frontend web page are developed using Next.js and Tailwind CSS frameworks.

Chapter 4

Methodology

Our system involves a complete pipeline in an end-to-end multimodal approach to understand civic complaints and help authority. As per text data, the civic complaints text dataset will be initially preprocessed, followed by its evaluation via comparative model assessment and further usage for model selection and fine-tuning to obtain the final model that is saved as a text classifier. Meanwhile, the image dataset would go through preprocessing followed by YOLO detector training and saving its learned visual weights. These two models, once obtained and fine-tuned, will be stored in the model storage space. In turn, when receiving a complaint from the citizen, the system will use the text data of the complaint to infer text, while using the image data to detect using a YOLO detector. Furthermore, the system uses location information provided in the complaint to process it through the geo module. All three outputs from the modules are subsequently fused to produce complaint understanding, following which the system undertakes duplicate detection and generates signals related to priority calculation based on the fusion output. The Figure 4.1 diagram shows Methodology of this research.

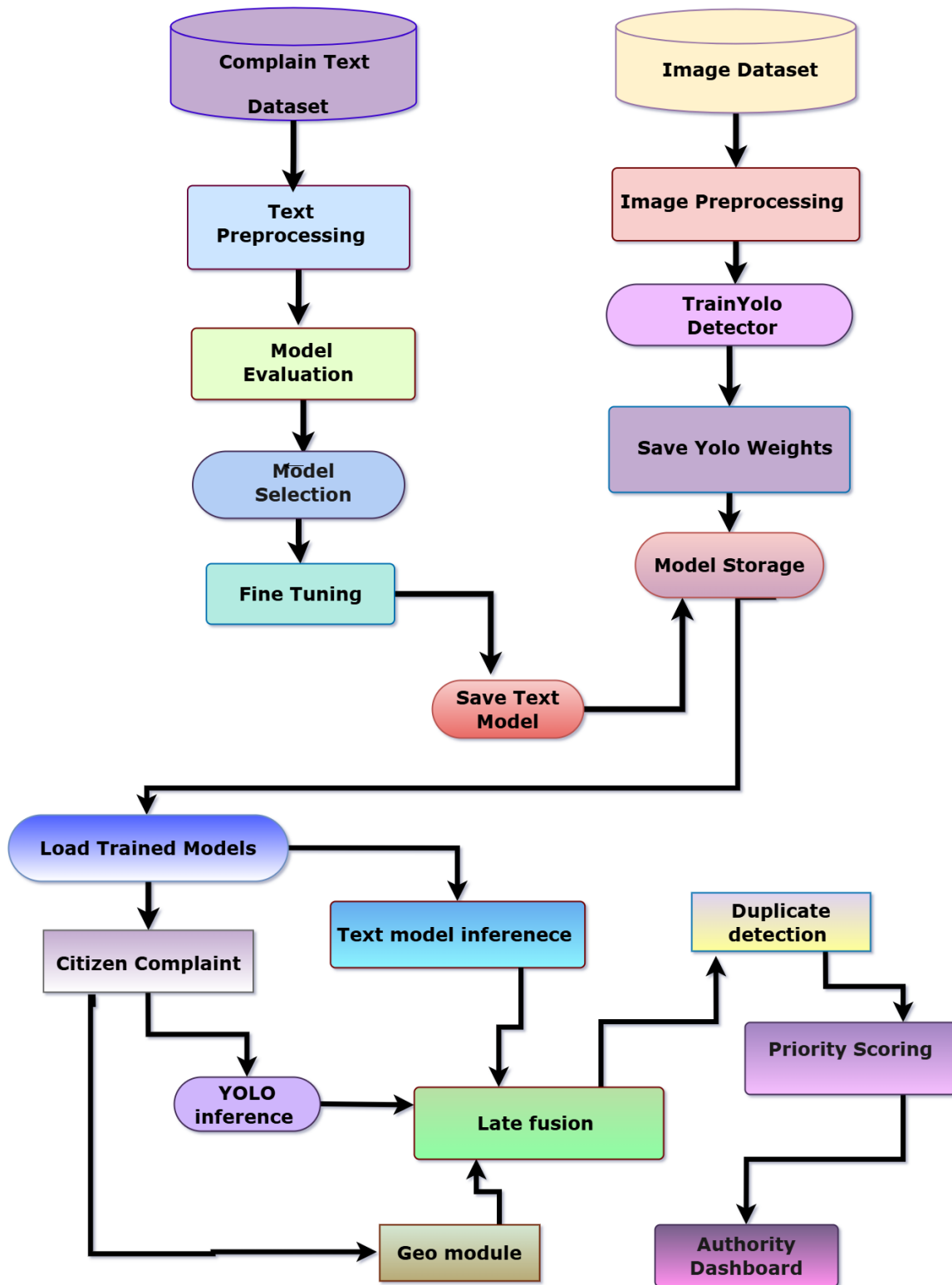


Figure 4.1: CivicAI Methodology Diagram

When someone reports an issue the system looks at what they wrote through one part of the system the picture they uploaded through another part called YOLO and where it happened through the geo part. The system then puts all this information together to understand the complaint. The system uses this information to check if someone already reported the same issue and to figure out how important it is. This information is then shown to the people in charge through a dashboard. The system does more than just categorize complaints. It looks at what people write checks for evidence in pictures uses location information avoids reports and helps make decisions based on priority all, in one process.

4.1 Complaint Text Dataset

In this study, we used two datasets for multimodal civic-complaint understanding. For the text modality, we used the "i-change-my-city-complaints-log—2019–2022" dataset from the OpenCity Urban Data Portal. It contains 11,000 complaint records with 18 columns, where each row represents one citizen complaint along with administrative and location metadata. For model input, we used the title and description fields, and the labels include 38 main categories (top 20 most frequent category shown in figure 4.2) and 168 sub-categories .

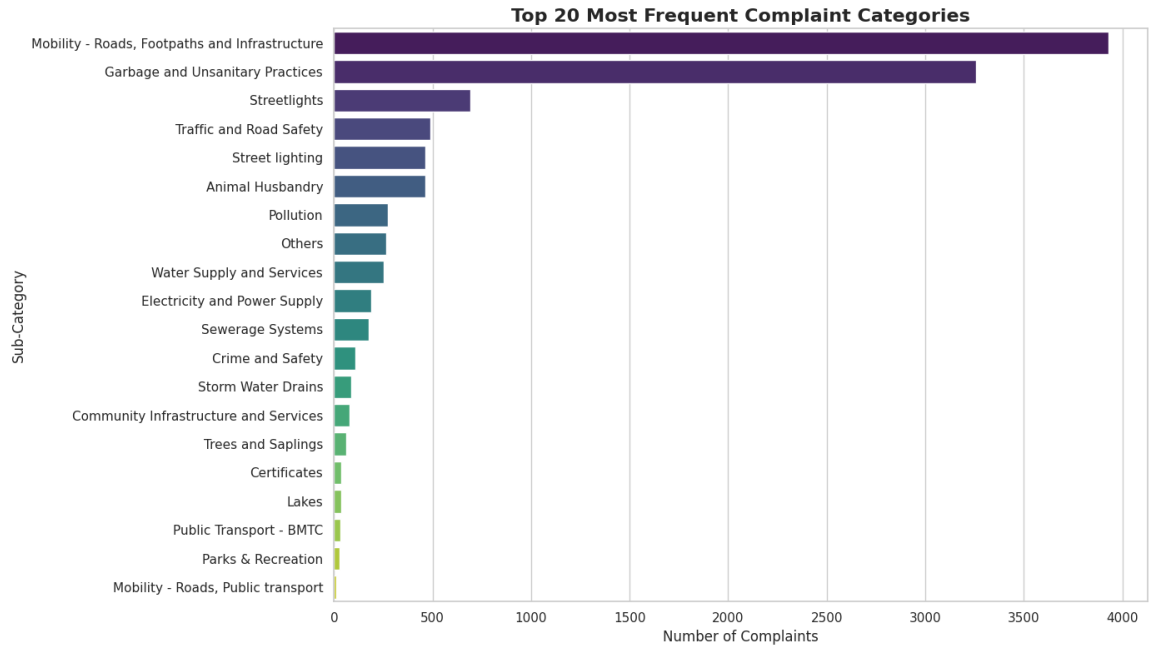


Figure 4.2: Visualization of Top category distribution in text dataset

4.2 Image Dataset

For the image modality, we used the Urban Issues Dataset from Kaggle, containing 47,140 images across 9 issue classes, already organized into train/validation/test splits. Since both datasets show class imbalance, we applied frequency-based filtering for rare text classes and used robust image loading and augmentation to improve training stability and reliability. Fig 4.3 shows overview of the image dataset with classes.

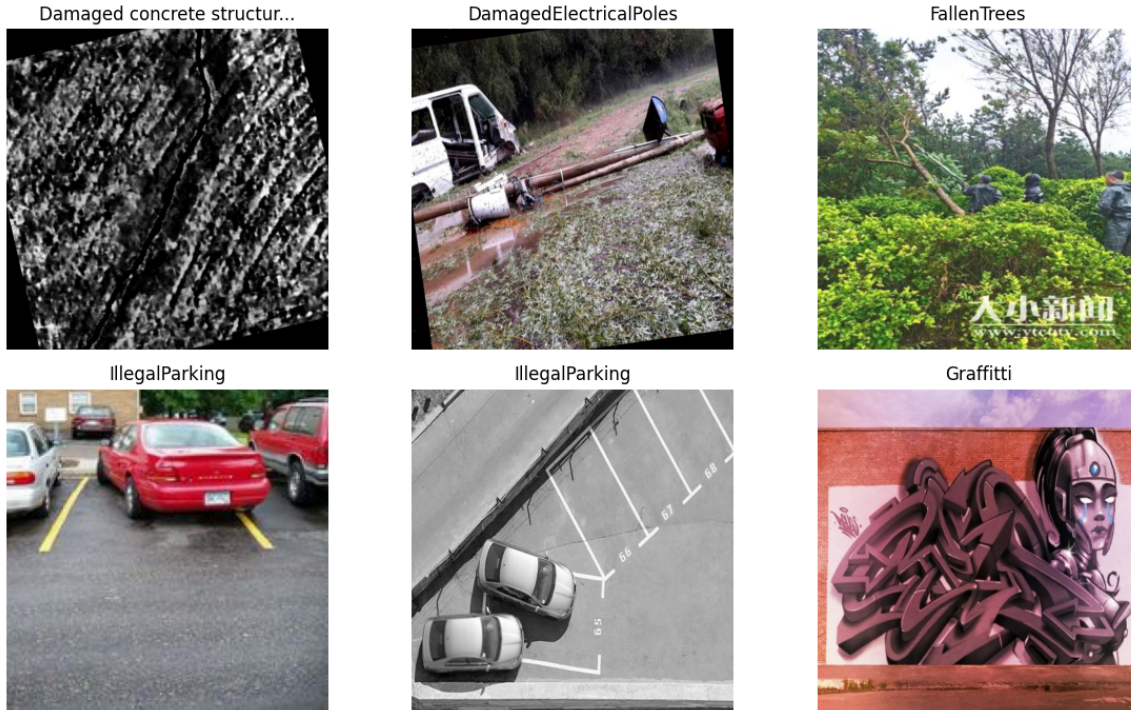


Figure 4.3: Overview of Images with class labels

Table 4.1 represent overall database summary. The image data is organized in a class-wise YOLO directory structure with splits for training and validation as well as testing sets.

Table 4.1: Overall Dataset Summary

Split	Total Images	Total Labels	Missing Labels
Train	38,086	38,084	2
Validation	4,894	4,894	0
Test	4,160	4,160	0
Total	47,140	47,138	2

Integrity checks of these directories verified that there is at least an image and a label directory in each class with splits for images and labels. There existed a total of 47,140 images and their corresponding YOLO labels in 47,138 files, which is close to full annotation coverage. There lacked only two image files in the "Garbage" class with their corresponding labels (approximate label coverage: 99.996%).

4.2.1 Text Dataset Cleaning and Preprocessing

We utilized a civic-complaint text corpus that includes a complaint title, description, and a supervised label (category title) for each record.

4.2.1.1 Text Normalization

Our initial preprocessing goal was to make sure that the raw text fields were safe and consistent for transformer tokenization and supervised learning. To do this, we converted both the title and description fields into valid string values, addressing any missing entries, and then normalized them to lowercase. This normalization is crucial as it ensures that any rule-based filtering works uniformly across different casing variations, preventing missed patterns due to capitalization differences.

4.2.1.2 Filtering Noisy and Ambiguous Phrases

After normalizing the text, we filtered out template-like or administrative entries by removing records where either the title or description contained ambiguous recurring phrases found in non-informative submissions. We implemented this filtering using substring matching and boolean masking, which allowed us to effectively exclude these records from the training corpus. The rationale behind this removal is that such entries act as noisy placeholders rather than providing meaningful descriptions of civic issues. Keeping them could lead the model to learn superficial patterns that do not accurately reflect actual complaints, ultimately harming its ability to generalize to real-world complaint texts.

4.2.1.3 Removing Label Ambiguity

Next, we focused on cleaning the labels to ensure that the supervised task aligned with complaint-management decision support. We removed three specific label strings: "Others," "Covid 19," and "Public Transport - BMTC" by directly filtering the label column. This step helps reduce label ambiguity and prevents the classifier from being trained on categories that are either too broad (like "Others") or irrelevant to the categorization framework of this study.

4.2.1.4 Merging Semantically Similar Categories

We also addressed label duplication by merging semantically similar category names, specifically combining “Streetlights” and “Street lighting” into one unified label. This consolidation helps the classifier avoid unnecessary differentiation between two labels that refer to the same civic issue, enhances category consistency, and minimizes fragmentation, which is particularly important in long-tail distributions where splitting already small classes can destabilize training and lower recall for that complaint type. To create the input for our model, we made sure that each training example included a single standard text field. When there wasn’t a clear text column available, we combined the title and description, using whichever field was present. This approach results in a single natural-language sequence for each complaint, aligning with the input format needed for transformer tokenizers and enabling the model to learn from both brief titles and more detailed descriptions.

4.2.1.5 Minimum-Sample Filtering Strategy for Stratified K-fold Cross-validation

After establishing a dependable text field, we tackled the issue of long-tail class distribution by implementing a minimum-sample filtering strategy. We analyzed the frequency of each class and tested various threshold values to determine how many categories and samples could be kept under each condition, while also checking the practicality of stratified cross-validation. This method requires that every class is represented in each fold. Initially, we had 11000 complaints across 34 categories, but the distribution was heavily skewed: some categories had very few samples, and the rarest category had only one sample (for instance, EMP Grievance Redressal). This is significant because stratified K-fold cross-validation necessitates that every class is present in every fold; thus, the maximum number of valid folds is limited by the smallest class size.

Stratified K-fold cross-validation (SKCV) is an evaluation technique designed to handle imbalanced datasets by ensuring that each fold maintains the same class distribution as the original dataset. This approach is particularly important in classification tasks where certain classes are underrepresented, as it prevents scenarios where minority classes may be absent from one or more folds, which could lead to biased performance estimates. The stratified k-folds cross-validation splits the data set on k folds such that every fold has (nearly) the same percentage of samples from

minority and majority classes as the complete set [Szeghalmy & Fazekas \(2023\)](#) . Furthermore, SKCV guarantees that each class is evenly distributed throughout the k-fold. [Widodo et al. \(2022\)](#). By preserving class proportions across folds, stratified cross-validation reduces variance in performance metrics, provides more reliable estimates of model generalization, and ensures that both majority and minority classes are consistently represented during training and validation. With a minimum class frequency of one, it becomes nearly impossible to implement stratified K-fold beyond K = 1, which is not suitable for reliable evaluation. To tackle this issue, we performed a threshold impact (Table 4.2) analysis by testing different minimum-sample thresholds.

Table 4.2: Effect of Minimum Sample Threshold on Dataset Filtering

Threshold	Categories Kept	Samples Kept	Data-Retained (%)	Categories Removed	Max Valid K
0	34	10,679	100.00	0	1
2	25	10,670	99.92	9	3
3	25	10,670	99.92	9	3
5	22	10,660	99.82	12	5
10	18	10,637	99.61	16	11
15	16	10,613	99.38	18	28
20	16	10,613	99.38	18	28
30	15	10,585	99.12	19	38
40	13	10,509	98.41	21	62
60	13	10,509	98.41	21	62

We looked at how many categories and samples were left after filtering, the volume of data we retained, and the possible values for K. Based on our findings, we decided on MIN SAMPLES = 10, which removed very rare categories while preserving nearly all the data . After the filtering process, our dataset consisted of 10,637 samples and 18 categories, providing a statistically stable label space for both training and evaluation. The remaining categories mainly centered on a few key civic issues, such as Mobility - Roads, Footpaths and Infrastructure, and Garbage and Unsanitary Practices. We also kept smaller but relevant categories like Public Toilets and Mobility - Roads, Public Transport, as they satisfied the minimum sample requirements.

4.2.2 Image Data Preprocessing

For the purpose of explicit examination of the annotation format before training, we considered the fact that YOLO-style data is present but could be in various labeling formats. To check whether these annotations are done in polygon/segmentation coordinate format or are in YOLO bounding box formats, we analyzed the configuration files related to the YOLO structure (such as the definition files for classes and splits) and took samples from the data files. Since detection learning requires uniform representation of the labels, coexistence of polygon format and bounding box formats may always end up in misleading signals or unparsable files. To ensure that the data is processed perfectly in the next steps, we calculated the number of data lines that appeared as polygon annotations (more than five numeric entries) as opposed to data lines representing normal bounding box annotations (five entries).

4.2.3 Train/Test Split

4.2.3.1 Text Dataset

For text dataset an 80/20 split is applied, where the 20% test set is reserved for final evaluation and is not used during cross-validation or model selection. All cross-validation experiments take place on the 80% training portion.

4.2.3.2 Image Dataset

Image dataset was well organized with class label. Table 4.3 shows the per-class dataset integrity and split distribution of the image dataset.

Table 4.3: Per-class Dataset Integrity and Split Distribution

Class	Train	Val	Test	Total Images	Total Labels	Missing
Damaged Concrete Structures	9,315 / 9,315	1,228 / 1,228	1,222 / 1,222	11,765	11,765	0
Damaged Electrical Poles	7,271 / 7,271	513 / 513	328 / 328	8,112	8,112	0
Damaged Road Signs	2,267 / 2,267	43 / 43	39 / 39	2,349	2,349	0
Dead Animals Pollution	172 / 172	32 / 32	16 / 16	220	220	0
Fallen Trees	8,500 / 8,500	982 / 982	1,632 / 1,632	11,114	11,114	0
Garbage	3,133 / 3,131	660 / 660	337 / 337	4,130	4,128	2
Graffiti	1,704 / 1,704	214 / 214	210 / 210	2,128	2,128	0
Illegal Parking	57 / 57	3 / 3	7 / 7	67	67	0
Potholes and Road Cracks	5,667 / 5,667	1,219 / 1,219	369 / 369	7,255	7,255	0

For the purpose of explicit examination of the annotation format before training, we considered the fact that YOLO-style data is present but could be in various labeling formats. To check whether these annotations are done in polygon/segmentation coordinate format or are in YOLO bounding box formats, we analyzed the configuration files related to the YOLO structure (such as the definition files for classes and splits) and took samples from the data files. Since detection learning requires uniform representation of the labels, coexistence of polygon format and bounding box formats may always end up in misleading signals or unparsable files. To ensure that the data is processed perfectly in the next steps, we calculated the number of data lines that appeared as polygon annotations (more than five numeric entries) as opposed to data lines representing normal bounding box annotations (five entries).

4.2.4 Applying Cross-Validation to Evaluate Text Classifier Models

Each fold will train on about 7.56k samples and validate on approximately 0.95k samples, maintaining the proportions of the samples. This strong optimization allows RoBERTa to effectively understand deep contextual details and long-range relationships. In real-world uses such as analyzing civic complaints, it shows excellent results in classifying complaint texts because it can grasp contextual subtleties, informal language, and long-range dependencies. When compared to more efficient models like DistilBERT, which are more compact, RoBERTa generally provides better classification accuracy and more consistent confidence levels, making it a dependable option for decision-support systems that prioritize accuracy. A significant aspect of our evaluation was the use of Stratified K-fold cross-validation on the training data (80%), while keeping a separate 20% test set untouched for an unbiased final evaluation.

Cross-validation offers a more accurate estimate of a model's generalization ability by minimizing reliance on a single train/validation split and measuring performance variability across different folds. The number of folds (K) was automatically determined based on the smallest total sample count per class in the training data, ensuring that stratified splitting was possible for all classes. For each metric (m), we summarized the results from each fold using the mean and standard deviation. These statistics allowed us to compare the models based on both their average performance and their stability across the folds.

4.2.5 Unified YOLO Detection Dataset Construction For Image Data

A unified detection dataset was created by combining the nested, class-separated directory structure into a typical YOLO dataset architecture with distinct train, validation, and test splits for both photos and labels in order to facilitate single-model training across all visual categories. Two practical difficulties were resolved with this consolidation: possible filename clashes between several class folders and inconsistent label formats.

Converting any polygon/segmentation annotations into axis-aligned bounding boxes so that they could be used with YOLO detection training. Each label line begins with a class ID followed by coordinates. If the label was already in normalized YOLO bounding-box form, it was retained right away. If the label included polygon vertices, a computation for the tightest bounding box that contained the polygon and converted it into YOLO format was applied. Degenerate boxes with non-positive width or height were removed to prevent invalid supervision. After that safe, unique output filenames (e.g., by prefixing with class identifiers and sanitizing names) was generated to enable collision-free dataset merging because identical filenames can exist across distinct class folders. This keeps all image-label pairs perfectly aligned after consolidation and stops quiet overwriting.

Empty label files were produced, which are acceptable in YOLO training (representing images with zero objects), in situations where images were present without matching label files. In order to maintain the preprocessing pipeline’s audit ability and reproducibility, we monitored important metrics during dataset building, such as the total number of photos processed, missing labels, and the number of polygon annotations translated. A unified YOLO dataset configuration file was produced that specified the number of classes, split pathways, dataset root, and class-name list. Without using class-specific sub-folders, direct training and evaluation under the Ultralytics YOLO framework were made possible by this uniform arrangement.

4.2.6 YOLOv8 Model Training Procedure

Using Ultralytics YOLOv8 initialized using pretrained weights, we trained the detector. In order to balance computational efficiency and detection performance, we chose the YOLOv8n (nano) variation, which is appropriate for quick training and

iteration in GPU-constrained situations. In order to increase speed while preserving a respectable detection capability, training was carried out with a lower input resolution. A lightweight epoch count and a batch size selected to optimize GPU efficiency were used in the training schedule; changes were made when memory limitations necessitated a smaller batch. We deactivated validation during each epoch and carried out evaluation as a separate step after training was finished in order to speed up training. This allowed for a thorough final evaluation on the test split while lowering training overhead. To facilitate their future use in multimodal fusion and deployment, the training weights were stored in persistent storage. The Ultralytics YOLOv8 object detector was used to construct the visual evidence module. The lightweight YOLOv8n backbone was used for training and validation, with two dataloader workers and an input resolution of 416x416. Ten epochs of the experiment were conducted. To increase resilience under real-world image noise, standard YOLO augmentations were used, mosaic augmentation was first enabled and then disabled ("closing dataloader mosaic"), and light instrumentation transformations (blur, grayscale, and CLAHE) were utilized. The detector demonstrated steady convergence with monotonically declining loss components (box loss, classification loss, and distribution focal loss) over the course of the ten epochs.

4.2.7 Multimodal Fusion and Consistency Checking

Multimodal inference layer fuses the input complaint text with its associated images, wherever both sources are present. While the textual branch of our model is capable of predicting labels for all categories, the visual part supports a portion of the civic categories only. This makes it important to align the two modalities into one civic space before their fusion.

4.2.7.1 Shared-space alignment of text and image evidence.

While the text classifier operates on the set of original text labels, the image classifier provides support only for certain civic issue classes. Thus, before performing data fusion, the predictions from the text classifier need to be mapped onto the shared civic issue set. Similarly, the predictions from the image classifier need to be mapped onto the same shared set.

This process guarantees that fusion does not take place between mismatching label

categories. For instance, assume that the text classifier identifies one of the initial 18 text labels such as *Drainage and Water Logging*, while the image detector recognizes only one of the common civic labels such as *Community Infrastructure and Services*. Then, the text label must be first converted to its equivalent common label prior to proceeding with the fusion process. Therefore, both text and image labels will be from the same semantic label category space at the decision stage.

4.2.7.2 Fusion decision and safe fallback.

All complaints are not necessarily fused multimodally. The complaint is that if belonging to a class that is by nature text-heavy and which cannot get meaningful support from images in the shared civic space setting, the model outputs the text-based predictions as the final outcome. Should the combination of both types of evidence exist, then fusion is performed on the shared-space text evidence and shared-space image evidence using weighted late fusion. Where no evidence from the image exists, the text result will be obtained by the framework.

The safe fallback option ensures that the framework does not force a fusion in situations where image evidence does not add any value. For example, a complaint like “*The area has been suffering from repeated power cuts since last night*”. This type of complaint might not give any useful information from the accompanying dark picture, therefore, keeping text-based predictions is the safer option.

4.2.7.3 Weighted late fusion.

Upon having both the shared space text distribution as well as the shared space image distribution, the system uses weighted late fusion as follows:

$$p_{i,k}^{(f)} = \frac{w_t \tilde{p}_{i,k}^{(t)} + w_v \tilde{p}_{i,k}^{(v)}}{w_t + w_v}, \quad k \in \mathcal{K}_8, \quad (4.1)$$

where $w_t > 0$ and $w_v > 0$ denote the weight of the text and image respectively. In this case, $\tilde{p}_{i,k}^{(t)}$ refers to the shared-space text probability for the complaint i and class k , $\tilde{p}_{i,k}^{(v)}$ is the shared space image probability, and $p_{i,k}^{(f)}$ refers to the fused probability after fusing both modalities. The output class for the complaint is the one that receives the maximum score, and the maximum of the category scores is the measure of fusion confidence. This strategy is consistent with multimodal decision level fusion strategies mentioned earlier in the literature [Meel & Vishwakarma](#)

(2021).

Here, i is the index for the specific complaint, while k is the index for the specific class within the civic label space $\mathcal{K}_8$. Weights w_t and w_v are used to balance the effect of the two branches. Specifically, if $w_t = w_v$, then both branches contribute equally; otherwise, when $w_t > w_v$, the textual branch has more contribution than the visual one.

Now let us assume that the i complaint includes the text “*Garbage has piled up beside the market and is causing bad smell*”, and the uploaded image also shows visible waste accumulation. Assume that for the shared class *Garbage and Unsanitary Practices*, the text branch produces $\tilde{p}_{i,k}^{(t)} = 0.80$, the image branch produces $\tilde{p}_{i,k}^{(v)} = 0.60$, and equal weights are used, i.e., $w_t = w_v = 1$, then -

$$p_{i,k}^{(f)} = \frac{1 \times 0.80 + 1 \times 0.60}{1 + 1} = 0.70.$$

Hence, the combined probability for this class will be 0.70. This will classify the compliance as *Garbage and Unsanitary Practices* provided 0.70 is the highest amongst the probabilities in all common classes.

4.2.7.4 Consistency Checking

Comparison of predictions for text and image is another proof that makes complaint classification interpretable for review by the authorities. Namely, if the predicted civic classes coincide in the branches dealing with the shared space text and image data, then the complaint is treated as modality-consistent, and vice versa. This rule cannot be used as a classifier, but serves as an interpretative criterion that allows us to find out if both branches lead to one interpretation of the civic problem.

If both branches classify the complaint under *Traffic and Road Safety* then the classification is considered to be modality-consistent. If, on the other hand, the text branch classifies the complaint as *Garbage and Unsanitary Practices* while the image branch classifies the same complaint as *Community Infrastructure and Services*, the classification is deemed inconsistent.

4.2.8 Decision Support Modules

Other than classifying complaints, the suggested solution also produces authority-oriented outputs which are helpful in reviewing and filtering of those complaints.

These include estimating urgency based on sentiment, prioritizing complaints, identifying duplication, spotting hotspots, and generating operations. Only the mathematical components have been formalized by equations below, while the other phases are just stated.

4.2.8.1 Sentiment scoring with VADER

In order to quantify negativity in a complaint expressed in natural language, the complaint text is subjected to the process of sentiment analysis using the VADER model. The compound sentiment score for complaint i is represented as

$$s_i = \text{VADERCompound}(x_i), \quad s_i \in [-1, 1], \quad (4.2)$$

where x_i denotes the complaint text itself. Since complaints are usually short, straightforward, and highly emotional in nature, polarity scoring using lexicons and rules is operationally feasible for decision-making purposes, hence the use of VADER [Hutto & Gilbert \(2014\)](#).

In this case, x_i stands for the complaint text as a whole, consisting of the title and description, whereas s_i refers to the polarity score obtained. The closer a score is to -1 , the more negative the language used, while scores near 0 and $+1$ represent neutral and more positive language usage, respectively.

In the case of the complaint text, *"The road is badly damaged and extremely dangerous for pedestrians."* Since this sentence includes negative words like *badly damaged* and *dangerous*, the VADER algorithm can give the sentence a negative polarity score of $s_i = -0.80$, which will be used as the basis for urgency estimation.

4.2.8.2 Urgency normalisation from sentiment

This sentiment score is normalized in order to obtain an urgency bound where the complaint text yields a high urgency level. Urgency bound is determined as follows:

$$U_i = \frac{1 - s_i}{2}, \quad U_i \in [0, 1]. \quad (4.3)$$

With this transformation, very negative complaint texts will generate urgency scores close to 1, neutral complaints will stay close to 0.5, and increasingly positive complaints will have lower urgency scores. Thus, the range of the urgency score will be kept bounded, while ensuring that the directionality of the score makes

sense for ranking civic complaints [Rajkumar et al. \(2026\)](#).

U_i denotes the urgency score of complaint i , since $s_i \in [-1, 1]$, the transformed value will lie in the interval $[0, 1]$, it is better suited for ranking processes in the future. If the sentiment score is $s_i = -0.80$, then

$$U_i = \frac{1 - (-0.80)}{2} = \frac{1.80}{2} = 0.90.$$

This indicates that the complaint has a high urgency score of 0.90, indicating that it is operationally significant due to its very negative phrasing.

4.2.8.3 Priority ranking using Simple Additive Weighting (SAW)

The framework uses the Simple Additive Weighting (SAW) method to integrate several criteria into a single priority score in order to assist triage. The criterion vector for complaint i is expressed as

$$x_i = (U_i, C_i^{(f)}, H_i), \quad (4.4)$$

where U_i represents the urgency level, $C_i^{(f)}$ indicates the confidence of the fusion, and H_i refers to the number of hours after the arrival of the complaint. However, other factors, such as the frequency of local recurrence, may be considered based on different configurations, but the SAW process stays consistent. Normalization process is

$$r_{ij} = \begin{cases} x_{ij} / \max_i x_{ij}, & \text{if criterion } j \text{ is a benefit criterion,} \\ \min_i x_{ij} / x_{ij}, & \text{if criterion } j \text{ is a cost criterion.} \end{cases} \quad (4.5)$$

Next, the final SAW score is calculated by

$$V_i = \sum_{j=1}^m w_j r_{ij}, \quad (4.6)$$

where the weight of criterion j is denoted by w_j . Lower values of V_i are used to prioritize complaints; higher scores indicate more importance [Panjaitan \(2019\)](#).

In the SAW formulation, the raw value of complaint i under criterion j is denoted by x_{ij} the normalized value is denoted r_{ij} and the weight applied to that criterion is denoted by w_j . As a result, the final score V_i is a weighted total of the normalized evidence for each active priority criterion. Assume complaint i has

$$U_i = 0.90, \quad C_i^{(f)} = 0.70, \quad H_i = 12.$$

Suppose, within the current active complaint pool, the maximum urgency is 1.00, the maximum fusion confidence is 0.90, and the maximum elapsed time is 24 hours. If all three criteria are treated as benefit criteria for illustration, then

$$r_{i1} = \frac{0.90}{1.00} = 0.90, \quad r_{i2} = \frac{0.70}{0.90} \approx 0.78, \quad r_{i3} = \frac{12}{24} = 0.50.$$

If the criterion weights are chosen as

$$w_1 = 0.5, \quad w_2 = 0.3, \quad w_3 = 0.2,$$

then the SAW score becomes

$$V_i = (0.5)(0.90) + (0.3)(0.78) + (0.2)(0.50) = 0.45 + 0.234 + 0.10 = 0.784.$$

This value is then compared with the scores of other complaints. A complaint with a larger V_i will appear higher in the review queue.

4.2.8.4 Threshold-based escalation

Only priority ranking does not ensure promptness in action. Thus, the model also employs a threshold-based escalation policy for pending complaints. Let T_h be the escalation threshold set in hours. The escalation condition is expressed as

$$E_i = \begin{cases} 1, & \text{if } t_i^{(c)} - t_i^{(s)} > T_h \text{ and } \text{status}_i \neq \text{Resolved}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.7)$$

where $t_i^{(c)}$ is the current time, while $t_i^{(s)}$ is the time of submission. Such an approach will serve as a safety net for complaints by indicating those complaints which have exceeded the allowable response period [Rajkumar et al. \(2026\)](#).

An indicator of escalation is the variable E_i . The complaint should be considered past due and brought to attention if $E_i = 1$. The complaint stays within the permitted response window if $E_i = 0$.

Let us assume that the complaint was submitted at 8:00 AM on Monday and the current time is 10:00 AM on Thursday. The total number of hours between these two moments is 74. For example, given that the threshold $T_h = 72$ hours, then

$$74 > 72 \quad \Rightarrow \quad E_i = 1.$$

Because the complaint has gone over the response threshold, it has been escalated.

4.2.8.5 Geolocation hotspot detection

Density-based clustering is used to find dense local groups after the geographic locations of complaints are compared using the Haversine distance in order to find spatial clusters of civic complaints. Between two complaints i and j , the great-circle distance is

$$d_{ij} = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_i) \cos(\phi_j) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right), \quad (4.8)$$

where $\Delta\phi = \phi_j - \phi_i$ and $\Delta\lambda = \lambda_j - \lambda_i$ in radians, r refers to the radius of the earth, ϕ represents the latitude, and λ represents the longitude. DBSCAN can be used after obtaining the matrix of pair-wise distances between locations [Ester et al. \(1996\)](#).

d_{ij} represents the distance between the two complaints on the surface of the earth. A Euclidean distance does not make sense in this situation since geographical coordinates are in use.

Assuming that there are two complaints coming from adjacent roads, where the distance between them is $d_{ij} = 35$ meters. Given the DBSCAN's radius parameter value of 100 meters, the two complaints will be classified as neighboring points in space. The presence of a large number of such neighbors in a certain location would prompt DBSCAN to cluster them into one hot spot. On the other hand, being distant from all the other complaints makes the particular complaint noise.

4.2.8.6 Duplicate detection

Semantic, physical, and temporal proximity between complaints are used to identify potential duplicates. Let the embedding vectors of complaints v_i and v_j be denoted by i and j . Then the cosine similarity between the two is given by

$$\text{sim}(i, j) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}. \quad (4.9)$$

Two complaints are only deemed potentially duplicates when all three of the following criteria are met at the same time:

$$\text{sim}(i, j) > \theta_{\text{text}}, \quad d_{ij} < \theta_{\text{loc}}, \quad |t_i - t_j| < \theta_{\text{time}}. \quad (4.10)$$

This combination lessens the possibility of connecting complaints that describe different local incidents or events but are only topically similar [Salton & McGill](#)

(1983).

In this case, $\text{sim}(i, j)$ is a metric that captures the similarity between two pieces of text, while d_{ij} captures geographical distance, and $|t_i - t_j|$ captures time difference. The thresholds θ_{text} , θ_{loc} , and θ_{time} define the degree of similarity, proximity, and recentness required for two complaints to be deemed duplicates.

Let the complaints be i says “*Large pothole in front of the school gate*” and complaint j says “*Road hole near school entrance causing traffic*”. If their text embeddings result in

$$\text{sim}(i, j) = 0.91,$$

their spatial distance is

$$d_{ij} = 32\text{meters},$$

and their time difference is

$$|t_i - t_j| = 3\text{hours},$$

then under thresholds

$$\theta_{\text{text}} = 0.85, \quad \theta_{\text{loc}} = 50\text{meters}, \quad \theta_{\text{time}} = 24\text{hours},$$

Each of the three requirements is met. The two complaints are therefore marked as possible duplicates.

4.2.8.7 Operational summary generation and dashboard output

In addition to the mathematically defined decision-support signals above, the framework is also able to produce complaint summaries for dashboard presentation. In this way, the complaint summary is constructed by the complaint description, its predicted category and the strongest visual evidence associated with the complaint. Finally, the processed outputs are prepared for display on a dashboard where authorities are able to see information about the category of complaints, the level of confidence, the urgency of the complaint, its priority status, duplicates, and hotspots in one place.

4.2.9 Evaluation Metrics

The text model is assessed by the following metrics: Accuracy, Macro F1, Weighted F1, Balanced Accuracy, Macro Precision, and Macro Recall. The former stressed the

minority class, while the latter stressed the majority class performances. The image detector is assessed by the Box Precision, Box Recall, mAP@0.5, and mAP@0.5:0.95 metrics.

4.2.10 Implementation Details

The development and training of models were conducted largely in Google Colab, taking advantage of GPU support using a Tesla T4 environment. For the text classification pipeline, the use of Hugging Face Transformers based on PyTorch was chosen. For the visual evidence pipeline, Ultralytics YOLOv8 was selected; weights of the trained models were saved to the cloud storage for later inference and application.

In terms of prototype deployment, the web-based multi-modal complaint analysis platform was used. In the back end, FastAPI served as the API layer where inference was performed using PyTorch, Ultralytics YOLO, VADER, and the cosine similarity algorithm; clustering based on DBSCAN was performed for spatial analysis purposes. Supabase was leveraged to store the data on complaints and to conduct user authentication. The citizen-side and authority-side UIs were constructed using Next.js with Supabase back-end support. Spatial visualisation of hotspots was performed using Leaflet together with the OpenStreetMap base maps.

Chapter 5

Design and Development

This chapter presents information about the design and development of the proposed CivicAI framework. Unlike other chapters which deal with the methodology, model creation, and analysis, this chapter aims at discussing how the entire system was developed. This chapter discusses the architecture of the proposed solution, major frontend and backend components, features of the proposed website, design of the user interface, test plan and its execution, as well as issues related to deployment. Therefore, the goal of this chapter is to demonstrate the transformation of trained multimodal models into an efficient complaint management system.

5.1 Functional Requirements

Functional requirements outline the primary functionalities the system is expected to provide in support of its usage scenario. For the proposed platform CivicAI, the initial functional requirements are as follows:

- **Citizen complaint submission:** The citizen should be provided with the ability to submit a complaint using textual information, optionally an image, and location data. The complaint submission process should gather important information about the reporter, complaint title, description, selected category, and geolocation.
- **Multimodal complaint analysis:** As soon as a complaint has been submitted, the system should be capable of processing the complaint text via the text classification model and the image via the visual evidence detector. Finally,

the system should combine the results in terms of the multimodal inference pipeline to generate the final AI-supported evaluation of the complaint.

- **Authority-side complaint review:** The authority should have the ability to examine each complaint via a special interface that would show the complaint category under consideration, confidence level, reliability features, image evidence, summaries, complaint urgency, queue position, complaint duplication information, and other data related to the reviewing process.
- **Complaint monitoring and analytics:** It is crucial that the system offers monitoring and analysis dashboards to allow officials to see the distribution of complaints by categories, the pattern of complaints, geographic hot spots, issue recurrence, linked duplicates and insights regarding operational reviews. The system must enable both complaint-level and monitoring-level analysis of complaints.
- **Complaint organization and filtering:** It is essential that the interface for officials includes tools for searching and filtering complaints and drilling down to see complaints by categories, geographic area, complaint type, operational process, and other derived criteria like issue recurrence or duplicates.
- **Geospatial visualization:** It is essential that the system be able to visualize complaints on maps to allow officials to find geographic hot spots and area-level issue concentration.

5.2 Non-Functional Requirements

Apart from being able to perform the desired functions, the system is subject to some non-functional constraints.

- **Usability:** Because of the dual audience the system has, it needs to maintain its usability. It means that citizens will not face difficulties submitting complaints and, on their end, authorities can easily review the results produced by AI algorithms without the need to have any expertise in machine learning.
- **Performance:** The interaction with the system regarding complaint submission, complaint retrieval, and complaint review needs to be fast enough.

Although the system does not aim at being used operationally right now but rather as a prototype, it should still be responsive when dealing with project-level amounts of requests.

- **Reliability and Robustness:** The application needs to remain working even if only partially filled complaints arrive at the system. Namely, the complaint analysis algorithm should work in cases when no images or low-quality images are provided and proceed with text analysis.
- **Maintainability:** The project needs to be designed in a modular fashion such that changes can be made separately to the front end, back end, database, and AI. This is particularly essential when there is a need for future improvement to any of these aspects.
- **Security and Data Integrity:** Since the system deals with sensitive user data, it needs to ensure their security by using access control, data structure integration with databases, and other measures related to backend systems. The data consistency should not only be maintained in any case but should also be protected from modifications.
- **Explainability:** The explainability is very important in this case because the project is designed in such a way that authorities make decisions based on output from the system. Thus, they need to understand the decision-making process using textual, visual, fusion, and context-based review signals.

5.3 System Architecture

Figure 5.1 represents the architecture of CivicAI is designed in a layered manner that ties the processes of user interactions, data handling, AI-based inference, and authority-related processes into one workflow. In this case, the user interacts through the citizen-focused interface, submitting complaints with textual information, possible images, and geo-localization data, which are subsequently saved to the application database and further sent to the backend processes.

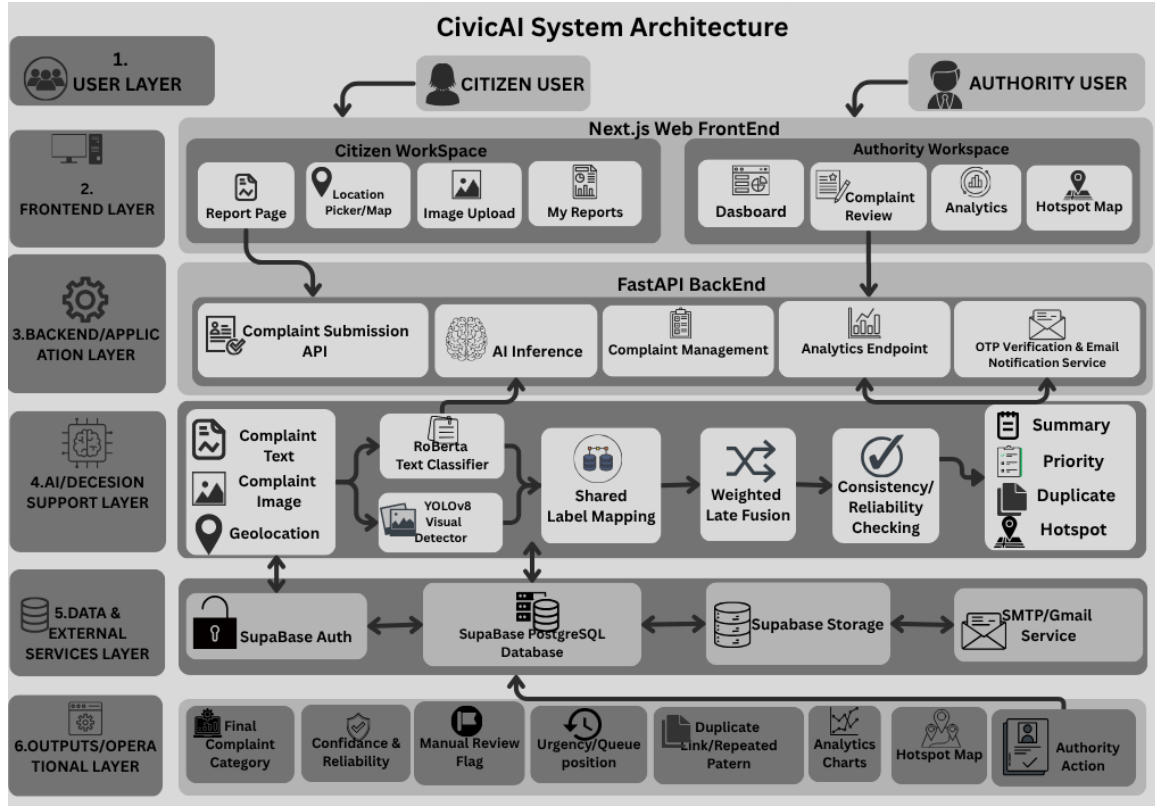


Figure 5.1: CivicAI System Architecture

Within the application layer, the processes of complaint retrieval, inference, updates, and authority-related decision making are managed by the backend. Namely, text-based complaints are handled by the fine-tuned RoBERTa classifier, whereas the images included in the complaint are managed by the YOLOv8 visual detector. Afterward, the results of the two inference branches are merged by using multi-modal fusion. To improve the results obtained by the fusion process, additional processes of urgency evaluation, duplicate detection, repetitive patterns detection, and geospatial hot spots identification are implemented.

The end results, on the other hand, are presented via authority-related interfaces such as the complaint reviews interface, the dashboard, analytics workspace, the hotspots map, and complaint management. In this respect, the system architecture goes beyond model inference since it also allows for the transformation of predictions into reviewable, triageable, and monitorable signals.

5.4 Frontend Design and Development

The frontend part of the proposed system was designed as a web-based system meant to facilitate citizens' complaints and authority's reviews of them. Its architecture is based on the concept of role-based interfaces that offer different interface options depending on whether it is needed to file a complaint, monitor authorities' work, or inspect particular complaints. At the same time, the frontend was designed to have visual consistency between pages and present key information easily and clearly.

As was mentioned earlier, the frontend of the system has been designed using the approach based on a web framework that allows dividing web pages into components such as side menus, complaint cards, review panels, status badges, analytics, maps, among others. Such an approach makes it easier to make modifications by adding new components or entire pages.

Finally, one more goal of the frontend development has been transparency of its AI outputs. Rather than presenting one type of output in terms of a complaint classification, the interface was developed to show separate pieces of evidence, including text-model result, image-model result, fusion result, visual evidence panels, reliability, urgency information, duplicate context, and summary. In other words, the frontend acts both as a presentation and interpretability layer.

5.4.1 Welcome Page

As an entry page to the website, the welcome page had the function of explaining the purpose of the system and guiding users into either making a complaint or navigating the authority side, based on their purpose. The importance of the welcome page (figure 5.2) is that it combines information and navigational functions. It gives brief information about the system, its complaint management flow, and entry actions that users should take for reporting and review.

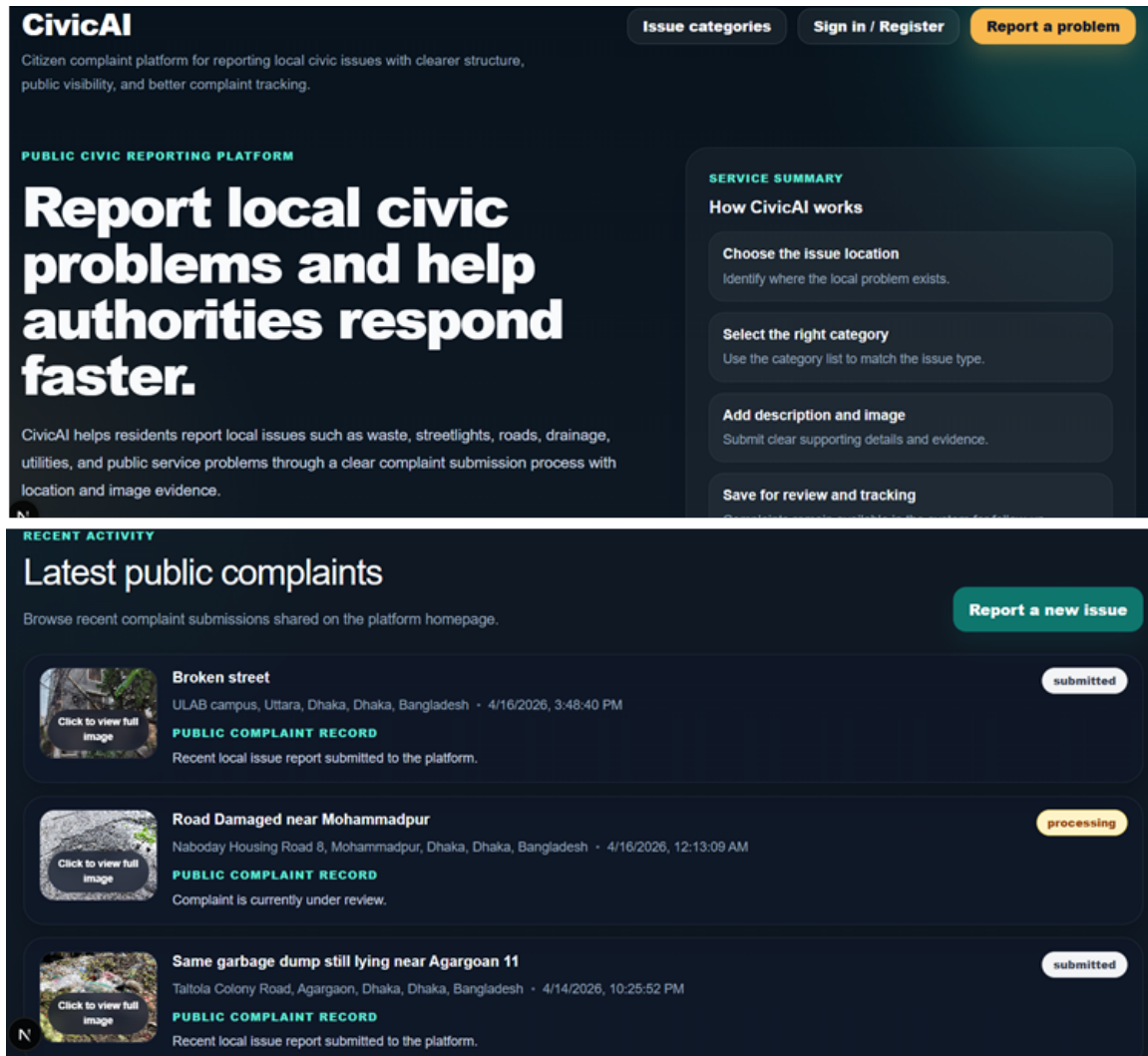


Figure 5.2: CivicAI Home Page

The welcome page is supposed to communicate to citizens and administrators about the purposes of the website. Instead of providing technical details, it introduces the website as a system for civic reporting.

5.4.2 Citizen Reporting Interface

The complaint reporting page (figure 5.3) was designed with the aim of structuring the process of submitting complaints through complaint titles, descriptions, categories and geographic locations.

CITIZEN WORKSPACE

Report page

Submit a clear complaint with location, description, category, and image so it can be reviewed faster and more accurately.

Back to homepage

Report complaint

VERIFIED COMPLAINT REPORTING

Report a civic complaint

Share the issue location, category, a clear description, and an optional image so the complaint can be reviewed faster and more accurately.

ACCESS

Verified user

You can submit a complaint because your account is verified.

CATEGORY

Citizen selected

Choose the closest complaint category so authorities can review it faster.

MAP FLOW

Select then mark

First choose the area, then click the exact problem point on the map.

CITIZEN WORKSPACE

Report page

Submit a clear complaint with location, description, category, and image so it can be reviewed faster and more accurately.

Back to homepage

Report complaint

Complaint submission form

Complete the required details below. Use the map to mark the exact problem location, not just your current location.

Citizen report

STEP 1

Reporter information

Reporter name

Enter your full name

Complaint title

For example, Broken street light near main road

Complaint category

Select category

Choose the category that best matches the issue. Authorities can correct it later if needed.

STEP 2

Location and map

Select the complaint area

Choose the division, district, and upazila or Dhaka city area first. The map will move automatically so the exact complaint point can be marked more easily.

Division

Select division

District

Select district

Upazila

Select upazila

Live location preview

After choosing the area, click the exact point on the map.

Use my current location

Mark another location on map

No location selected yet.

After selecting division and district, the map will move to that area. Then click the exact complaint location on the map.

Location details

Road name, village, market, nearby landmark, ward number, building name, or any details that help identify the place.

STEP 3

Complaint details

Description

Describe the issue clearly. Include what is happening, how long it has existed, who is affected, and any safety or access concerns.

Upload an image (optional)

Choose File No file chosen

Supported formats: JPG, PNG, WEBP. Maximum size: 8 MB

Submit complaint

Figure 5.3: CivicAI Citizen Reporting Interface

The complaint reporting page also provides an option for uploading pictures in order for the multimodal AI processing pipeline to analyze any visual evidence submitted by citizens.

5.4.3 Authority Dashboard

The authority dashboard (figure 5.4) was designed as an operational overview page rather than a simple list of complaints. The dashboard's purpose is to provide a quick means of inspection regarding the number of complaints received, the operational state of affairs, priorities, and trends identified via AI reviews.

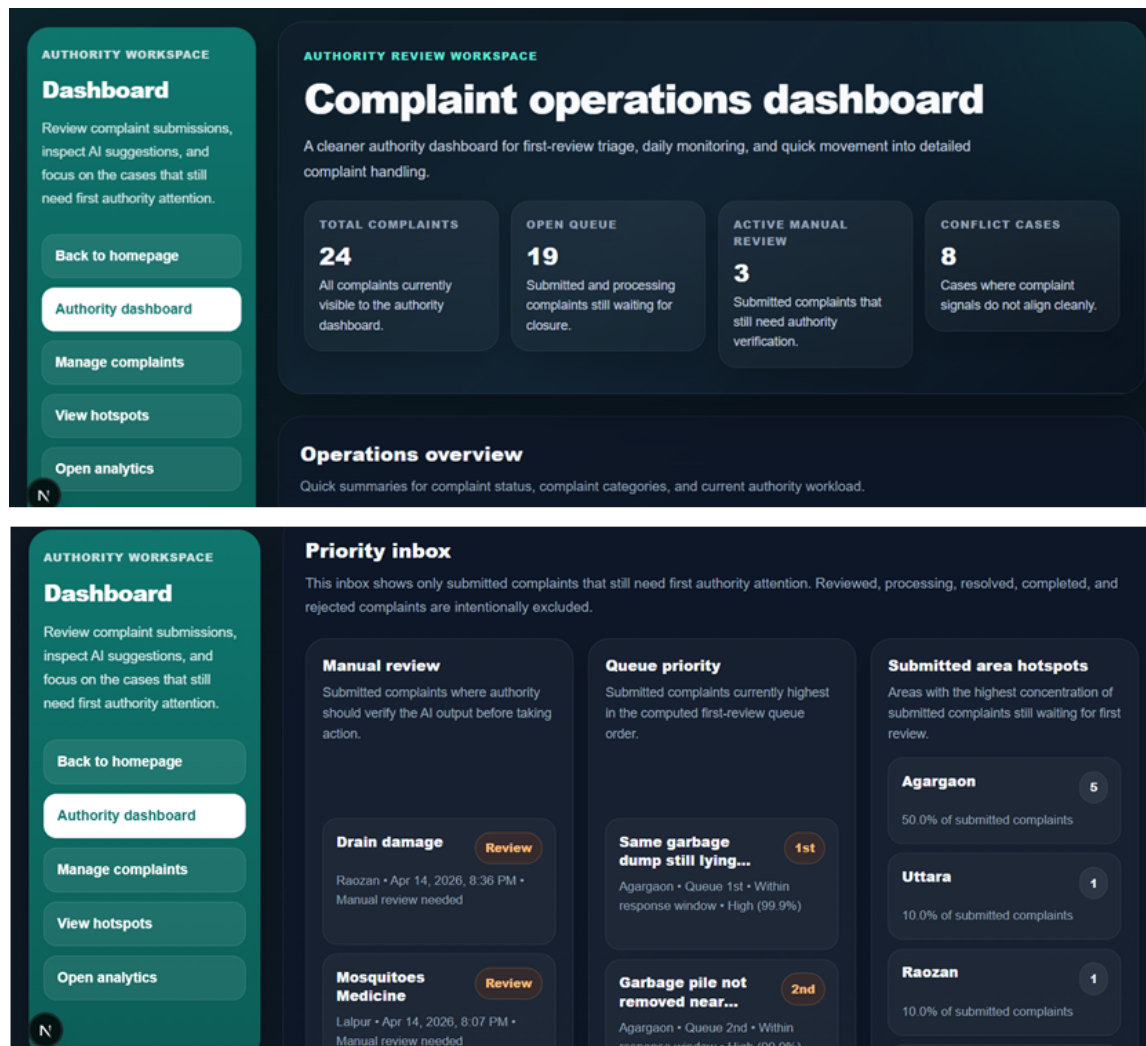


Figure 5.4: CivicAI Authority Dashbboard

The dashboard design provides a fast means of navigating to specific areas. Instead of having the authority look through all of the complaints in search of

certain trends, a summary indicator will lead the authority to areas within the complaint management application that provide additional context.

5.5 Backend Design and Development

The backend infrastructure was engineered to enable complaint persistence, authentication-savvy processes, AI inference, and authority-side complaint administration. This is because the backend plays the role of the intermediary layer between the web-based frontend, database, and multimodal AI models. In terms of architecture, it is imperative to acknowledge that the backend does not only entail model inference but also entails transformation of model outputs to actionable information usable by the frontend.

The backend infrastructure offers complaint submission, acquisition, review, status change, and AI processing services. During complaint analysis, the backend system acquires the appropriate model outputs, executes the multimodal fusion algorithm, generates additional review signals, and saves the inference outputs generated. Authorities are thus able to retrieve complaint data without having to redo all computations.

One of the crucial considerations made during the development of the backend system was prudent decision support. The model was never intended to fully substitute human review. Consequently, the backend infrastructure was designed such that model outputs would be easily reviewable and even overridden by authority members via the frontend interface.

5.5.1 API Layer

The role of the API layer was that it provided a framework through which communication could occur between the frontend and backend of the application. Specifically, this would involve submitting complaints, retrieving complaints, processing AI results on submitted complaints, updating complaints with reviews, and providing analytics-friendly data access. The separation of the frontend and backend components makes the architecture easy to manage.

It is particularly relevant in a multimodal complaint platform because there would be many different types of access requirements for complaint records depending on

where the component accesses these records. Complaints must be submitted by the frontend reporting module by inserting records, retrieved by the review module to provide detailed complaints and inferences, and provided in aggregate form for analytics.

5.5.2 AI Inference Pipeline

The pipeline for performing inferences through AI is the intelligent piece that operates in the backend. This comprises the text branch, the image branch, and the multimodal fusion phase. Complaint narratives undergo processing via the transformer classifier that has been fine-tuned, whereas image branches detect visual evidence in uploaded images via the detector based on YOLO training. The results are then integrated into a common civic interpretation space and fused into an end complaint understanding output.

This inference pipeline can additionally perform more authority-oriented functions beyond category determination. These are confidence reporting, urgency indicators, repetitive pattern recognition, duplicate checking, and geospatial analysis capabilities. The presence of these supplementary elements makes the backend operate as a decision-making tool and not just as a machine learning server endpoint.

5.5.3 Database and Storage Integration

A structured approach to storing complaints, their images, the users' information, and the AI's inferences has been adopted since all these elements play an important role in carrying out the project. The design of the database ensures that the functions that involve transactions as well as those related to retrieval for use by the AI are taken care of.

It becomes necessary to integrate storage in order to ensure that the images attached to complaints as well as their output results are kept accessible over time. For this reason, the database stores complaints and media files separately while ensuring that their connection remains clear.

5.6 Website Features

The website that is developed consists of various components designed to facilitate all phases in the management of complaints.

- Welcome page for introducing the platform and directing users to complaint submission or authority-side workflows.
- Citizen complaint reporting page with fields for complaint title, description, selected category, and location.
- Welcome page to introduce the system and guide the user to submit a complaint.
- Citizen complaint submission page containing the complaint title, description, category selected, and location.
- Image upload component for analyzing images using the AI.
- Use of maps to assist in selecting the location of complaints.
- Authority dashboard page that provides an overview of the complaint management process.
- Complaints review page that gives information regarding the particular complaint and the AI's analysis.
- Text model result, image model result, and the fusion result displayed separately for ease of comprehension.
- Decision summary panel displaying the current category, confidence, reliability, urgency, and queue position.
- Evidence panel featuring the uploaded complaint image and detection annotation.
- AI generated complaint description for fast authority analysis.
- View duplicates tool that helps in locating duplicate complaints about the issue.

- Repeated pattern context that helps in finding similar repeated complaints based on area-category context.
- Status updater tool along with final categorization tool for authorities.
- Resolution note section and review note for any further administration work.
- Supporting reporter notification for reporting updates to the complaint reporter.
- Complaint management queue page that is helpful in managing all complaints efficiently.
- Analytics interface that facilitates analyzing complaint categories, areas, statuses, and operations trends.
- Hotspots or heat map visualization for understanding complaint concentration through geographical visualization.

These features demonstrate a working prototype of an AI-assisted complaint-management environment.

5.7 Testing and Validation

Model and system tests have been conducted to ensure the quality of the created solutions. Model level test has covered the following aspects: text classifier validation, image detector assessment and the fusion workflow verification. The latter two tasks evaluated predictive performance, comparative results and the possibility for the system to generate meaningful outputs in terms of complaints management. At the system level, a full cycle of actions was tested, namely: complaint submitting, retrieving records of complaints, presenting outputs generated by artificial intelligence, conducting authority's reviewing process, duplicate-based review inspection, visualization of hotspots and complaints filtering using management interface. The aim of this test phase was to check how well various components of the platform function together as a single prototype.

5.8 Deployment and Hosting

CivicAI was first tested and validated in the local development environment where both the frontend and backend are separated into individual services but connected via the database, authentication, and storage layers. It is important to note that this setup is well-suited for the development process of implementing, testing, and validating the functionality of the entire solution.

During the final stage of this work, the system was deployed in the cloud environment as proof of concept demonstrating the practical usability of this project outside the local development environment. The front end is deployed to Vercel, while the backend API is running in Hugging Face Spaces. Supabase is used to provide the database, authentication, and media storage capabilities, and Brevo is employed for OTP-based email verifications and notifications for reporters. The design follows the modular nature of the architecture proposed by the system. Although each component is independent, it can interact with other components seamlessly. The system is thus easier to maintain and can be upgraded in the future without having to make fundamental changes to the entire structure.

Chapter 6

Result and Discussion

6.1 Text Model Comparison

Table 6.1, 6.2, 6.3 shows the results of the fold-by-fold cross-validation carried out on each text model candidate. Overall, the results show that the performance of RoBERTa generalizes better than DistilBERT and the SBERT model on the classification task when faced with class imbalance. Our work on the training and evaluation of the DistilBERT model used a cleaned and filtered text dataset containing 18 categories of complaints. A Stratified cross-validation on the 80% of the data used for training is employed. The number of samples used for training and validation during the folds, once the GPU environment is achieved, ranges between 7,563 to 7,564 and 945 to 946, respectively. Table 6.1 shows the K Fold Cross Validation results for DistilBERT model.

Table 6.1: DistilBERT 9-fold Stratified Cross-validation Results

Fold	Val Loss	Accuracy	F1 Macro	F1 Weighted	Balanced Acc.	Precision Macro	Recall Macro
1	0.5287	0.8552	0.4959	0.8413	0.5139	0.4832	0.5139
2	0.4694	0.8763	0.4757	0.8606	0.4752	0.5028	0.4752
3	0.5135	0.8636	0.4675	0.8462	0.4615	0.5043	0.4615
4	0.5877	0.8436	0.4396	0.8278	0.4280	0.4737	0.4280
5	0.5118	0.8646	0.5229	0.8491	0.4990	0.5769	0.4990
6	0.5162	0.8635	0.4999	0.8478	0.4860	0.5628	0.4860
7	0.4938	0.8730	0.5240	0.8588	0.5401	0.5562	0.5401
8	0.5302	0.8667	0.5011	0.8525	0.5119	0.5122	0.5119
9	0.5440	0.8550	0.4849	0.8392	0.4812	0.5048	0.4812

Over the nine folds, DistilBERT maintained a strong accuracy rate, usually

ranging from mid to high 0.86 levels, accompanied by a macro-F1 value of 0.84 to 0.86. This indicates that the performance of DistilBERT is good in most classes. The macro-F1 and balanced accuracy of DistilBERT stood at 0.49, showing the difficulty introduced by the long-tail distribution of categories for civic complaints, for which the less common classes are tougher to handle and hence have a negative effect on macro-level performance. Overall, DistilBERT sets up a good baseline performance that remains constant over the folds, and for macro-level performance, models having more capacity, like RoBERTa, might perform better.

We evaluated RoBERTa-base on filtered text data with 18 complaint categories using Stratified 9-fold cross-validation on 80% of the training data (with support for GPUs). In each fold, there were about 7,563 to 7,564 samples for training and 945 to 946 for validation, maintaining the proportion of classes in each fold. In comparison to DistilBERT, it was observed that there is a significant improvement in macro-level performance, which is highly desirable in class-imbalanced problems because it considers all classes equally. In all folds, RoBERTa maintained a macro-F1 of about 0.53 and a balanced accuracy of about 0.53, while also maintaining high accuracy and weighted-F1 of about 0.87 and 0.86, respectively.

This table 6.2 shows that the performance of RoBERTa on the major complaint types, evident from the weighted F-1 score, has been coupled with improvements in the minority class detection capabilities, and this is evident from the macro F-1 and balanced accuracy scores.

Table 6.2: RoBERTa 9-fold Stratified Cross-validation Results

Fold	Val Loss	Accuracy	F1 Macro	F1 Weighted	Balanced Acc.	Precision Macro	Recall Macro
1	0.4683	0.8763	0.5496	0.8669	0.5418	0.5880	0.5418
2	0.4297	0.8858	0.5217	0.8766	0.5129	0.5830	0.5129
3	0.5027	0.8742	0.5290	0.8625	0.5296	0.5798	0.5296
4	0.5499	0.8425	0.4986	0.8348	0.5024	0.5141	0.5024
5	0.4776	0.8646	0.5224	0.8560	0.5250	0.5428	0.5250
6	0.4773	0.8794	0.5578	0.8690	0.5451	0.6077	0.5451
7	0.4687	0.8815	0.5721	0.8734	0.5828	0.5946	0.5828
8	0.4848	0.8624	0.5254	0.8561	0.5301	0.5532	0.5301
9	0.5276	0.8603	0.5164	0.8505	0.5231	0.5279	0.5231

The execution time spent per fold was higher compared to the time taken for execution of the DistilBERT model, but this is expected considering the model size,

and the process has been stable and foolproof. The choice of using the model in a civic complaint setting for the final text classifier can be verified from the above points.

We evaluated an SBERT (Sentence-BERT) pipeline using Stratified k-fold Cross Validation for an 18-class complaint text classification task. In this approach, for each complaint, a fixed-size sentence embedding is produced using a Sentence-Transformer pre-trained model. Following this, a simple classifier is trained on the sentence embeddings to classify them. Training and evaluation were in a manner that took minimal computational power in terms of computing the embedding representations in the training and evaluation phases. Thus, SBERT was less computationally expensive compared to transformer training.

The metrics on fold-wise performance from table 6.3 suggest that SBERT’s performance was slightly lower than RoBERTa, especially on the minority classes.

Table 6.3: SBERT-based Classifier 9-fold Stratified Cross-validation Results

Fold	Accuracy	F1 Macro	F1 Weighted	Balanced Acc.	Precision Macro	Recall Macro
1	0.8552	0.5075	0.8403	0.5118	0.5068	0.5118
2	0.8562	0.4443	0.8382	0.4438	0.4781	0.4438
3	0.8467	0.4854	0.8328	0.4695	0.5252	0.4695
4	0.8182	0.4379	0.8034	0.4120	0.4942	0.4120
5	0.8455	0.4718	0.8299	0.4685	0.4823	0.4685
6	0.8508	0.4925	0.8347	0.4693	0.5463	0.4693
7	0.8614	0.5463	0.8514	0.5437	0.5761	0.5437
8	0.8402	0.4611	0.8240	0.4474	0.4817	0.4474
9	0.8529	0.4815	0.8374	0.4658	0.5205	0.4658

Though accuracy was relatively stable for each fold, macro-F1 measures were lower and had larger variability. This was expected because a long-tail representation could have been difficult to capture compared to fine-tuning RoBERTa. In conclusion, although SBERT was a strong semantic baseline, it was not as accurate as RoBERTa for capturing fine distinctions between complaint types.

6.2 Final Selected Text Model (RoBERTa)

RoBERTa is chosen based on crossvalidation and is re-trained on the full 80% training set by using an internal train/ validation split of approximately 90/10 and

employing early stopping. The behavior of models in terms of validation at each epoch is summarized in Table 6.4.

Table 6.4: RoBERTa Epoch-wise Validation Metrics (Internal Validation Split)

Epoch	Train Loss	Val Loss	Accuracy	F1 Macro	F1 Weighted	Balanced Acc.	Precision
1	0.8023	0.5640	0.8320	0.4315	0.8137	0.4348	0.4970
2	0.4383	0.4912	0.8613	0.5632	0.8550	0.5755	0.5910
3	0.3219	0.4899	0.8660	0.5961	0.8655	0.6285	0.5913
4	0.2487	0.4851	0.8813	0.6352	0.8761	0.6206	0.6871
5	0.1909	0.5210	0.8707	0.6234	0.8688	0.6289	0.6456

The final locked-test performance on the held-out 20% test set is reported in Table 6.5. The final results of the RoBERTa test were Macro-F1 = 0.6149, Weighted-F1 = 0.8604, and Accuracy = 0.8661. A long-tail distribution is projected in the difference between weighted-F1 and macro-F1: macro measures assess fairness across all categories, while performance on most frequent categories is high.

Table 6.5: Final RoBERTa Performance on the Locked 20% Test Set

Model	Accuracy	F1 (W)	F1 (M)	Balanced Acc.	Precision (M)	Recall (M)	Loss
RoBERTa-base	0.8661	0.8604	0.6149	0.6045	0.6731	0.6045	0.5619

6.3 Visual Model Results (YOLOv8n) – Validation and Test Performance

The YOLOv8n detector is trained for 10 epochs with input resolution 416. Table 6.6 summarizes training loss trends across epochs. Table 6.7 and Table 6.8 report per-class validation and test detection performance, respectively.

Table 6.6: YOLOv8n Training Losses Across Epochs

Epoch	GPU Mem (GB)	Box Loss	Cls Loss	DFL Loss
1	2.57	1.741	2.791	1.689
2	2.59	1.712	2.161	1.652
3	2.59	1.722	2.070	1.661
4	2.61	1.653	1.922	1.614
5	2.62	1.599	1.810	1.573
6	2.62	1.552	1.713	1.534
7	2.62	1.511	1.636	1.506
8	2.74	1.464	1.564	1.475
9	2.74	1.432	1.505	1.450
10	2.74	1.403	1.452	1.430

Table 6.6 Training loss for each epoch from 10 for YOLOv8n detector, input resolution is 416x416. From the table, one can see that box loss, classification loss, and dfl loss all tend to converge along with more epochs, thus showing that the training is stable. The GPU memory usage remains consistent, peaking to 2.74 GB by the end of training.

Table 6.7 summarizes the performance of the best.pt YOLOv8n model on the dataset at an input resolution of 416x416, listing Precision, Recall, mAP@0.5, and mAP@0.5:0.95 values for each class, including excellent performances on classes such as Broken Road Sign Issues and Damaged Electric wires and poles, with lower values on other classes such as Illegal Parking Issues and Damaged concrete structures.

Table 6.7: YOLOv8n Validation Performance (Best.pt, imgsz=416)

Class	Imgs	Instances	Box P	R	mAP50	mAP50-95
All	4894	15731	0.544	0.502	0.518	0.315
Damaged Road Issues	Is- 789	1405	0.487	0.383	0.386	0.189
Pothole Issues	570	1818	0.571	0.400	0.443	0.187
Illegal Parking Issues	Is- 24	38	0.230	0.026	0.050	0.014
Broken Road Sign Issues	Is- 41	41	0.727	0.878	0.882	0.603
Fallen Trees	982	1237	0.656	0.641	0.662	0.356
Littering/Garbage	660	1175	0.609	0.601	0.648	0.437
Vandalism Issues	208	393	0.529	0.664	0.659	0.473
Dead Animal Pollution	Is- 25	25	0.442	0.476	0.462	0.235
Damaged Concrete Structures	1225	9113	0.516	0.149	0.200	0.098
Damaged Electric Wires/Poles	409	486	0.669	0.800	0.791	0.558

This model possesses a fair level of precision and recall on most classes.

Table 6.8 shows the test performance of the YOLOv8n detector with an input resolution of 416x416. The table depicts for each class, the Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95. It is observed that the Broken Road Sign Issues and Damaged Electric wires and poles are doing well, whereas Illegal Parking Issues and Damaged concrete structures are not.

Table 6.8: YOLOv8n Test Performance (Imgsz=416)

Class	Imgs	Instances	Box P	R	mAP50	mAP50-95
All	4160	15081	0.619	0.498	0.536	0.341
Damaged Road Issues	Is- 61	83	0.494	0.575	0.484	0.333
Pothole Issues	352	1492	0.688	0.359	0.472	0.215
Illegal Parking Issues	Is- 18	38	0.364	0.151	0.123	0.066
Broken Road Signs	Is- 39	40	0.878	0.825	0.880	0.688
Fallen Trees	1632	2015	0.761	0.689	0.761	0.409
Littering/Garbage	331	1032	0.574	0.403	0.470	0.262
Vandalism Issues	203	433	0.566	0.665	0.673	0.458
Dead Animal Pollution	Is- 14	14	0.572	0.429	0.487	0.303
Damaged Concrete Structures	1220	9580	0.579	0.124	0.206	0.100
Damaged Electric Wires/Poles	278	354	0.718	0.763	0.803	0.572

In general, the model is doing reasonably well with a mean mAP@0.5 of 0.536 and mAP@0.5:0.95 of 0.341.

The model does an efficient job in detecting various issues in the civic area, including damage to electricity cables and poles, damaged road signs, fallen trees, vandalism, and potholes. The results provided by the model are given with bounding boxes and the classes along with their respective confidence score. The following samples (Fig 6.1) highlight the abilities of the model.



Figure 6.1: Model detection of urban infrastructure issues, including damaged electric wires, broken road signs, and potholes.

Such a model is capable of detecting different categories of complaints, like littering in public places and fallen trees. It does that through bounding boxes and labels illustrated in the pictures provided. Examples provided below (Fig 6.2) highlight the ability of the model to detect and localize multiple issues in cities.

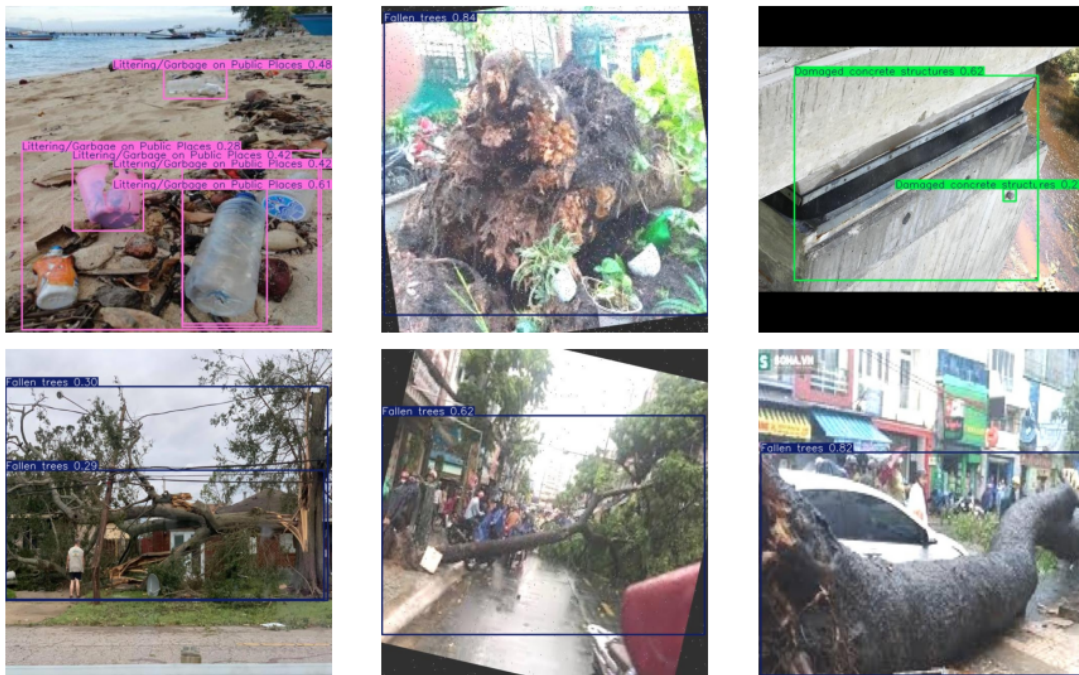


Figure 6.2: Model detection of urban issues, including littering and fallen trees in public areas.

This machine learning model has been able to detect problems associated with the potholes in the image of the road (fig 6.3) with varying degrees of confidence. In addition, it is able to detect faulty electricity wires and poles in the image of the building (fig 6.3), although with a lower degree of confidence. This shows that the model can generalize and recognize issues in new data, though its accuracy may vary.



Figure 6.3: Model recognition of Images outside of dataset/real time.

The curve 6.4 illustrates the dependency between the recall and confidence metrics for different complaint types.

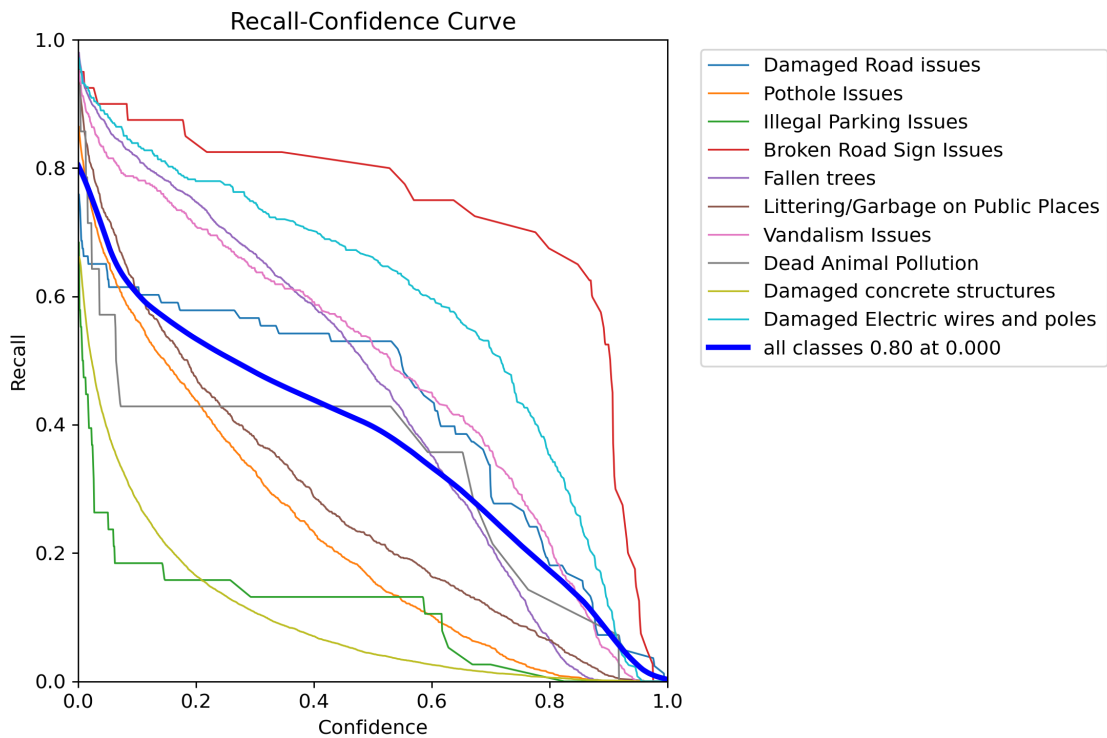


Figure 6.4: Recall–Confidence curve of the YOLOv8n detector

The "all classes" curve (blue) displays the average behavior for all complaint types. Some classes, such as "Damaged Road Problems" and "Pothole Problems," demonstrate high recall at low confidence values, while other classes, such as "Illegal Parking Problems," demonstrate a sharp decline in recall as the confidence value grows. The curve 6.4 illustrates the dependency between the recall and confidence metrics for different complaint types. The "all classes" curve (blue) displays the average behavior for all complaint types. Some classes, such as "Damaged Road Problems" and "Pothole Problems," demonstrate high recall at low confidence values, while other classes, such as "Illegal Parking Problems," demonstrate a sharp decline in recall as the confidence value grows.

The curve 6.5 represents the relationship between precision and recall for each individual class.

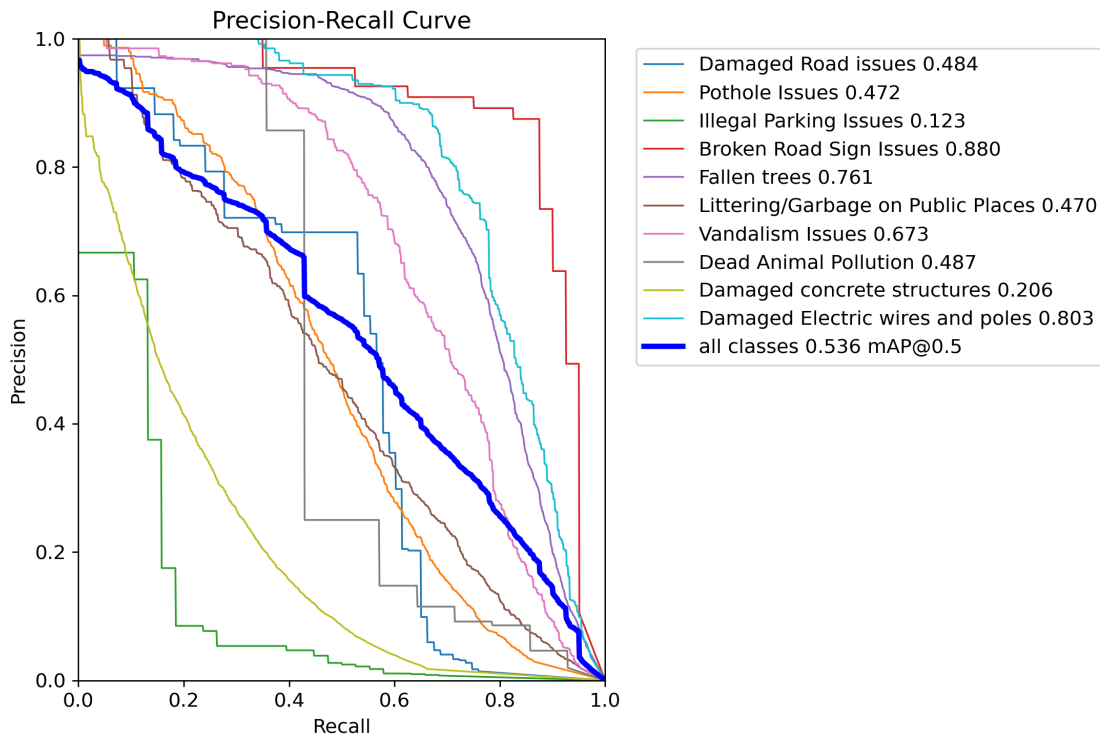


Figure 6.5: Precision–Recall curve of the YOLOv8n detector

The line "all classes" (blue) demonstrates the optimum balance of precision and recall, scoring an average macro-precision of 0.536 with a recall rate of 0.5.

"Broken Road Sign Problems" and "Damaged Electric Cables" are classes with high precision and low recall.

This 6.6 curve displays the relationship between precision and recall under various confidence values.

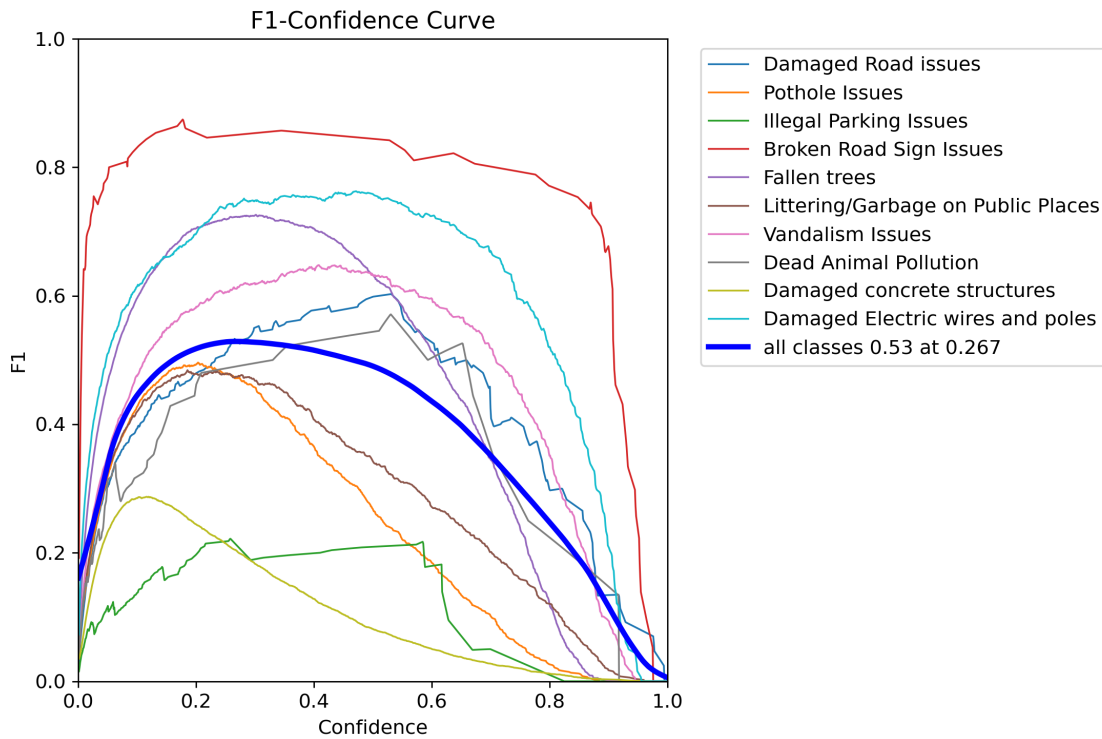


Figure 6.6: F1–Confidence curve of the YOLOv8n detector

The line "all classes" (blue) depicts the point at which the highest F1 score is attained at a confidence value of 0.27. Other categories like "Damaged Road Issues" and "Broken Road Sign Issues" attain high F1 scores under various confidence values.

In the matrix in Figure 6.7 it can be seen that damaged roads have a prediction accuracy of 60%, whereas potholes and illegal parking have low prediction accuracies.

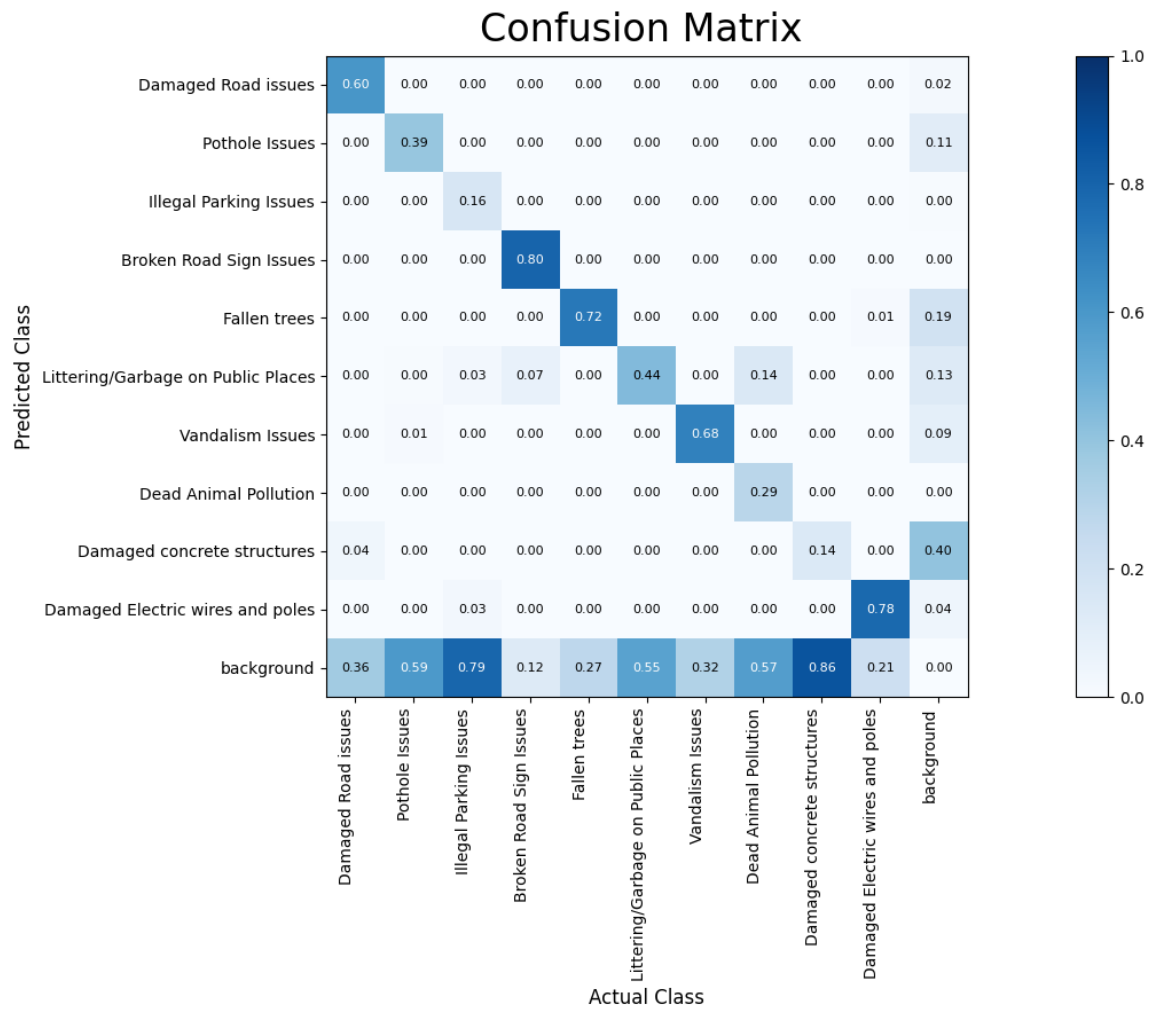


Figure 6.7: Confusion matrix of the YOLOv8n detector

In this case, the background class has attained a high normalization score, implying that the proportion of mistakes among this category is more. The Per-class mAP chart in Figure 6.8 illustrates how well the YOLOv8n model performs on the test set.

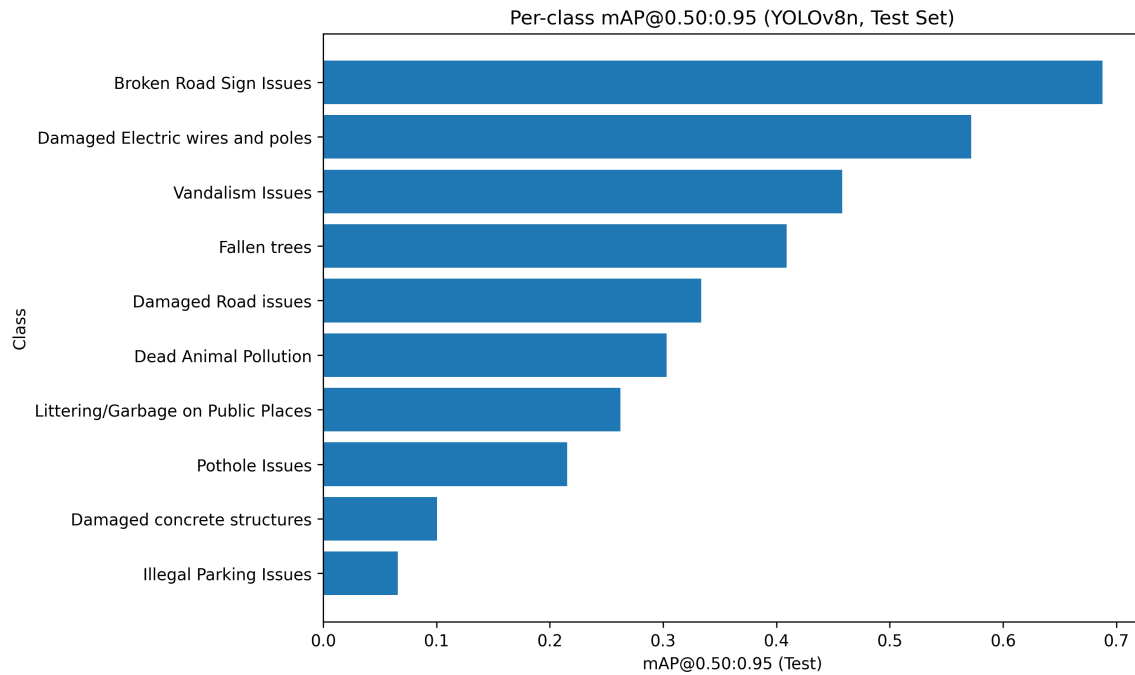


Figure 6.8: Per-class mAP50-95

The categories of Broken Road Sign Issues and Damaged Electric Wires and Poles achieve the highest mean average precision (mAP), whereas Illegal Parking Issues and Damaged Concrete Structures score the lowest. In general, the model shows the best performance for more frequent problems such as Fallen Trees and Vandalism Issues.

The figure 6.9 shows precision per class on the image detection model for each of the 10 civic issue types.

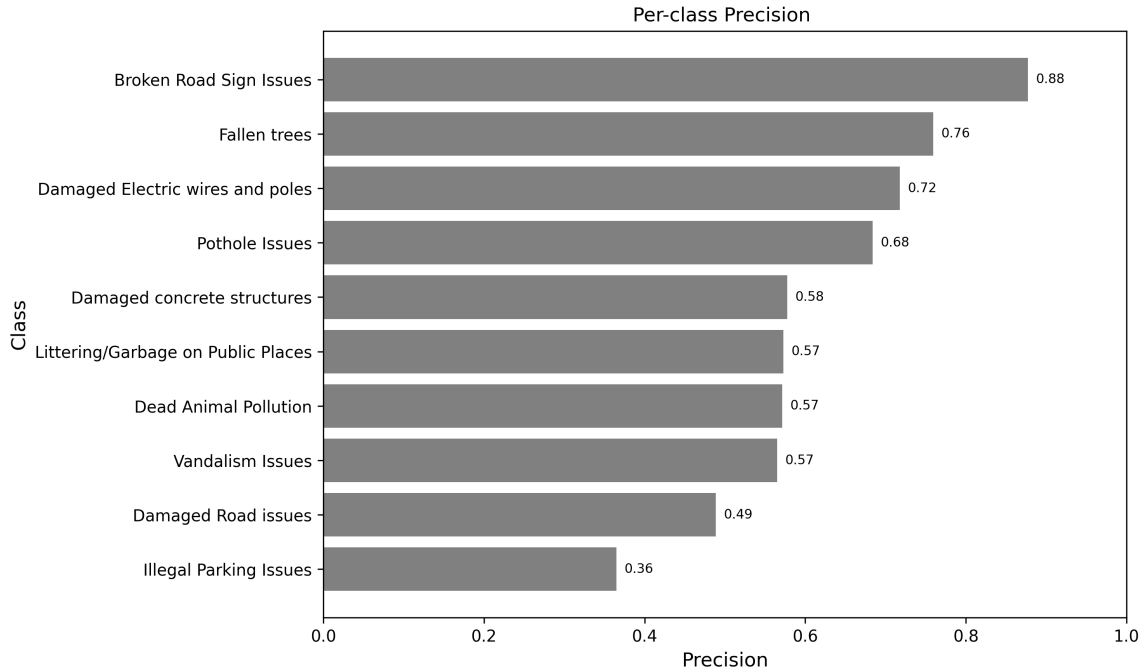


Figure 6.9: Per-class Precision

The model's strongest precision was for Broken Road Sign Issues (0.88) and Fallen trees (0.76), showing strong reliability for the model in these classes. Weaker precision was shown for Illegal Parking Issues (0.36) and Damaged Road Issues (0.49). The greater the visual distinction in a class, the better the model was able to perform. The opposite is true for visual classes that lacked distinction.

6.4 Late Fusion and Decision Support Pipeline

For the assessment of the proposed approach as a decision-support tool for authorities instead of just a classifier, the trained RoBERTa text classifier was combined with the YOLOv8 visual evidence detector, and an end-to-end evaluation was conducted. Such a stage was intended to find out whether it is possible to convert the results of the multimodal model into useful signs for governing authorities like complaint classification, confidence, modality agreement, priority, brief summaries for quick scanning, duplicate clustering, hotspot detection, and other results ready

for dashboarding. In this regard, the following discussion of the findings will touch upon both predictive and operational aspects of the system’s performance.

6.4.1 Multimodal fusion behavior and consistency checking

The process flow of the unified complaint review system reveals that multimodal fusion using CivicAI involves more than just a simple summing up of two results. In cases where both the complaint text and its visual evidence are present, the system harmonizes the result of the RoBERTa text classifier with the YOLOv8 visual evidence classifier within a civic classification space. This makes it possible for the text result, visual result, and fused result to be understood within the same decision-making context.

As shown in Figure 6.10, the interface for reviewing purposes makes it clear what the category path looks like.

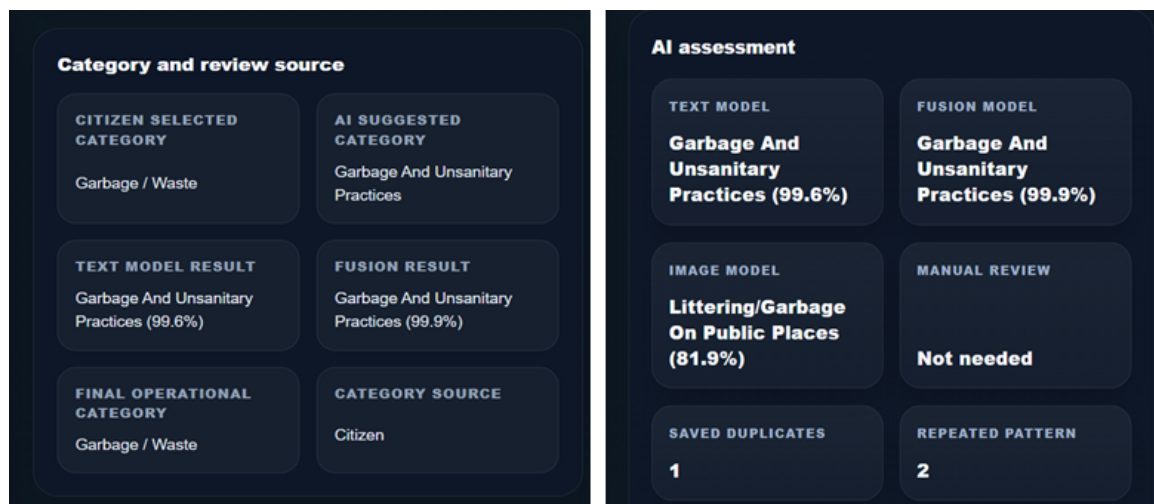


Figure 6.10: Category review-source and AI assessment panel showing results.

The user-chosen category, along with the suggested category by the AI, the text model output, the fusion output, and the operational category is all shown together. This enhances the ability to explain the results, since the reviewer is able to understand the process by which the system went from raw data to the category used in the operational procedure. In the case of the garbage example, the text model suggests Garbage and Unsanitary Practices with extremely high confidence,

and the fusion result matches it. The AI-based assessment technique depicted in Figure 1 has made it much clearer to see the contribution of each of the branches. The authorities can easily analyze how the contribution of each branch impacts the final decision because the interface segregates the output of the text model result, image model result, and the fusion result. As per the example mentioned, the text model outputs the prediction for "garbage" and "unhygienic behavior," whereas the image model detects "littering" and "garbage in public places." The fused result carries a high level of confidence score and even makes semantic sense to the text complaint.

Figure 6.11 demonstrates the importance of visual evidence within the multimodal process flow.

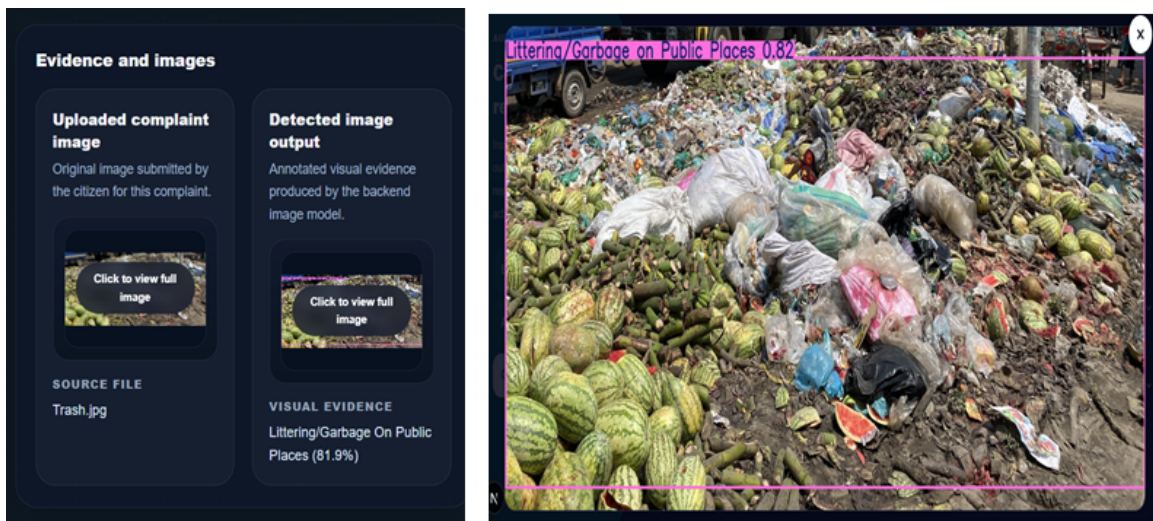


Figure 6.11: Evidence panel showing the uploaded complaint image and the annotated detector output together with the extracted visual-evidence label and confidence score.

The visual evidence that was uploaded and the results of object detection are presented within one reviewing interface, enabling the authority to contrast the presented scene with the interpreted scene generated by the algorithm. In effect, the visual pathway of this process becomes almost entirely auditable due to this interface. The reviewer will be able to see whether the interpretation of the objects and the visual label make sense in the context of the complaint rather than relying on logs from detectors.

Moreover, an intelligent fusion technique that would work with evidence provided by citizens is utilized within this process. Visual evidence must not overwhelm the process of evaluation because complaint pictures might be messy or have other aspects to them that can confuse a machine vision algorithm. By displaying the result in text, picture, and fusion form separately, it allows the authority to judge whether a manual inspection is required or whether the two forms complement each other.

6.4.2 Authority-Oriented Decision Outputs and Operational Support

An important outcome of the integrated pipeline is that unlike the simple classification process in a regular machine learning model, the pipeline generates structured information for decisions to be made by authorities rather than simply labeling the final complaint type. The structure in this case entails showing the authorities the working category of the complaint, alongside the confidence level of the model in its choice, whether the model is reliable, the urgency of the complaint, where it sits in the queue, and any other linked information.

Figure 6.12 demonstrates that it is not one opaque prediction that is revealed in the review interface.

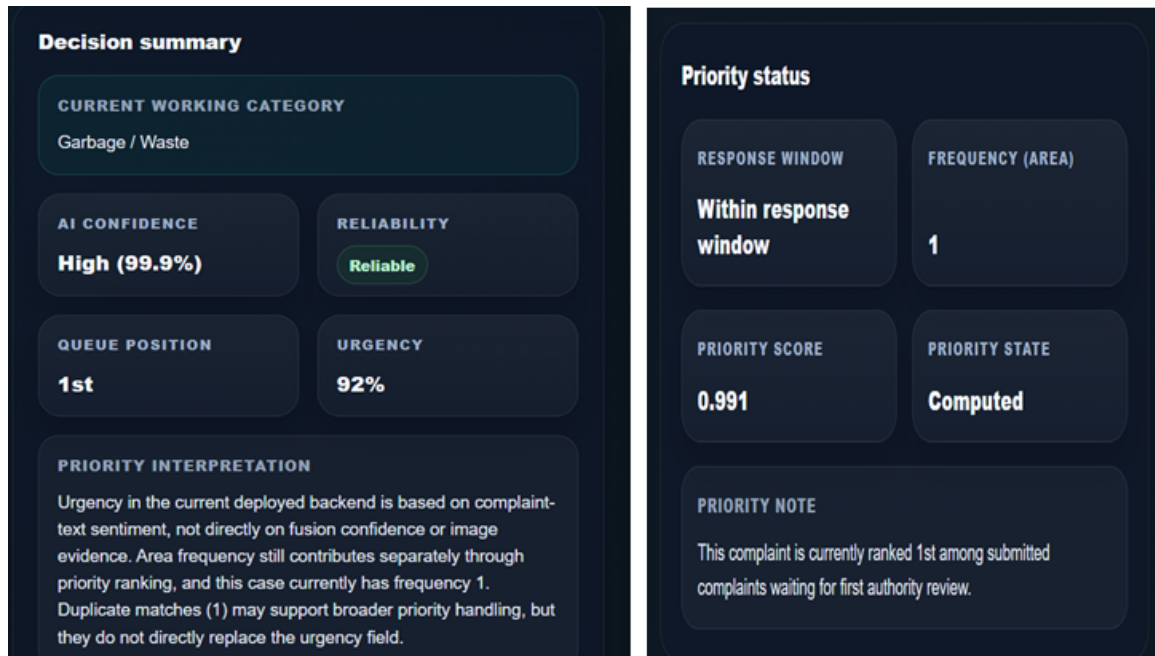


Figure 6.12: Complaint review interface showing the decision summary and priority status.

The review interface actually shows decision components which can be examined independently by the reviewing authority. The working category is shown along with the priority indicators and the actions menu, enabling the authority to understand the AI-suggested decision in its workable form. The importance of such an interface cannot be understated, especially in complaint management systems, as authorities have to know about not only the nature of the problem, but also its priority and the present possible steps towards its resolution.

The priority-indicator's output becomes crucial from the operations perspective. The screenshots show that no complaints are taken as equal even if they fall into the same larger category; rather, each complaint gets some triage-oriented indications like urgency percentage, queuing position, and a quick analysis of priority. This is important because authorities need to know not only what kind of problem has been reported, but also how soon it might need to be resolved and how it should be ranked in relation to other complaints that are currently in the review queue. In this way, the multimodal system functions not just as a classification model but also as an operational priority tool.

6.4.2.1 Complaint Summarization for Rapid Review

The system also produces a short summary in text form for authority review. The summary appears on the same complaint review screen as the complaint itself and aims to provide a compact representation of the complaint, predicted complaint type, and key supporting factors. This works well in the real world, where authorities will typically receive many complaints and have to review them one after another. In such cases, it is much easier to quickly scan an automatically generated summary rather than reread a lengthy complaint description every time. From a UI design standpoint, summarization also helps make the complaint review process easier for authorities working from a dashboard that displays complaints in card form or as lists. Summarization is better suited for quick browsing in these conditions than reviewing full complaint descriptions, especially in situations where authorities have to switch rapidly between individual complaints.

6.4.2.2 Duplicate Detection and Workload Reduction

Duplicate-awareness is yet another key decision-making aid that this prototype provides. The system does not consider each repetition as something new or distinct. Rather, it highlights duplicated reports along with repeated area and category combinations in order for the authorities to make a judgment on whether a particular report falls within the context of a bigger problem.

As shown in figure 6.13, this information is useful since it enables us to look at recurring complaints as an integral component of a bigger pattern of issues rather than as individual reports alone. Recurring or duplicated complaints might imply that there is a larger effect on the general population, that the issue is persistent, or that there is a necessity for joint action.

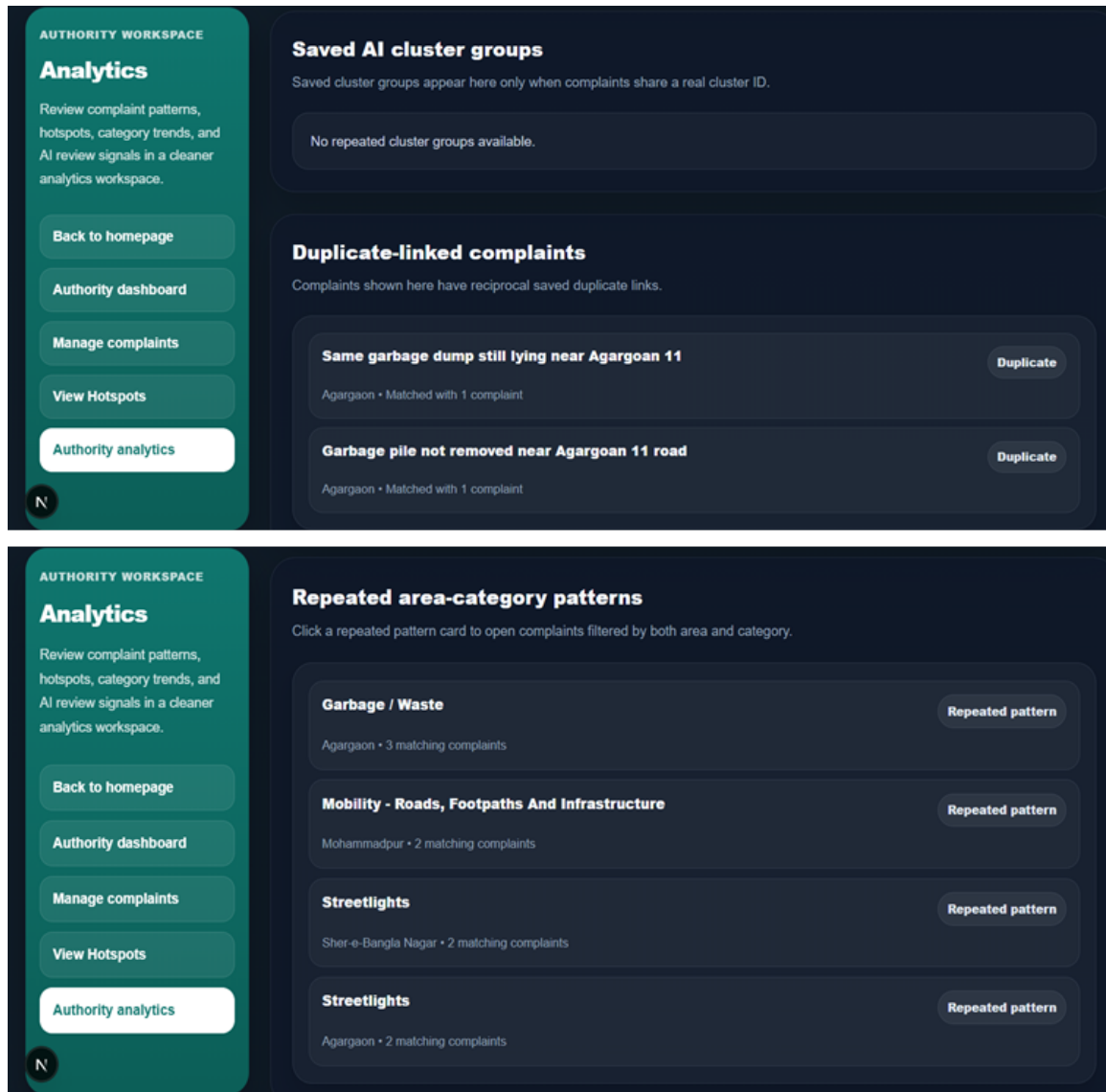


Figure 6.13: Analytics view showing repeated area-category patterns together with duplicate-linked complaints.

6.4.2.3 Hotspot Detection and Geospatial Decision Support

The prototype additionally enhances decision-making support not only at the complaint-by-complaint level but also at the geographic level. Rather than just saving complaints based on their location in raw coordinate form, the system converts those locations into geographic cues indicating where the concentration of

complaints is.

The hotspot interface for geospatial analysis is depicted in Figure 6.14. The heatmap visualization, pop-up points clustered together, and labeling of the areas ensure an easily comprehensible depiction of the complaints cluster.

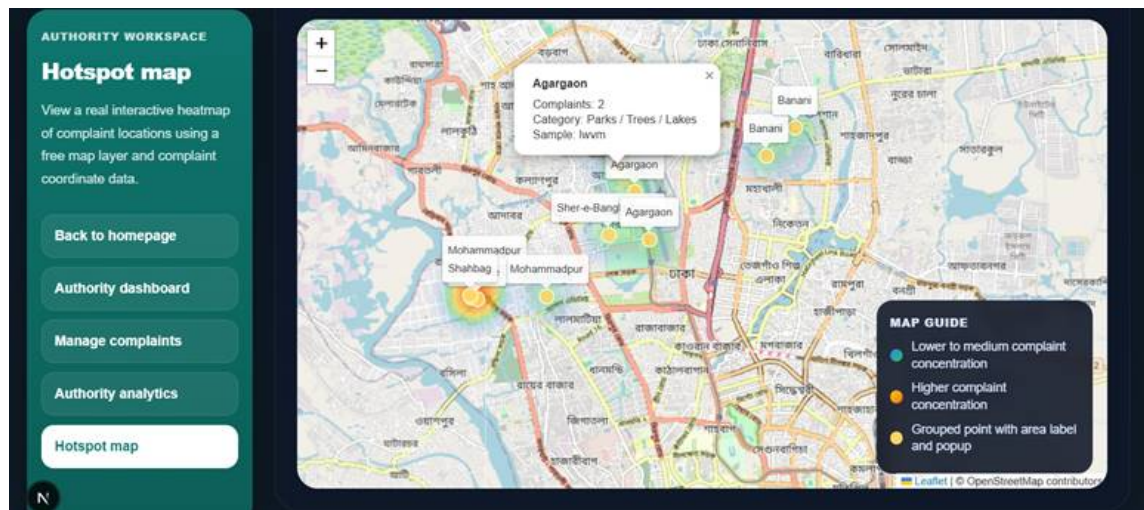


Figure 6.14: Hotspot map showing complaint concentration through a heatmap overlay, grouped area markers, and interactive pop-up labels for interpretation.

Such a spatial depiction proves to be practically useful since, in many cases, decision makers may not only be interested in knowing which particular complaint holds importance but also where the problem clusters lie.

6.4.3 End-to-end Readiness and Dashboard Integration

Another benefit provided by the developed solution lies in the usability of the outputs generated. Indeed, the pages that have been uploaded demonstrate an end-to-end process in which users could find complaint review page, hotspot map, operations dashboard, complaints management queue, as well as an analytics interface. Thus, the system generates the same outputs but in different forms: complaint review, priority inbox management, category filters, area analysis, repeated pattern inspection, and duplicate-linked complaint review.

The images uploaded in relation to the dashboard and analytics show that the system outputs cannot be regarded merely as a record of the models' activity. On

the contrary, it seems like those outputs have been included into an operational monitoring environment in which complaint trends, complaint status distribution, categories of complaints, areas involved, easy drill-down possibilities, and repeated issues can be analyzed on the go. Thus, the value of the pipeline increases significantly because it becomes possible to use the model output for micro and macro-level monitoring of complaints.

The results obtained indicate that the proposed solution could prove to be effective for multimodal complaint understanding and subsequent actions related to it. This increases the pipeline's relevance to real-world complaint-management environments, where operational usability, interpretability, prioritization, and repeat awareness are just as crucial as actual predictive performance.

Overall, the results suggest that the proposed system is not only capable of multimodal complaint understanding, but also capable of converting those model outputs into review-ready, triage-ready, and dashboard-ready authority support. This makes the implemented pipeline more relevant to realistic complaint-management settings, where interpretability, prioritization, repetition awareness, and operational usability are as important as predictive performance itself.

6.5 Strengths, Limitations and Implications

Overall, these findings suggest that the framework is an authority-centric decision support system that goes beyond merely classifying the input complaint. It should be noted that its primary advantage is associated with incorporating multimodal and operational outputs into a meaningful workflow. Not only does the framework classify the complaint, but it also provides the user with multiple outputs from text and image models, along with hints on how reliable each output is, urgency estimation, queue position, duplicates, repeated patterns, hotspots, and authorities' actions. The presented approach is more pertinent in the context of complaint management when an authority must analyze the outputs of models and make decisions based on them. Finally, the review page, analytics interface, hotspot view, and complaints queue show that model outputs can be turned into something actionable.

Another advantage of the approach taken here is that of interpretability. When the semantic content of textual and visual evidence matches each other, coherent

predictions for the complaint are produced at the fusion stage with high confidence levels. In this case, the decision can be more easily explained. Importantly, the interface makes it possible to see how much each individual modality contributes to the prediction, rather than just providing one result under an opaque label. It is necessary in this way, since the person reviewing the case will be able to check the text model result, the visual evidence result, the fused decision, and the authority category being applied.

Nevertheless, there are several remaining constraints. Firstly, the visual stream of information is intrinsically less wide than that of text-based one due to the limited nature of visual information that is only applicable to some common complaint types and may vary greatly in quality depending on the user's picture upload. In the case of real-life applications, the visual information may be compromised by such factors as image noise, low cues, occlusions, bad lighting conditions, or unclear images despite clear textual data. Secondly, not all complaints feature any valid image-based information, which limits the application of the method to text-based inferences. While this approach works well in practice, it also precludes the evaluation of the full potential of multimodal inference just yet. Finally, the analysis of duplicates, repeated events, and hotspots has obvious practical value, but its success is contingent upon the scope of tested complaints.

In light of these shortcomings, there are some clear implications for improvements moving forward. Performance could potentially be improved by incorporating more nuanced multimodal data, particularly with regard to visual categories that are currently less represented, as well as complaints that feature highly complementary text-image pairs. Additional collection of complaint data in the real world would enable more detailed investigation into the relative benefits of multimodal fusion versus text-based inference alone, more comprehensive examination of agreements and disagreements between the models, and greater stability in evaluating duplicate, repetitive pattern, and hot spot detection. Nonetheless, despite the presence of these limitations, it appears from the current findings that the project has transcended its early focus on experimental model development and is well underway in creating a functional AI-assisted complaint management system prototype.

6.6 Social, Legal and Ethical Issues

Use of AI in public complaint management, especially when using multimodal data in the form of texts, images, and geolocation, raises several ethical, legal, and social issues that have to be taken into account. Here are some of the most important questions to consider in relation to the implementation of this technology:

- **Privacy Concerns:** The processing of personal data and citizens' complaints poses serious risks for violation of people's privacy. It is crucial to keep personal data anonymous to avoid any problems related to data security and protection from possible misuse.
- **Bias and Fairness:** Text and image recognition systems can be inherently biased against certain demographics if not trained on an adequately diverse set of images or complaints. This may pose a problem in terms of proper classification and analysis of citizen complaints.
- **Accountability and Transparency:** It is imperative that AI systems be transparent and easily explainable, especially concerning decision-making procedures, like complaint prioritization or severity determination. Guidelines need to be set on how the process works to build trust between people and the authorities in regard to the use of the system.
- **Legal Implications:** The proposed solution must be legal and, more specifically, comply with various legislation, like the GDPR and other related regulations. In particular, the process must meet the data-protection legislation in regard to collecting, storing, and processing sensitive data, like images and location data.
- **Dependence on Technology:** There may be issues related to overdependence on automated systems, as sometimes it is important to take into consideration human input when evaluating complaints and interpreting their nature, as sometimes the machine may not capture all aspects of the issue.

Chapter 7

Conclusion

The project introduced CivicAI, a multi-modal AI-based system for public complaint understanding and authority decision support. The system combines complaint text, uploaded images, and geolocation data to overcome the limits of manual and single-modality complaint handling. RoBERTa achieved best text classification performance with 86.61% accuracy, 0.8604 weighted-F1 and 0.6149 macro-F1 and YOLOv8n visual detector achieved 0.619 precision, 0.498 recall and 0.536 mAP@0.5. The text model performs well for common complaint categories, while the visual model offers useful evidence for image supported civic issues, although the Class imbalance and visually similar categories affect the performance. The key advantage of the system developed within this project is its combination of text classification, visual detection, shared space fusion, geolocation detection, duplicates finding, hot spots detection, priority score generation, reliability validation, and presentation using a dashboard. In general, CivicAI shows how multimodal AI can help achieve faster complaint processing, smarter prioritization, lower duplicate load, and a more informed response to public services issues. Further development might focus on collecting additional real-life data, class balancing, fusing weights tuning, and actual users of authorities' evaluation.

References

- Arjunan, G. (2024). Ai beyond text: Integrating vision, audio, and language for multimodal learning. *International journal of innovative science and research technology*, 9(11), 1911–1917. Retrieved from <https://doi.org/10.5281/zenodo.14287143>
- Dattasmita, H., Supriya, M., et al. (2025). Civicxai-net: A lightweight multi-output distilbert framework for explainable civic sentiment and sarcasm detection. *International Journal of Advanced Research and Interdisciplinary Scientific Endeavours*, 3(2), 842–850. Retrieved from <https://doi.org/10.61359/11.2206-2542>
- Devanathan, R., Singh, A., Poornash, A., & Saha, S. (2024). Seeing beyond words: Multimodal aspect-level complaint detection in ecommerce videos. In *Proceedings of the 32nd acm international conference on multimedia* (pp. 243–252). Retrieved from <https://doi.org/10.1145/3664647.3681595>
- Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using yolo: challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6), 9243–9275. Retrieved from <https://doi.org/10.1007/s11042-022-13644-y>
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, pp. 226–231). Retrieved from <https://cdn.aaai.org/KDD/1996/KDD96-037.pdf>
- Gusdevi, H., Setyanto, A., Kusrini, K., & Utami, E. (2025). Cosine similarity-based evidences selection for fact verification using sbert on the fever dataset. *CogITo Smart Journal*, 11(1), 52–66. Retrieved from <https://doi.org/10.31154/cogito.v11i1.917.52-66>
- Hutto, C., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international aaai conference on web and social media* (Vol. 8, pp. 216–225). Retrieved from <https://doi.org/10.1609/icwsm.v8i1.14550>

- Jiao, Y., Li, C., & Lin, Y. (2021). Can urban environmental problems be accurately identified? a complaint text mining method. *Applied Sciences*, 11(9), 4087. Retrieved from <https://www.mdpi.com/2076-3417/11/9/4087>
- Karren, K., Schmitz, M., & Schaffer, S. (2024). Improving conversational user interfaces for citizen complaint management through enhanced contextual feedback. In *Proceedings of the 6th acm conference on conversational user interfaces* (pp. 1–11). Retrieved from <https://doi.org/10.1145/3640794.3665562>
- Kim, B., Yoo, M., Park, K. C., Lee, K. R., & Kim, J. H. (2021). A value of civic voices for smart city: A big data analysis of civic queries posed by seoul citizens. *Cities*, 108, 102941. Retrieved from <https://doi.org/10.1016/j.cities.2020.102941>
- Kumar, S., Atreja, S., Singh, A., & Jain, M. (2019). Adversarial adaptation of scene graph models for understanding civic issues. In *The world wide web conference* (pp. 2943–2949). Retrieved from <https://doi.org/10.1145/3308558.3313681>
- Liu, X., Zhang, J., Zhou, S., van der Plas, T. L., Vijayaraghavan, A., Grishina, A., ... others (2025). Towards deployment-centric multimodal ai beyond vision and language. *Nature Machine Intelligence*, 1–13. Retrieved from <https://doi.org/10.1038/s42256-025-01116-5>
- Madyatmadja, E. D., Pristinella, D., Dewa, M. D. K., Nindito, H., & Wijaya, C. (2020). Data mining techniques of complaint reports for e-government: a systematic literature review. In *2020 international conference on information management and technology (icimtech)* (pp. 841–846). Retrieved from [10.1109/ICIMTech50083.2020.9211277](https://doi.org/10.1109/ICIMTech50083.2020.9211277)
- Madyatmadja, E. D., Sianipar, C. P., Wijaya, C., & Sembiring, D. J. (2023). Classifying crowdsourced citizen complaints through data mining: Accuracy testing of k-nearest neighbors, random forest, support vector machine, and adaboost. In *Informatics* (Vol. 10, p. 84). Retrieved from <https://doi.org/10.3390/informatics100400894>
- Mamta, Singh, G. V., Kori, D. R., & Ekbal, A. (2024). Aspect-based multimodal mining: Unveiling sentiments, complaints, and beyond in user-generated content. In *Proceedings of the 32nd acm international conference on multimedia* (pp. 6433–6442). Retrieved from <https://doi.org/10.1145/3664647.3681703>
- Meel, P., & Vishwakarma, D. K. (2021). Multi-modal fusion using fine-tuned self-attention and transfer learning for veracity analysis of web information. *arXiv preprint arXiv:2109.12547*.
- Moya, M. G., Phan, T. T., & Gatica-Perez, D. (2021, October). Zurich like new: Analyzing open urban multimodal data. In *Proceedings of the 1st international workshop on multimedia computing for urban data* (pp. 1–8).

- Munot, D., Wakharde, D., Vispute, S., Kamble, V., Potdar, C., & Rajeswari, K. (2022). An automated system for rural people complaint handling, monitoring and analysis using data analytics. In *2022 6th international conference on computing, communication, control and automation (iccubea)* (pp. 1–7). Retrieved from <https://doi.org/10.1109/ICCUBEA54992.2022.10010933>
- Najeeb, R. F., Dhannoon, B. N., & Alkhalidi, F. Q. (2025). Topic modeling based bert & sbert transformer pretrained language modeling: A survey (2019-2023). In *Aip conference proceedings* (Vol. 3169, p. 030020). Retrieved from <https://doi.org/10.1063/5.0254615>
- Ortakci, Y., & Borhan, B. (2025). Optimizing sbert for long text clustering: two novel approaches with empirical insights: Y. ortakci, b. borhan. *The Journal of Supercomputing*, 81(8), 950. Retrieved from <https://doi.org/10.1007/s11227-025-07414-4>
- Panjaitan, M. I. (2019). Simple additive weighting (saw) method in determining beneficiaries of foundation benefits. *Login*, 13(1), 19–25. Retrieved from <http://login.seaninstitute.org/index.php/Login>
- Peng, X., Li, Y., Si, Y., Xu, L., Liu, X., Li, D., & Liu, Y. (2022). A social sensing approach for everyday urban problem-handling with the 12345-complaint hotline data. *Computers, Environment and Urban Systems*, 94, 101790. Retrieved from <https://doi.org/10.1016/j.compenvurbsys.2022.101790>
- Putra, R. R., Putri, N. A., & Putra, A. D. (2024). Implementation of cosine similarity and tf-idf techniques for web-based community complaint analysis in pertumbukan village, wampu district. *Instal: Jurnal Komputer*, 16(04), 10–19. Retrieved from <https://journalinstal.cattleyadf.org/index.php/Instal/index>
- Rajkumar, S. C., Yuvasini, D., Selvarajan, S., et al. (2026). A zero-shot LLM framework for multimodal grievance classification, urgency scoring, and abuse detection in civic feedback systems. *Scientific Reports*, 16, 2299. doi: [doi:10.1038/s41598-025-32079-7](https://doi.org/10.1038/s41598-025-32079-7)
- Ramalho, R. E. C., Firmino, A. A., de Souza Baptista, C., Falcao, A. G. R., de Oliveira, M. G., & de Andrade, F. G. (2020). Using natural language processing for extracting geospatial urban issues complaints from tv news.
- Reddy, A. S. (2024). Multimodal gen ai: integrating text, image, and video analysis for comprehensive claims assessment.
- Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

- Szeghalmy, S., & Fazekas, A. (2023). A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning. *Sensors*, 23(4), 2333. Retrieved from <https://doi.org/10.3390/s23042333>
- Vairetti, C., Aránguiz, I., Maldonado, S., Karmy, J. P., & Leal, A. (2024). Analytics-driven complaint prioritisation via deep learning and multicriteria decision-making. *European Journal of Operational Research*, 312(3), 1108–1118. Retrieved from <https://doi.org/10.1016/j.ejor.2023.08.027>
- Vrabie, C. (2025). Improving municipal responsiveness through ai-powered image analysis in e-government. *Viešoji politika ir administravimas*, 24(1), 9–23. Retrieved from <https://arxiv.org/pdf/2504.08972>
- Widodo, S., Brawijaya, H., & Samudi, S. (2022). Stratified k-fold cross validation optimization on machine learning for prediction. *Sinkron: jurnal dan penelitian teknik informatika*, 6(4), 2407–2414. Retrieved from <https://doi.org/10.33395/sinkron.v7i4.11792>