Deep Learning-Based Complaining Task Distribution towards Smart City

Lamisa Bintee Mizan Deya*, Abdul Aziz†, Farhatun Shama‡
*Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh
Email: lamisa.deya2001@gmail.com
†Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh
Email: abdulaziz@cse.kuet.ac.bd
‡Department of Computer Science and Engineering, Khulna University of Engineering & Technology, Khulna, Bangladesh
Email: farhatunshama@gmail.com

***Abstract*—The absence of an automated online platform for citizens to report complaints, along with the challenges faced by city corporations in managing them, poses significant hurdles. Additionally, there is a lack of resources for higher authorities to monitor the city corporations' progress effectively. The main objective of this research is to develop two Android applications aimed at automatically categorizing complaints. This was achieved using a deep learning model developed on the Teachable Machine platform, trained on a dataset comprising 5494 images of municipal issues. This study considered four categories of complaints, with each class comprising approximately 1500 images. Furthermore, measures were implemented to filter out fake complaints and ensure the accurate classification of images that were initially misclassified. The user application facilitates convenient reporting of complaints by capturing images of municipal issues, while the authority application delivers categorized complaints along with location and status updates. Higher authorities can oversee the progress of municipal operations, leading to increased transparency, efficiency and fostering nationwide smart city development.**

***Index Terms*—Smart City, Image Classification, Deep learning, Teachable Machine, Mobile Application**

I. Introduction

The lack of a streamlined online platform poses challenges for citizens in reporting issues and tracking the progress of complaints. City corporations face difficulties in categorizing and handling reported problems, leading to delays. Furthermore, higher authorities lack effective monitoring mechanisms for overseeing city corporations' management of reported issues. As a result, many problems go unresolved, while others are addressed slowly. This underscores the crucial necessity for improved systems and technologies in civic governance.

The paper aims to create a user-friendly Android application that enables citizens to report various issues. Leveraging an embedded deep learning model, the application will accurately categorize reported problems, offering city authorities a systematic approach to address each concern. Furthermore, the application will feature several graphical representations, providing insights into issue distribution across categories, status updates on reported problems, and a breakdown of issues by originating cities. This will allow higher authorities to effectively monitor and assess the city corporation's tasks. Additionally, the application will prioritize user privacy, ensuring the confidentiality and security of private data. Moreover, to promote inclusivity and accessibility, a dual-language feature has been incorporated, catering to diverse linguistic preferences and facilitating seamless interaction with the platform for individuals from various backgrounds. Users will also have access to feedback from the city corporation within the application.

The integration of Google's fused location services offers dual functionality, allowing users to automatically retrieve their location or manually select it. This functionality boosts user convenience by removing the necessity for manual input during complaint reporting. Users can also report issues offline or provide location details later. Precise location information is provided to authorities, who can easily visualize it on Google Maps. Additionally, authorities can communicate directly with users via email to gather more information about reported problems. They also can correct misclassified complaints and categorize fraudulent reports separately. These features streamline the reporting process and improve the accuracy and efficiency of complaint management in civic administration.

The major contribution of this research paper includes the following:

i Two mobile applications have been developed for the purpose of automating the complaining task distribution incorporating deep learning-based models.
ii A benchmark image dataset has been prepared that contains four classes for the training of the models.
iii Fused location services are integrated with the application that automatically fetches the location of the user.
iv Redirection to email and map gives authorities precise location information and enables direct communication with users for further details.
v Several graphical representations have been integrated to provide an overview of the progress of complaints.

II. Related Works

Numerous applications have been created with a similar aim, including the Complaint Management System [1]. In this current web application, citizens are required to manually choose the category and location of the issue to file a complaint or inform the authorities about problems in their vicinity. However, these processes lack automation and are tailored for specific municipal entities. Issues, including those outside city boundaries, are visible to authorities, complicating data filtering. Text-based complaints also pose challenges in verifying authenticity, while a centralized monitoring system for city corporation employees is absent.

The authors have published another research [2] where we have developed two mobile applications for distributing the complaining task. In that research, we just published the software part. In this research, we are publishing the rest of the part including the detailed result analysis.

The proposed study in [3] contributes to the existing literature. It delves into image classification and recognition technology, covering convolution operation, image feature extraction, and storage formats. Additionally, it explores the application of deep convolutional neural networks for image feature extraction and validates the model's performance with a dataset of 3000 images, demonstrating its effectiveness.

Google's Teachable Machine employs transfer learning and can achieve an accuracy rate of up to 100% [4]. In research [5], transfer learning was used to fine-tune the parameters of the pre-trained VGG19 network for image classification. Then its performance was compared with AlexNet, VGG16, and a hybrid learning approach involving convolutional neural network (CNN) architectures. Performance evaluation revealed that the fine-tuned VGG19 architecture outperforms other CNN architectures and the hybrid learning approach in the task of image classification. Again, the Teachable Machine employs MobileNets, which uses depthwise separable convolutions instead of standard CNN. Depthwise separable convolutions are used in the Xception architecture, which outperforms traditional CNN in many image classification tasks [6]. They are also used in ShuffleNet which shows outstanding accuracy compared to CNN across mobile devices [7]. So this overall structure with transfer learning and depthwise separable convolutions helps teachable machine achieve their state-of-the-art performance.

The study in [8] focuses on the development of an image recognition system for automatically classifying urban infrastructure issues. By using CNNs, the system achieves high accuracy in categorizing various types of municipal problems, thus streamlining the complaint resolution process.

The Municipal Complaints Unit application [9] empowers citizens to report issues by manually specifying categories and locations. Once submitted, authorities can review and update the status of complaints. However, the app's limitations include its restriction to a single city, which may result in irrelevant data and user errors due to manual category selection.

III. Methodology

A. System Overview

The overall system depicted in Fig. 1 has four primary components: the user application, the authority application, the database, and internal computation. The user application serves as the interface for citizens to interact with the system, while the authority application is designed for city corporation officials and higher authorities to manage and respond to citizen-reported issues. The database component stores all the data related to user-reported complaints, including text descriptions, images, timestamps, and status updates. Automatic location fetching utilizes GPS to determine the geographical coordinates of the reported issues. Image classification involves the use of deep learning models to analyze and categorize images based on their content.

B. Tools Used

The research consists of two Android applications. One for the user (the citizens) and the other for the authorities (both central and city corporation authorities). The following tools were used for the development of this system:

1) Java: A widely used, reliable, and portable object-oriented programming language, popular among developers for Android app development due to its large community support.

2) Android Studio: A user-friendly and comprehensive Integrated Development Environment (IDE) for creating Android applications, compatible with Windows, macOS, and Linux operating systems.

3) Fused Location Services: A component of Google Play Services that aggregates location data from various sensors, utilized in the project to collect location information, viewable through the Google Maps application.

4) Firebase: A cloud-based database and computing service from Google, offering features such as authentication, real-time databases, and Firebase Storage, is chosen for its security, ease of use, and free access.

5) Teachable Machine: A web-based tool created by Google for developing machine learning classification models using the Tensorflow.js library, allowing for model training even with limited hardware resources. It generates compact models executable on mobile devices [10].

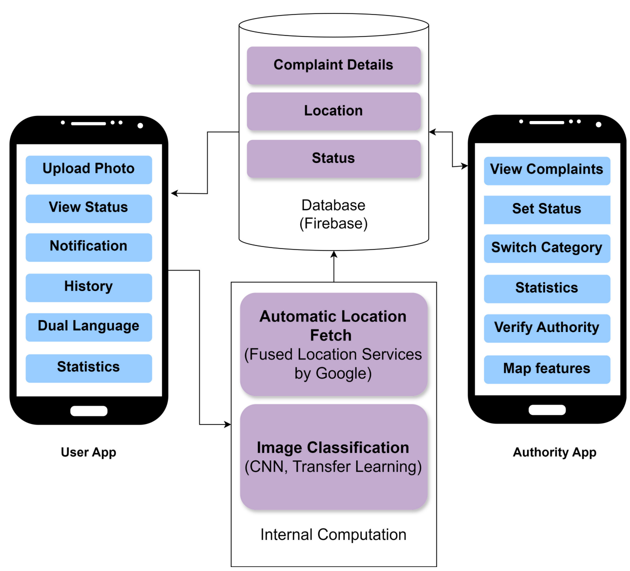


Fig. 1: System overview

C. Working Mechanism

Deep learning, or deep neural networks, refers to multiple-layered artificial neural networks that can handle large amounts of data and perform better than classical methods in several fields [11]. Teachable Machine utilizes transfer learning, a deep learning method, to develop its classification model. This process involves adapting a pre-trained model that has already been trained on a large dataset to recognize various objects and fine-tuning it on a specific dataset provided by the user [12]. In this research, the Teachable Machine has been used which employs the MobileNets model as its base model for image classification. This model was initially trained on a substantial dataset called ImageNet, which includes 1000 classes such as animals, household items, and technology. However, In this research, this base model has been customized to identify the four new classes- damaged road, flood, trash, and homeless people. The advantage of transfer learning in this research is that it allows leveraging the pre-trained model's knowledge, reducing data requirements and training time while maintaining accuracy. The pre-trained model's underlying traits, such as those learned by MobileNets to recognize the original classes, are leveraged to identify the four new classes.

MobileNets are lightweight convolutional neural network architectures designed for mobile and embedded vision applications [13]–[15]. Using MobileNets, Teachable Machine offers a potent yet lightweight framework for custom machine-learning models without significant computational resources. The foundational dataset, ImageNet, contains millions of labeled, high-resolution images [16]. The deep learning model for this paper was trained locally in the browser and deployed to the Android application.

MobileNets architecture features depthwise separable convolution, replacing standard convolutional layers. This involves depthwise convolution, which captures spatial information within each channel, followed by pointwise convolution, combining information from different channels. This decoupling reduces parameters and computations, enhancing efficiency. Fig. 2 illustrates depthwise separable convolution.

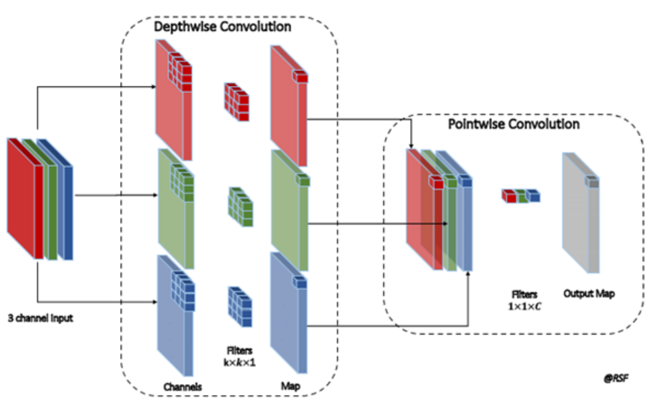


Fig. 2: Depthwise separable convolution [17]

MobileNets architectures come in various configurations with respect to their depth and width, allowing for adaptability to strike a balance between model size and performance. The term "depth" pertains to the number of layers or the extent of the neural network structure's depth. A deeper network has the ability to recognize more intricate patterns but may also lead to increased computational expenses and memory usage. The term "width" alludes to the number of channels (or filters) in each layer. Enhancing the width of the network expands its capacity to represent information, but it also entails more parameters, which in turn necessitates increased memory and computational resources. The MobileNets architecture includes an expansion factor parameter in certain versions. This parameter controls the number of input channels before the depthwise convolution operation. These architectures also frequently employ stridden convolutions to lessen the spatial dimensions of the feature maps, effectively downsampling the feature maps and reducing computational costs. At the end of the network, a global average pooling layer is used to aggregate spatial information across the feature maps. This reduces the spatial dimensions to a single vector, which is then fed into a fully connected softmax layer for classification. MobileNets architectures achieve efficiency through the use of depthwise separable convolution, controlling model depth and width, and incorporating techniques like stridden convolutions and residual connections. These design choices allow MobileNets models to attain state-of-the-art performance on various computer vision tasks while being lightweight and suitable for deployment on devices with limited resources. For these reasons, the Teachable Machine has been used in this research, with MobileNets as the pre-trained model, to generate a highly efficient and accurate deep-learning model for the classification of images.

IV. Implementation and Results

A. Implementation of the Deep Learning Model

The following steps outline the process for developing the deep learning model:

1) Dataset Collection: A benchmark dataset has been established to develop the applications. The total of four classes comprises 5494 images. Specifically, the "Flood", "Homeless People", "Damaged Road", and "Trash" classes possess 1183, 1623, 1072, and 1616 images, respectively. The collection of images was manually carried out by taking pictures using a mobile phone camera and from various sources, such as the Internet. Afterward, the images were uploaded to the Teachable Machine. The Teachable Machine conducted preprocessing on the data, which included resizing each image to a square shape to guarantee that inputs are of equal size and dimensions. Moreover, it usually adjusts the images to ensure they fall within the range of [0, 1]. The 85% of the images from each problem category were used for training purposes, while the remaining 15% were allocated for testing.

2) Transfer Learning: The Teachable Machine trains the model directly in the user's browser, incorporating concepts like "Supervised Learning" and "Transfer Learning". Utilizing MobileNets as the pre-trained model, it leverages images from ImageNet for training. MobileNets, known for its lightweight nature and depth-wise separable convolutions, ensures efficient parameter handling, enabling compatibility with mobile devices. Transfer learning adjusts the pre-trained MobileNets model to identify new data, negating the necessity of initializing weights randomly from scratch. Leveraging the pre-trained model's extensive recognition capabilities and optimal training parameters such as epoch, batch size, and learning rate results in high accuracy, even with limited datasets and lower epoch values. Given MobileNets' lightweight design, it seamlessly integrates into the Android application. The training configuration includes 10 epochs, a batch size of 16, and a learning rate of 0.001.

3) Model Export: After training, the model was exported from the Teachable Machine directly in tflite format. The tflite model enables us to deploy the model directly in the mobile application.

B. Implementation of the "CitySolution" Application

The CitySolution user application facilitates seamless interaction between citizens and municipal authorities, streamlining the reporting and resolution of civic issues. Users register on the app to report municipal issues by capturing photos and providing feedback. The app uses Google's Fused Location Service to fetch the issue's location automatically. A deep learning model embedded in the app analyzes submitted images, classifying complaints into categories like Damaged Road, Flood, Trash, or Homeless People. Users can track the status of their reports, which are initially set to "pending" and later updated to "processing" or "solved" by the city corporation. The app also provides in-app notifications, progress tracking via graphs, and a dual-language feature for user convenience. The flow chart of the user application is illustrated in Fig. 3.

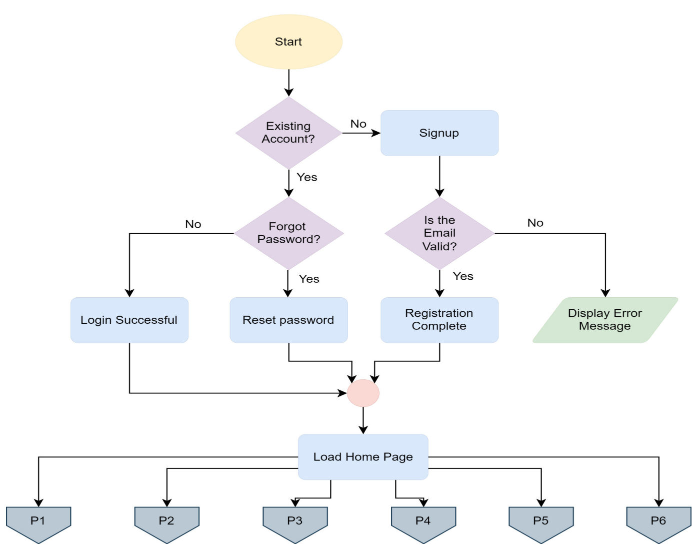


Fig. 3: Flowchart of CitySolution User Version

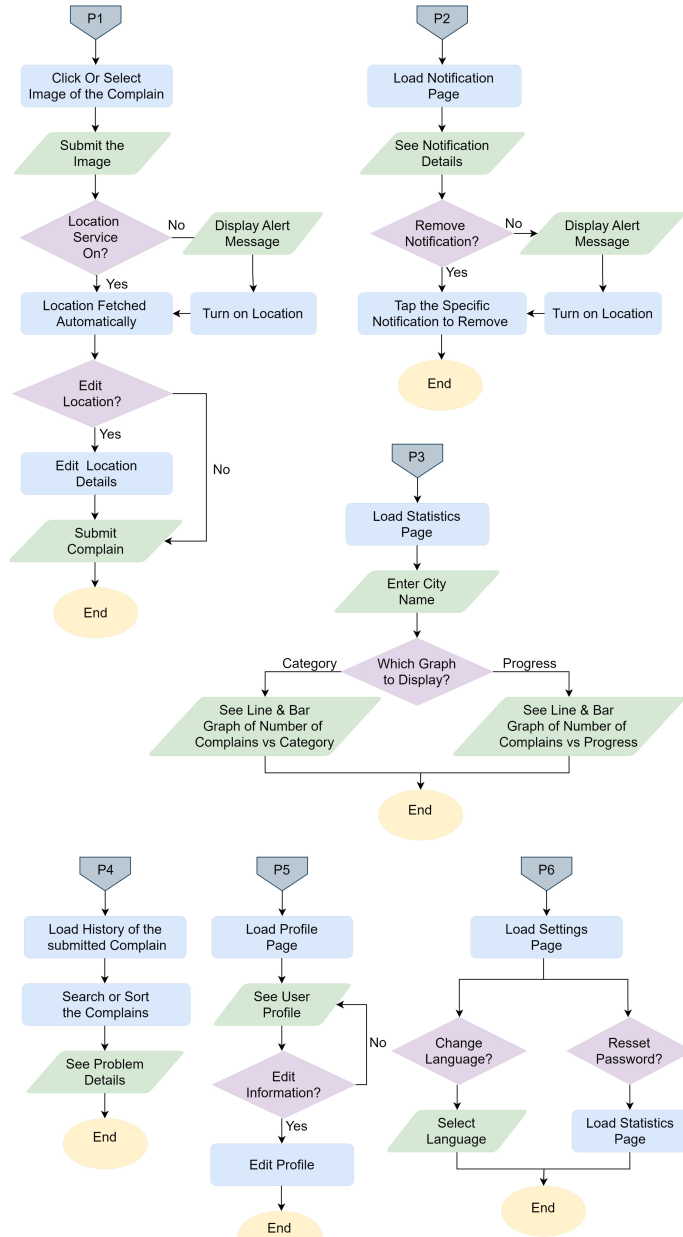


Fig. 3: Flowchart of CitySolution User Version(continued)

The CitySolution authority application allows authorities to register, view complaints submitted by citizens, and also update the progress of their actions. Using integrated Google Maps, authorities can accurately locate reported issues, improving response times. They can update complaint statuses, ensure transparency, and manually identify and address fake complaints. Authorities can also provide feedback to users, enhancing communication. The higher authority has a dedicated panel with additional authentication for monitoring system performance and verifying city corporation members. They can remove registered employees if necessary. The app includes graphical representations of complaint progress and distribution, aiding in monitoring and decision-making. The flow chart of the authority application is illustrated in Fig. 4.

C. Analysis of the results

With a total of 178 test data instances in the ”Flood” class, 244 instances in the ”Homeless People” class, 161 instances in the ”Damaged Road” class, and 243 instances in the ”Trash” class, the confusion matrix provides insights into the accuracy and effectiveness of the classification model. The confusion matrix, as illustrated in Fig. 5, presents a comprehensive overview of the classification results. Each cell in the matrix represents the count of instances where the predicted class matches the actual class. Specifically, the diagonal elements of the matrix correspond to the true positive (TP) counts, indicating the number of correctly classified instances for each class. Conversely, off-diagonal elements represent misclassifications, providing insights into the types of errors made by the classification model.

Further analysis of the confusion matrix reveals various performance metrics, including accuracy, precision, recall, and F1-score, which are summarized in Table I. The necessary formulae for calculating these performance metrics are mentioned in equations 1-4. As mentioned in Equation 5 and shown in Table I, the Matthews Correlation Coefficient (MCC) for the four classes offers additional insights into the evaluation metrics.

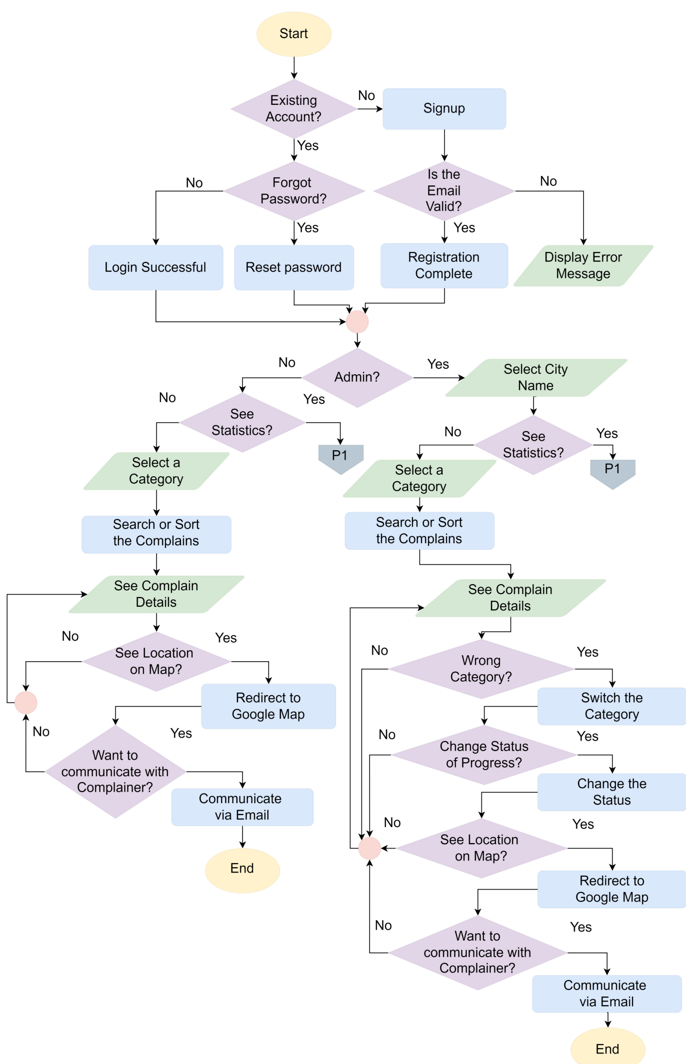
$$\text{Accuracy for Class } i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{1}$$

$$\text{Precision for Class } i = \frac{TP_i}{TP_i + FP_i} \tag{2}$$

$$\text{Recall for Class } i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

$$\text{F-1 Score for Class } i = \frac{2 \times \text{precision}_i \times \text{recall}_i}{\text{precision}_i + \text{recall}_i} \tag{4}$$

$$\text{MCC} = \frac{\text{TP}_i \times \text{TN}_i - \text{FP}_i \times \text{FN}_i}{\sqrt{(\text{TP}_i + \text{FN}_i)(\text{TP}_i + \text{FP}_i)(\text{TN}_i + \text{FN}_i)(\text{TN}_i + \text{FP}_i)}} \tag{5}$$



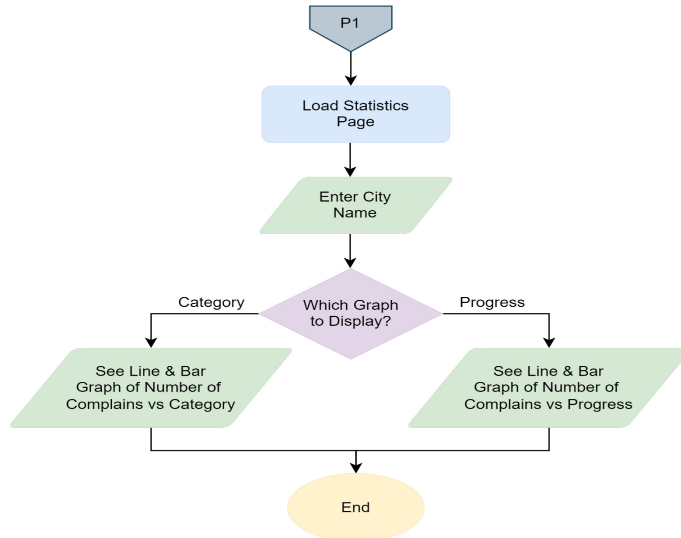


Fig. 4: Flowchart of CitySolution Authority Version

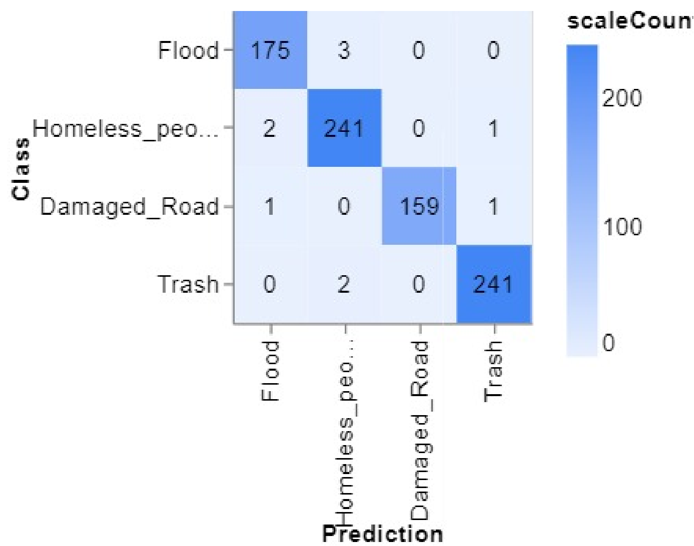


Fig. 5: Confusion Matrix

TABLE I: Performance Metrics for Four Classes

Class	Accuracy (%)	Precision (%)	Recall (%)	F1-score	MCC
Flood	99.27	98.31	98.31	0.9831	0.9785
Homeless People	99.03	97.97	98.77	0.9837	0.9768
Damaged Road	99.76	100.00	98.76	0.9938	0.9923
Trash	99.52	99.18	99.18	0.9918	0.9883
Overall(avg)	98.79	98.87	98.76	0.9881	0.9840

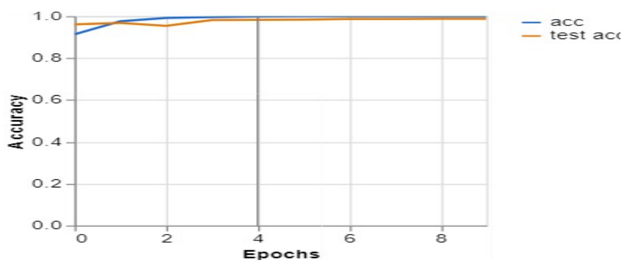


Fig. 6: Accuracy vs Epoch

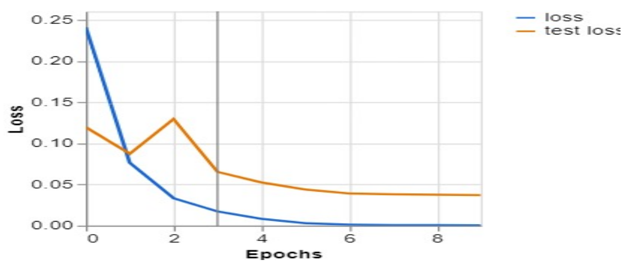


Fig. 7: Loss vs Epoch

This high-performance matrix for the four classes is achieved because of the use of transfer learning where the pre-trained model is MobileNets, a highly accurate, compact CNN model. The accuracy vs epoch and loss vs epoch in Fig. 6 and in Fig. 7 are illustrated for a better understanding of the model evaluation.

V. Conclusion

This paper bridges the gap between citizens and authorities, promoting urban development through 'Develop the City Together'. Using a deep learning model with an accuracy of 98.79%, F1 score of 0.9881 and MCC score of 0.9840, it ensures precise problem classification and solutions, enhancing accessibility. Addressing transparency, human error, and lack of centralized monitoring highlighted in [1] and [9], the system automates complaint submission and categorization, improving communication between citizens and officials. Real-time updates build trust and motivate city employees. The system detects fraudulent complaints, ensuring resource integrity. It is designed for use in Bangladesh, so it supports both English and Bengali languages. Both applications can be found at [18]. These innovations aim to make city corporations smarter and more responsive, enhancing urban development and citizen well-being.

Incorporating transfer learning through the Teachable Machine may introduce bias, affecting the model's accuracy and fairness. Its efficacy depends on the quality of pre-trained models, potentially limiting applicability. Current research focuses on four categories, which need expansion. Additionally, some Google services entail costs and have changing privacy policies. Therefore, features like mobile notifications, OTP verification, and unlimited storage were not included in the applications.

Acknowledgment

The authors would like to thank their university for providing a conducive atmosphere for this research.

References

[1] S. .S, N. Yasini, and M. Yogesh, "A web application for managing and resolving complaints of public," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 213–216, 08 2019.

[2] F. Shama, A. Aziz, and L. B. M. Deya, "Citysolution: A complaining task distributive mobile application for smart city corporation using deep learning," *SoftwareX*, vol. 27, p. 101829, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352711024002000

[3] R. Su, Q. Wang, and C. He, "Research on improving image classification technology based on deep convolutional neural network model," in *2023 7th International Conference on Electrical, Mechanical and Computer Engineering (ICEMCE)*, 2023, pp. 907–911.

[4] D. Agustian, P. Pertama, P. Crisnapati, and P. Novayanti, "Implementation of machine learning using google's teachable machine based on android," 10 2021, pp. 1–7.

[5] M. Shaha and M. Pawar, "Transfer learning for image classification," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, pp. 656–660.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807.

[7] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," 06 2018, pp. 6848–6856.

[8] Y. Anagun, N. S. Bolel, S. Isik, and S. E. Ozkan, "Deep learning-based customer complaint management," *Journal of Organizational Computing and Electronic Commerce*, vol. 32, no. 3-4, pp. 217–231, 2022.

[9] Y. Shrinewar *et al.*, "Online complaints registration and management system to municipal corporation," *International Journal of Research in Advanced Science, Engineering and Technology*, vol. 11, 2023.

[10] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen, "Teachable machine: Approachable web-based tool for exploring machine learning classification," *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218482766

[11] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[12] M. Hussain, J. Bird, and D. Faria, "A study on cnn transfer learning for image classification," 06 2018.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, ..., and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[14] M. K. Reza, A. Prater-Bennette, and M. S. Asif, "Mmsformer: Multimodal transformer for material and semantic segmentation," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 599–610, 2024.

[15] M. K. Reza, A. Prater-Bennette, and S. Asif, "Robust multimodal learning with missing modalities via parameter-efficient adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 10 2024.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," 06 2009, pp. 248–255.

[17] C. Cheuque, M. Querales, R. León, R. Salas, and R. Torres, "An efficient multi-level convolutional neural network approach for white blood cells classification," *Diagnostics*, vol. 12, p. 248, 01 2022.

[18] F. Shama and L. B. M. Deya, "Citysolution," https://github.com/Shama-33/CitySolution, 2024, gitHub repository.