



Wall Crack Detection, Solution Recommendations And Early Mold Detection Using Image Processing

¹Saritha Narayanan, ²Naseem Swabah V, ³Abhinav Manasan, ⁴Adithyan P, ⁵Sayanth Raj KK

¹Asst. Professor, ²Student, ³Student, ⁴Student, ⁵Student

Department of Computer Science and Engineering
St. Thomas College of Engineering and Technology, Kannur, Kerala, India

Abstract : Cracks and mold in buildings can lead to serious structural damage and health risks if left untreated. This project presents an AI-powered detection system using Convolutional Neural Networks (CNN) to identify different crack types—such as masonry, shear, and corrosion—and link them to specific repair solutions. It also detects early signs of mold, enabling timely intervention. The system includes repair cost estimation based on damage severity, helping users plan maintenance effectively. Designed for mobile use, it offers real-time diagnostics to support safer, more resilient buildings.

Index Terms - CNN, Crack Detection, Mold Detection, Image Processing, Structural Health Monitoring, Repair Cost Estimation..

I. INTRODUCTION

Maintaining structural integrity is critical for ensuring the safety, longevity, and functionality of buildings. Among the most common signs of structural distress are wall cracks, which if left unaddressed can lead to severe damage and expensive repairs. Mold growth, another frequently overlooked issue, poses both structural and health risks when not detected early. Together, these problems highlight the need for intelligent, real-time monitoring systems that can support timely maintenance and risk mitigation.

To address these challenges, we present an AI powered Structural Health Monitoring system designed to detect and classify wall cracks and identify early signs of mold. Built using a Convolutional Neural Network (CNN), the system applies advanced image processing techniques to recognize various crack types including masonry, shear, shrinkage, corrosion, flexural, and torsion cracks and suggests targeted solutions based on severity. Additionally, the mold detection feature enables early intervention to prevent spread, reduce health hazards, and cut down remediation costs.

By combining crack classification, mold detection, and repair cost estimation in a mobile accessible platform, the system empowers homeowners and building managers to take proactive action. This integrated approach ensures timely maintenance, minimizes long term expenses, and supports safer living and working environments.

II. PROBLEM DEFINITION

Traditional building inspection methods rely heavily on manual visual checks to detect structural issues like wall cracks and mold. These methods are time consuming, prone to human error, and often miss early signs of damage. Small cracks or mold patches frequently go unnoticed until the damage becomes severe, leading to costly repairs and potential safety hazards.

Moreover, different types of cracks such as masonry, shear, corrosion, shrinkage, or torsion require specific repair strategies. Misidentification can result in ineffective maintenance and recurring issues. Mold detection poses a similar challenge, especially in the humid environments where it spreads quickly and silently, risking both structural integrity and occupant health.

Existing systems lack precision in identifying crack types or detecting mold early, and they rarely provide actionable repair guidance. To solve this, we propose an AI-based detection system using Convolutional Neural Networks (CNN) for image analysis, enabling real-time classification of cracks, early mold detection, and cost estimation. This automated approach supports faster, more accurate maintenance decisions, enhancing structural safety and reducing long-term repair costs.

III. EXISTING SYSTEM

The current approach to wall crack detection relies on traditional image processing techniques such as grayscale conversion, Gaussian filtering, Sobel edge detection, and thresholding to identify surface-level cracks. These systems are typically implemented using Python and operate on static images to highlight visible damage. While useful for basic detection, they lack the intelligence to classify different crack types or assess severity, and are ineffective in detecting early-stage issues like mold. Additionally, they offer limited decision support and are not optimized for real-time analysis, making them less practical for proactive maintenance or large-scale deployment.

IV. RELATED WORKS

Before developing our system, we reviewed existing studies that focus on crack detection, structural analysis, and image-based diagnostics in civil infrastructure. These works explore various image processing and machine learning methods to automate inspection tasks, highlighting both capabilities and limitations. The insights gained from these studies helped shape our approach toward building an intelligent, real-time crack and mold detection system.

A. AI-Based Crack Detection and Classification in Civil Structures

This system employs image processing and Convolutional Neural Networks (CNN) to detect and classify wall cracks into categories such as hairline, medium, and wide. While it demonstrates potential in automating visual inspections, the classification is limited by crack size, which may impact the accuracy of repair decisions in complex structural cases. [2].

B. Image Processing and Machine Learning in Concrete Cube Crack Detection

Focused on detecting cracks in concrete testing cubes, this study uses CNNs for accurate damage identification. It aims to reduce testing errors, minimize material waste, and enhance structural quality assessments. However, the solution is constrained by limited sensor availability and lacks scalability for broader architectural applications. [11]

C. The system operates by capturing images of the wall surface and applying advanced image processing algorithms to analyze the visual data

This approach uses grayscale conversion, Gaussian filtering, Sobel edge detection, and thresholding to identify wall surface cracks. Implemented in Python, the system offers a user-friendly interface for homebuyers to assess property conditions and estimate repair costs. While effective for surface-level detection, it lacks the intelligence to differentiate crack types or detect early-stage issues like mold [1].

TABLE I. LITERATURE SURVEY TABLE

PAPER	DESCRIPTION	AUTHOR
AI Based Crack Detection And Correction System.	Classifies cracks into categories like hairline,medium,wide;limited in deeper crack analysis or type specific classification.	Ashok Sutagundar, Rashmi M. Kittali
Image Processing and Machine Learning in Concrete Cube Crack Detection	Focuses on crack detection in concrete cubes; improves material testing accuracy; lacks scalability and sensor availability.	Meenakshi Somnadh Patil, Dr.R.B.Ghongade
The System Operates by Capturing Images Of The Wall Surface And Applying Advanced Image Processing Algorithms to Analyze The Visual Data	Detects visible cracks; useful for homebuyers; limited to basic detection and lacks classification or mold identification features	Wiwit Suksangaram, Worawut Yimyam

V. PROPOSED SYSTEM

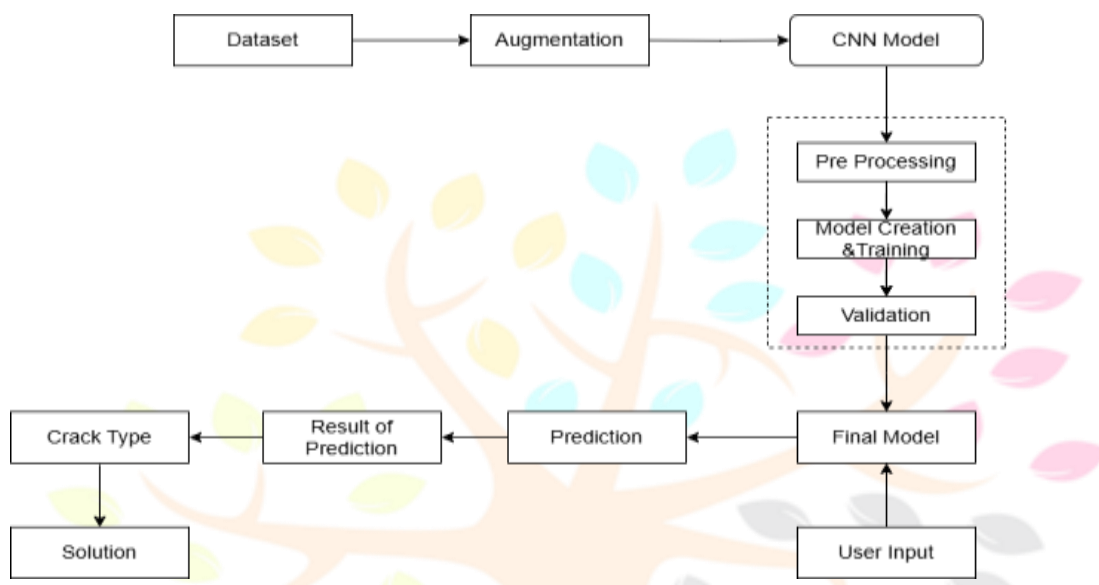
The proposed system is designed to effectively detect and classify various types of wall cracks and early signs of mold using a Convolutional Neural Network (CNN) implemented with TensorFlow and Keras. The system will leverage advanced image processing techniques to analyze images captured from building surfaces, allowing for the identification of specific crack types, including structural crack, shear crack, crazing crack, hairline crack, horizontal crack, vertical crack and spalling. By utilizing a robust dataset, the system will be trained to recognize these features accurately, providing reliable classification results. In addition to crack detection, the system will also incorporate early mold detection capabilities, addressing potential health risks and structural damage. The system also includes an approximate repair cost estimation feature, providing users with cost projections for repairs based on the severity and type of detected cracks or mold. The application will feature a user-friendly interface that enables users to upload images for real-time analysis, generating immediate feedback on the condition of the walls. Furthermore, the system will provide tailored recommendations for repair solutions based on the identified crack types and mold presence, promoting proactive maintenance and extending the lifespan of building structures. By integrating machine learning with practical applications in

building maintenance, the proposed system aims to enhance the efficiency and effectiveness of structural health monitoring, contributing significantly to the field of civil engineering.

A. Architecture

An architecture diagram visually represents the structure and interactions within a system. It provides a high-level view of the system's components, their relationships and communication flows, offering insights into how the system functions and its scalability, resilience and security. Architecture diagrams are essential for planning and aligning stakeholders, as they help developers, architects and business teams understand both the overall design and specific aspects like data flow, integration points and dependencies. They're also useful for troubleshooting, as they highlight possible bottlenecks and vulnerabilities in the system.

Figure 1. Architecture Diagram



1. **Dataset:** This is the collection of labeled images of different types of cracks and mold. The dataset is essential for training the CNN model to recognize and classify various crack types and detect mold. A well-curated dataset with diverse examples will improve the model's performance.
2. **Augmentation:** Data augmentation techniques are applied to increase the diversity of the dataset. This step involves transforming the images (e.g., rotating, flipping, scaling) to make the model more robust and capable of handling various image orientations and conditions.
3. **Pre-Processing:** The images are processed to make them suitable for training. This may include resizing, normalization and contrast adjustments, ensuring that the images are in a consistent format that the model can process efficiently.
4. **Model Creation and Training:** The CNN model is built and trained on the pre-processed images. The model learns to extract features and patterns associated with each crack type and mold presence through multiple training epochs, adjusting its internal parameters to minimize error.
5. **Validation:** During training, the model's accuracy is evaluated on a validation dataset. This step ensures that the model is not overfitting and can generalize well to new data. The validation set helps in fine-tuning the model.
6. **Final Model:** After satisfactory training and validation, the final model is saved for use in predictions. This version of the model is optimized for deployment, allowing it to be used on new images for crack and mold detection.
7. **User Input:** In real-world usage, the user can input images of walls or surfaces with suspected cracks or mold. These images are fed into the model for analysis.
8. **Prediction:** The model processes the input images and provides predictions on the type of crack or the presence of mold.
9. **Result of Prediction:** The output of the prediction indicates the specific type of crack detected (e.g., masonry, shear, corrosion) or the presence of mold. This information helps in assessing the structural issues.

10. **Crack Type and Solution:** Based on the type of crack identified, the system suggests solutions tailored to the specific type. This guidance helps the user take appropriate remedial actions, contributing to timely maintenance and structural health.

B. Dataset Collection

The first phase involves the meticulous collection of a diverse dataset comprising images of wall cracks and mold. This dataset will include various crack types, such as masonry, shear, corrosion, shrinkage, column, flexural and torsion cracks, as well as images indicating mold growth. Each image will be labeled to represent the specific type of crack or mold present. To enhance the dataset's robustness, data augmentation techniques, including rotation, flipping, zooming and shifting, will be employed using Keras's ImageDataGenerator class. This step ensures that the model can generalize well to unseen images by simulating a wider range of scenarios during training.

C. Preprocessing

Preprocessing is conducted using OpenCV to enhance the quality of images before they are fed into the Convolutional Neural Network (CNN) for training. The process begins with loading images and resizing them to a uniform dimension of 224x224 pixels to ensure consistency across the dataset. Images are then converted to grayscale to simplify the data and reduce computational complexity, allowing the model to focus on structural features rather than color variations. Following this, normalization of pixel values is performed to scale them to a range of 0 to 1, which aids in improving model convergence during training. Various data augmentation techniques are also applied to diversify the dataset, including horizontal flipping, random rotations, shifting, scaling to simulate zoom effects and brightness and contrast adjustments. These techniques help the model generalize better to various real-world conditions. Additionally, each image is labeled according to the specific type of crack or mold present, facilitating accurate classification during model training. By employing these preprocessing techniques with OpenCV, the project aims to create a high-quality dataset that significantly enhances the CNN's ability to accurately detect and classify different types of wall cracks and early signs of mold, thereby contributing to the overall effectiveness of the detection system.

D. Model Development

In this phase, we will design a CNN architecture using TensorFlow and Keras. The CNN will consist of multiple layers, including convolutional layers to extract features from the images, activation layers (such as ReLU) to introduce non-linearity, pooling layers to reduce dimensionality and dropout layers to prevent overfitting. Batch normalization layers may also be integrated to stabilize and accelerate the training process. The final layer of the CNN will be a softmax layer, which will classify images into distinct categories corresponding to the identified crack types or the presence of mold.

E. Training The Model

The dataset will be partitioned into training, validation and test sets. The training set will be used to teach the model to recognize different crack types and mold, while the validation set will be used to fine-tune hyperparameters and monitor the model's performance to prevent overfitting. The training process will involve defining a loss function (such as categorical crossentropy) and an optimizer (like Adam or RMSprop), both available in Keras. The model will be trained over several epochs, with continuous evaluation on the validation set to track improvements in accuracy and loss.

F. Evaluation And Testing

Once training is complete, the model's performance will be evaluated on the test dataset. Performance metrics such as accuracy, precision, recall and F1-score will be calculated to assess the model's ability to detect and classify the various crack types and mold effectively. Confusion matrices will be utilized to visualize the model's performance across different classes, providing insights into areas where the model may need further improvement.

G. Deployment

Once the model has been validated for effectiveness, it will be integrated into a user-friendly mobile application developed using Flutter. This application will allow users to upload images of walls for real-time analysis. The trained CNN model will be packaged using TensorFlow Lite, optimizing it for mobile use to ensure efficient performance and responsiveness. Upon receiving an image, the application will utilize the model to detect and classify any cracks or signs of mold present in the image. Users will receive immediate feedback on the condition of the wall, including a detailed report on the types of cracks detected, the severity of the damage and tailored repair recommendations. This interactive interface will facilitate proactive maintenance efforts, empowering users to take timely action based on the system's analysis.

VI. SYSTEM IMPLEMENTATION

During the system implementation phase, multiple components were integrated to ensure smooth functionality. The first step involved developing the user interface in Flutter, enabling users to upload images of wall cracks or mold easily. Simultaneously, the backend in Django was configured to handle API requests, process images and store necessary data in a database. The deep

learning model, trained using TensorFlow/Keras, was deployed in the backend to classify different types of cracks and detect mold. The system was designed to handle real-time predictions while ensuring scalability and efficiency

A. User Input

The system begins by collecting user inputs, which play a crucial role in getting the best output. These inputs include:

- **Image Upload:** Users provide an image of a wall with cracks or mold using the mobile application. This image serves as the primary input for the system to analyze and classify the type of defect.
- **Feedback Submission:** After receiving the classification results and recommendations, users can provide feedback on the accuracy of the detection. This helps improve the model over time.
- **Login Credentials:** Users must enter their login details (username and password) to access the system, ensuring secure access to their data and past reports.

B. Data Flow and Integration

User Input Handling: Users can upload images of wall cracks or mold through the Flutter-based mobile application. The system verifies the image format (JPG, PNG, JPEG) before processing. Along with image uploads, users can provide feedback on detection accuracy and seek clarification regarding wall conditions. If an invalid format is detected or the upload fails, the system prompts an "Invalid Image" message, ensuring proper input handling.

Model Execution: The CNN model processes the image and classifies it into different crack types (e.g., structural cracks, shear cracks, hairline cracks etc.) or detects the presence of mold. The model extracts relevant features and assigns a confidence score to the prediction. If uncertainty is high, the system may recommend retaking the image under better lighting conditions for improved accuracy.

Result Display: The mobile application displays the detected crack type or mold presence along with suitable repair and maintenance recommendations retrieved from the database. Users can re-upload an image for further evaluation and provide feedback on the classification results. The system ensures a seamless and user friendly experience, assisting property owners and maintenance professionals in making informed decisions about structural health.

C. User App

The Wall Crack and Mold Detection User App, developed using Flutter, is designed to provide a seamless and intuitive experience for users who need to detect wall cracks and mold through image analysis. The app features a clean and user-friendly interface, making it easy for users to navigate through different functionalities. Users can either capture an image using their device's camera or upload an existing image from their storage. Before submission, the app verifies whether the uploaded image is in a supported format (JPG, PNG, or JPEG) to ensure compatibility and prevent errors.

- Once the image is uploaded, the app establishes a secure connection with the backend, ensuring efficient and reliable data transmission. While the image is being processed, the app displays a loading indicator, informing users that the system is analyzing the wall condition. This real-time feedback enhances user engagement, reducing uncertainty while they wait for the classification results. The app is optimized for quick processing and multiple submissions, allowing users to analyze multiple images if needed.
- After processing, the app retrieves the predicted crack type or mold detection results and displays them in a clear and structured format. The classification results include the detected crack type (e.g., masonry cracks, shear cracks, corrosion cracks) or mold presence, ensuring users receive accurate and relevant information about their wall condition. Additionally, the app provides repair and maintenance recommendations specific to the detected issue, helping users take appropriate action. These recommendations are retrieved from a database of pre-stored repair guidelines, ensuring users receive useful and problem-specific solutions.
- Beyond image analysis, the app includes interactive features such as feedback submission and query handling. Users can provide feedback about their experience using the app, helping improve future updates. Additionally, the app allows users to submit queries related to their wall conditions, enabling them to engage with the system more effectively. These features enhance the overall usability of the app, making it more than just a diagnostic tool but also an informative and interactive platform for users.
- The Flutter based design ensures that the app is lightweight, responsive and optimized for both Android and iOS devices. Its smooth animations, real-time interactions and structured data flow ensure a consistent user experience. With ongoing improvements and updates, the app aims to provide a more efficient, accessible and reliable solution for users looking to analyze wall conditions quickly and accurately while also receiving helpful maintenance recommendations for their detected issue.

D. Admin Website

The Admin Website, developed using Django, provides administrators with tools to manage user details, handle feedback and queries and oversee system functionality. It features a secure login system to restrict access to authorized personnel only. The dashboard offers an intuitive interface, allowing admins to efficiently navigate and manage various system components.

- One of the key functionalities of the Admin Website is managing user details. Administrators can view a list of registered users, ensuring compliance with system policies. If necessary, they can modify user details, deactivate accounts, or handle account-related issues to maintain the integrity and security of the platform. This ensures that only genuine users have access to the system.
- The Admin Website also allows administrators to manage user feedback and queries. Users can submit feedback on the app's functionality and admins can review this feedback to understand potential improvements. Additionally, users can ask questions related to wall crack types, mold detection, or app usage and admins can respond to these queries, ensuring users receive the necessary guidance. This feature enhances user engagement and improves the overall experience.
- While administrators do not have access to user-uploaded images or prediction history, they play a crucial role in ensuring the platform runs smoothly. They can monitor system activity, address technical issues and report persistent errors to the development team. By keeping track of platform stability, admins help ensure that the system remains efficient and reliable.
- The Admin Website is designed to be lightweight, secure and efficient, focusing on user and feedback management rather than modifying core model functionality. Future enhancements may include better analytics for tracking user engagement, automated feedback analysis and improved query management. The admin panel plays a vital role in maintaining the Wall Crack and Mold Detection System, ensuring smooth operations and a user-friendly experience.

VII. RESULT

The Wall Crack and Mold Detection System underwent a thorough result analysis to assess the accuracy, reliability and efficiency of its predictions. The CNN-based deep learning model was evaluated on a diverse dataset of wall crack and mold images, ensuring that it could accurately classify various crack types and detect mold formations. Various performance metrics such as accuracy, precision, recall and F1-score were used to measure the model's effectiveness. A confusion matrix was generated to identify areas where the model misclassified images, helping detect patterns in false positives and false negatives. This analysis was crucial for refining the model, improving feature extraction and making necessary enhancements to boost predictive capabilities. Additionally, testing under different lighting conditions, surface textures and camera angles ensured the system's robustness and generalizability for real-world applications.

Beyond model performance, the overall system efficiency was also analyzed to ensure a smooth and reliable user experience. The response time of the Flutter app, Django backend and deep learning model was monitored to evaluate system performance under different conditions. Tests were conducted to measure image processing speed, API response time and treatment recommendation retrieval efficiency. Additionally, user feedback was collected to assess how well the app conveyed predictions and treatment tips. Any discrepancies in model predictions, slow response times, or usability issues were documented for further optimization. This continuous analysis helped improve system accuracy, user engagement and overall functionality, ensuring the Lung Abnormality Detection System remained a highly effective and user-friendly medical tool.

A. Admin Page

The Admin Panel of the Wall Crack and Mold Detection System, built using Django, provides administrators with a centralized platform to manage user details and handle feedback. The panel features a secure dashboard where admins can view registered users and review feedback submissions related to crack types, mold detection and app functionality.

While administrators do not have access to user-uploaded wall images, they play a crucial role in ensuring the system operates smoothly by addressing user concerns, resolving technical issues and maintaining platform security. The panel is designed to be lightweight, efficient and easy to navigate, allowing administrators to monitor system performance and improve user experience effectively.

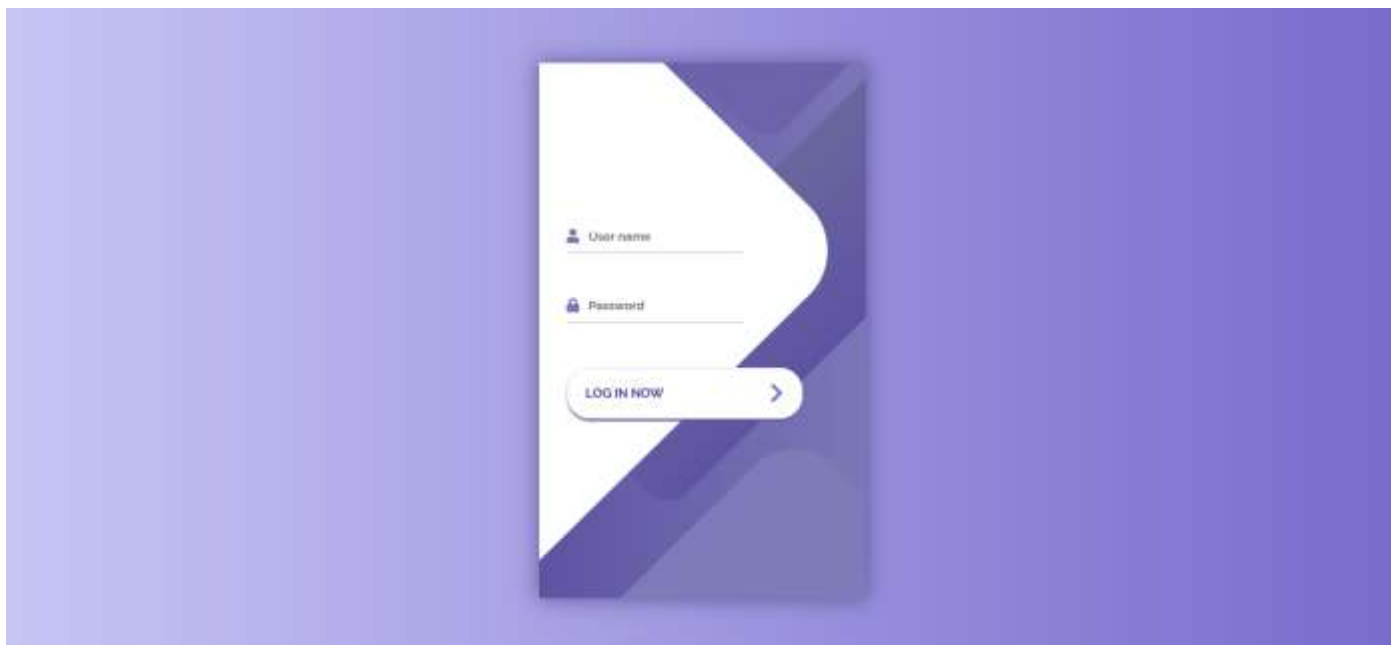


Figure 8.1: Admin Login form

The Admin Login Page (figure 8.1) of the Wall Crack and Mold Detection System serves as a secure authentication gateway for administrators to access the admin panel. It features a simple and professional design, requiring an email and password for login. To ensure secure access management, the interface includes a “Forgot Password” option for easy password recovery. The background design aligns with the system’s purpose, reinforcing its role in building maintenance and structural health monitoring. Once logged in, administrators can efficiently manage user details, monitor system activity and oversee performance. They have the ability to review and respond to user feedback regarding wall crack and mold detection results, ensuring continuous improvement of the system. Additionally, admins can analyze system logs, track errors and implement necessary updates to maintain smooth functionality. Security is a key aspect of the admin panel, allowing administrators to monitor login attempts, prevent unauthorized access and ensure data privacy. By managing user interactions and addressing concerns, the admin panel plays a crucial role in maintaining system efficiency, reliability and user satisfaction.

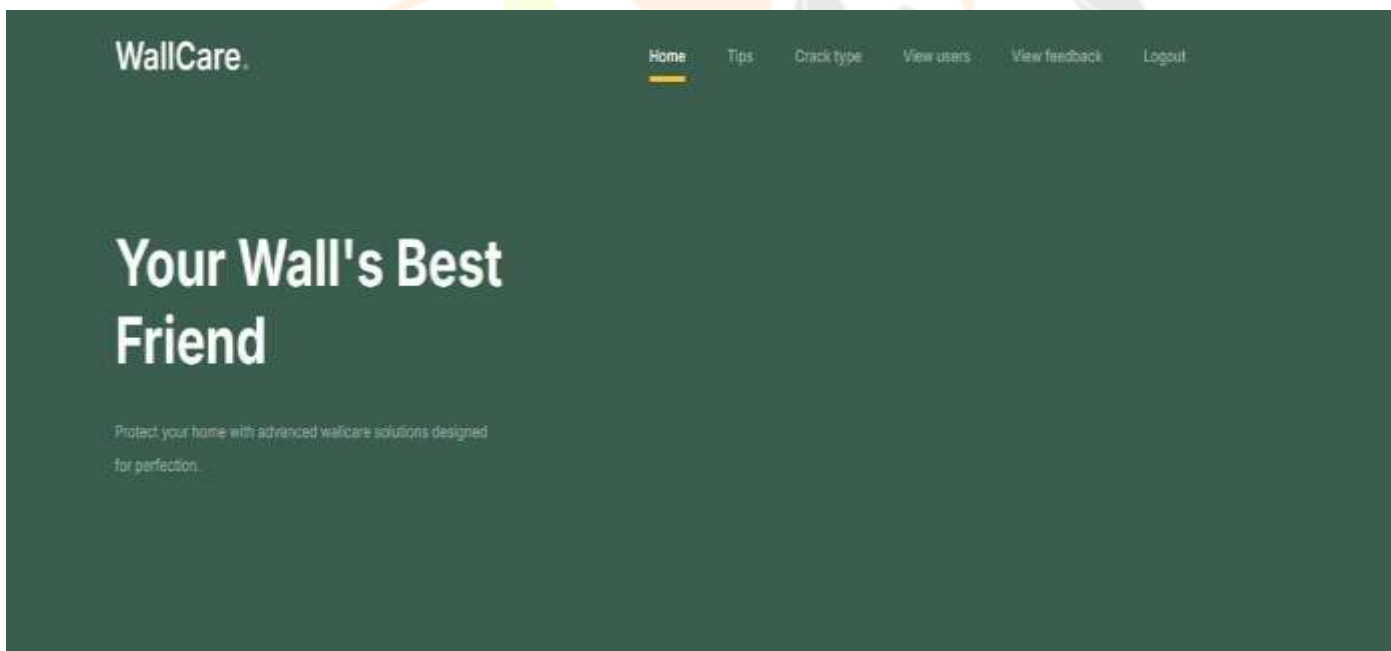




Figure 8.2: Homepage

The Admin Home Page (figure 8.2) of the WallCare system provides a centralized interface for administrators to manage and oversee various aspects of the platform. Designed with a clean and professional layout, the home page serves as the primary dashboard, offering easy navigation to essential features such as crack type classification, user management, feedback review and maintenance tips. The top navigation bar includes options like “Tips,” “Crack Type,” “View Users,” “View Feedback,” and “Logout,” ensuring quick access to different administrative functions. The homepage prominently displays a welcoming message, emphasizing the system’s commitment to advanced wall care solutions. By offering an intuitive and user-friendly experience, the admin home page enables seamless monitoring and management, ensuring smooth system operation and enhanced user support.

Crack Type	Details	Image	Actions
shearcrack	Occur due to lateral forces. Usually diagonal and appear near window, doors, or beam column joints. Indicates serious structural issue.		Delete View
crazing crack	crazing cracks are fine surface level fractures that appear in materials like concrete, ceramics and paint		Delete View

[Add New](#)

Figure 8.3: crack type and solution

The Crack Type Management Page (figure 8.3) in the WallCare admin panel allows administrators to oversee different types of wall cracks and their corresponding details. The interface presents a structured table with columns for the crack type, description, associated image and administrative actions. Each row represents a specific crack type, displaying its name, characteristics and an image for easy identification.

The "View" button serves as an option for adding or managing solutions related to a specific crack type. When clicked, it allows administrators to input appropriate repair methods or solutions tailored to the crack type, ensuring users receive accurate guidance for wall maintenance. Additionally, the "Delete" button enables administrators to remove crack types from the system if needed. The "Add New" button at the bottom provides an option to introduce new crack types, ensuring the database remains updated with relevant classifications. This page streamlines the process of crack type management, facilitating better user assistance and maintenance recommendations.

B. User App

The WallCare App is designed to assist users in identifying and managing wall cracks through advanced AI-based analysis. Users can upload images of their walls and the app detects various crack types along with mold detection, the app provides detailed insights into each crack type and suggests appropriate repair solutions to prevent further damage. The user-friendly interface ensures accessibility for homeowners, contractors and construction professionals, promoting early detection and effective maintenance of walls and structures.

IJNRD
Research Through Innovation



Figure 8.5: App Login

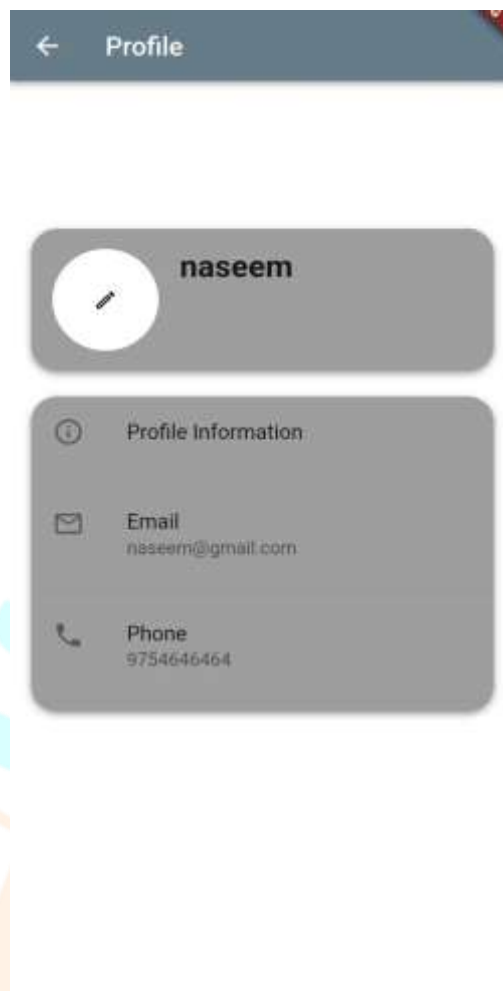
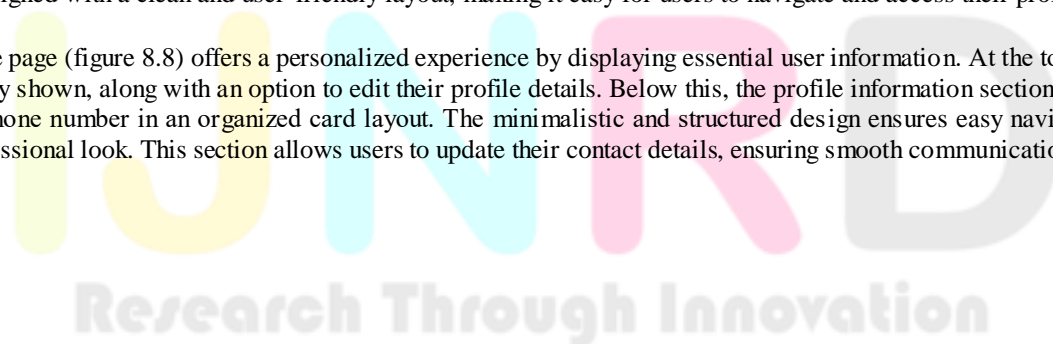


Figure 8.8: User Account

The user authentication system in app consists of two main functionalities: Login and Registration. The Login page (figure 8.5) allows existing users to access their accounts by entering their username and password. It includes an eye icon for toggling password visibility, a "Forgot Password?" option for account recovery and a "Register here" link for new users who don't have an account. The interface is designed with a clean and user-friendly layout, making it easy for users to navigate and access their profiles quickly.

The user profile page (figure 8.8) offers a personalized experience by displaying essential user information. At the top, the user's name is prominently shown, along with an option to edit their profile details. Below this, the profile information section presents the user's email and phone number in an organized card layout. The minimalistic and structured design ensures easy navigation while maintaining a professional look. This section allows users to update their contact details, ensuring smooth communication within the app.



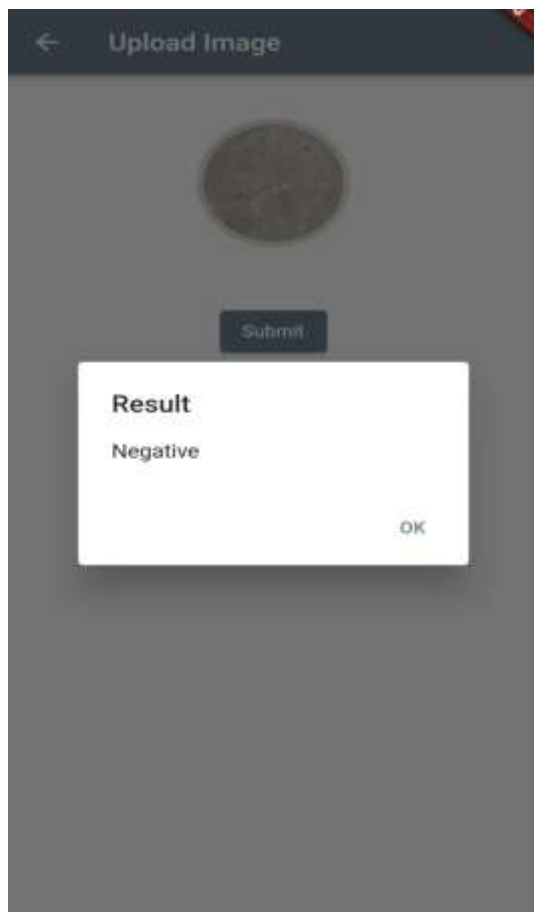


Figure 8.9: Defect Negative

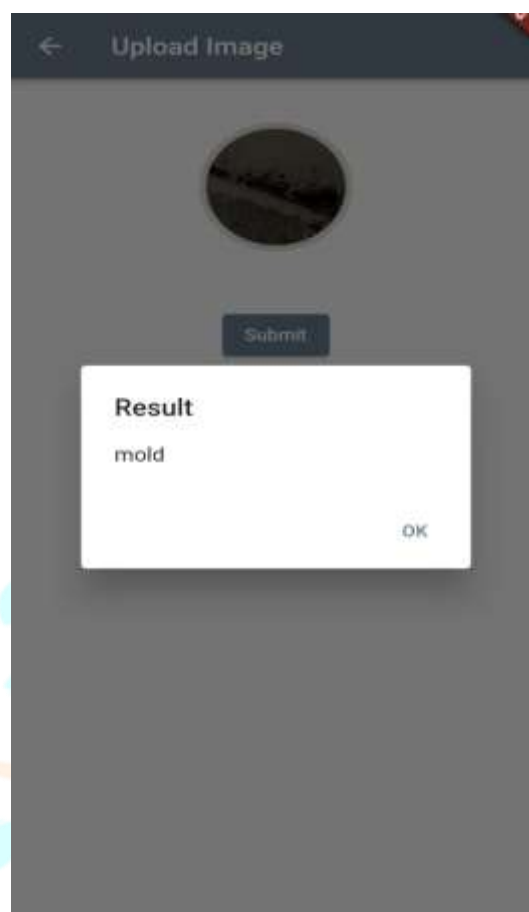


Figure 8.10: Mold

Above figure (Figure 8.9) shows a "Negative" result, meaning that no visible cracks or mold were found in the uploaded image. This indicates that the system did not detect any structural damage, suggesting that the wall surface in the image is intact and does not require immediate repairs. The negative result helps users confirm the condition of their walls.

In figure (Figure 8.10) the system detects the presence of mold on the wall. The result displayed in the dialog box is "mold," which signifies that the uploaded image contains fungal growth. The identification of mold is crucial, as it can indicate underlying moisture problems that may require intervention to prevent further damage or health risks.

Below figure (Figure 8.11) shows the detection result of a structural crack in the wall crack detection system. When an image is uploaded, the system processes it and classifies the crack type, displaying the result in a dialog box. Structural cracks are deep and indicate serious structural issues. The system can also detect six other types of cracks: crazing, shear, horizontal, vertical, hairline and spalling. This feature helps users identify wall defects and take necessary repair actions.

The figure (Figure 8.12) illustrates a crack that occurs due to excessive stress, often near openings or structural joints. Proper reinforcement is necessary to prevent further damage. The recommended repair method involves adding support, such as steel stirrups or carbon fibers, to strengthen the affected area. The repair cost varies depending on the size of the crack, with larger cracks requiring more extensive reinforcement and higher expenses.

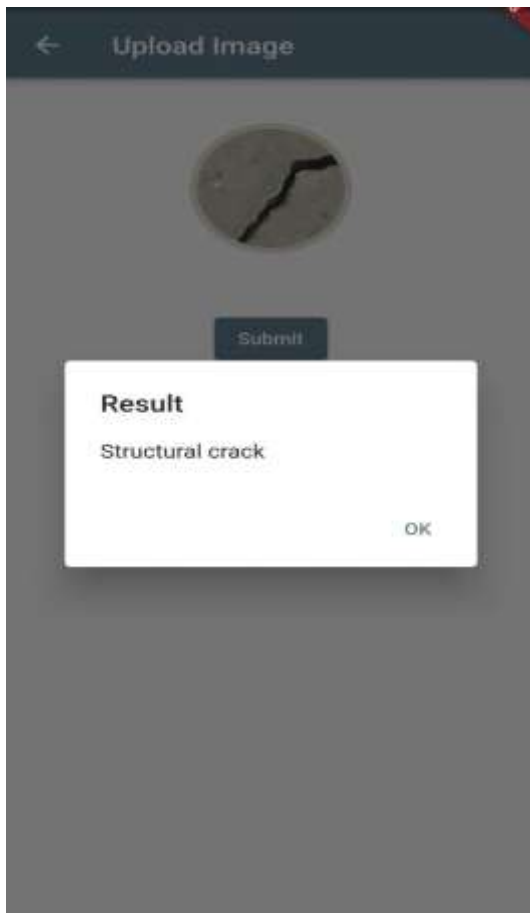


Figure 8.11: Structural crack



Figure 8.12: View Type, Solution And Repair Cost

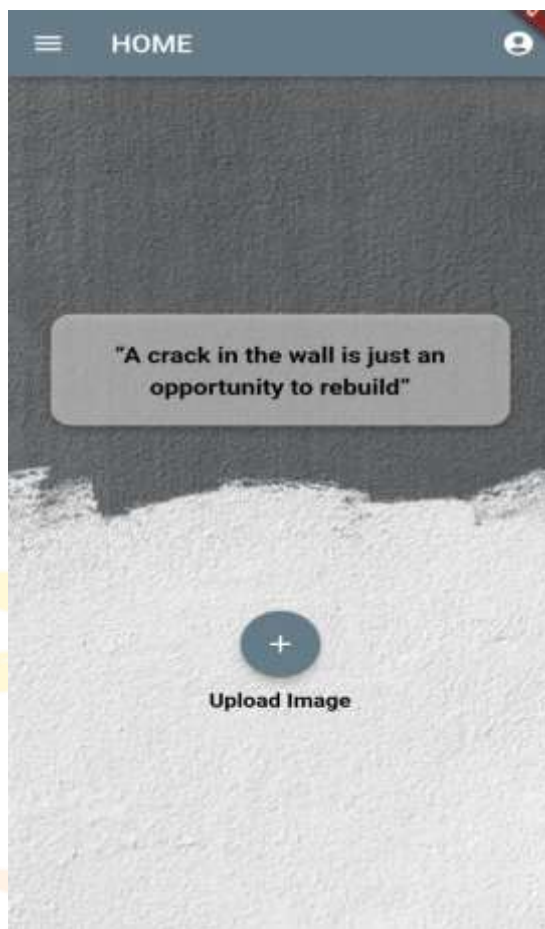


Figure 8.13: App Home

The home page (figure 8.13) of the Wall Crack Detection App provides users with a simple and intuitive interface to begin the wall inspection process. The background features a textured wall design, reinforcing the app's purpose. At the center, a prominent "Upload Image" button allows users to select and upload an image of a wall surface for crack detection and analysis. The clean design ensures ease of use, making it accessible for all users.

VIII. CONCLUSION

The proposed system for wall crack and mold detection represents a significant advancement in building maintenance and structural health monitoring. By leveraging a Convolutional Neural Network (CNN) developed with TensorFlow and Keras, along with advanced image preprocessing techniques using OpenCV, the system can accurately detect and classify various types of wall cracks, including masonry, shear, corrosion and more, as well as early signs of mold growth. This detailed detection is essential for addressing structural issues promptly, helping to prevent further deterioration, costly repairs and health risks associated with mold exposure. Built using Flutter for cross-platform accessibility on both Android and iOS, the mobile application makes the solution accessible to both professionals in building maintenance and individual property owners.

The project's comprehensive approach covering data collection, preprocessing, model training and user feedback integration ensures that the solution remains accurate, adaptable and user friendly. By focusing on early detection, the system promotes proactive solutions, allowing for timely interventions that can extend the lifespan of structures, enhance safety and reduce repair expenses. Economic analysis underscores the project's financial viability, with potential savings on repairs and a growing market demand for efficient, technology driven building maintenance solutions. Operationally, the system's design promotes ease of use and smooth integration into existing workflows, supported by training and resources to assist users of all technical levels.

Prioritizing user satisfaction, the project also incorporates ongoing engagement and iterative improvements to provide a reliable and enduring tool for structural monitoring. The real world impact of this system particularly in reducing structural degradation and fostering safer environments demonstrates considerable promise. In meeting the current needs of the construction and property

management sectors, the project establishes a solid foundation for future innovations in preventive building maintenance technology. By delivering a solution that combines accuracy, accessibility and proactive management, the proposed system positions itself as an invaluable asset for advancing building infrastructure health and safety.

REFERENCES

- [1] Ashok Sutagundar, Rashmi M. Kittali "AI Based Crack Detection and Correction System" , Bagalkote, Karnataka 2023 NMITCON.
- [2] Meenakshi Somnadh Patil, Dr.R.B.Ghongade "Image Processing and Machine learning in concrete cube crack detection " , Pune 2023 ICCAMS.
- [3] Wiwit Suksangaram, Worawut Yimyam "The system operates by capturing images of the wall surface and applying advanced image processing algorithms to analyze the visual data" 2023 18th iSAI-NLP
- [4] Vaughn Peter Golding, Zahra Gharineiat , Hafiz Suliman Munawar and Fahim Ullah." Crack Detection in Concrete Structures Using Deep Learning ", MDPI Sustainability 2022.
- [5] Jun Wang, Yail J. Kim and Chao Liu. "Deep Learning for Detection and Characterization of Cracking in UHPC" , ACI Structural and Materials Journals-17 March 2022.
- [6] Y. Fujita, Y. Mitani, Y. Hamamoto. "A Method for Crack Detection on a Concrete Structure", IEEE Xplore: 18 September 2022.
- [7] SUN LIANG 1 , XING JIANCHUN1 , (Member, IEEE) and ZHANG XUN2. An Algorithm for Concrete Crack Extraction and Identification Based on Machine Vision", IEEE Access 201.
- [8] Jung Jin Kim, ,Ah-Ram Kim, Seong-Won Lee. "Artificial Neural Network-Based Automated Crack Detection and Analysis for the Inspection of Concrete Structures", Appl. 10(22), 8105, 2022.
- [9] Paulius Palevičius, Mayur Pal, Mantas Landauskas, Ugnė Orinaitė, Inga Timofejeva and Minvydas Ragulskis. "Automatic Detection of Cracks on Concrete Surfaces in the Presence of Shadows", Sensors (Basel). 2022 May; 22(10): 3662. Published online 2022 May 11. doi: 10.3390/s22103662.
- [10] C.A. Issa , "Methods of crack repair in concrete structures" , Failure, Distress and Repair of Concrete Structures, Page: 169-193 Publication Year 2009, Available online 27 March 2014.
- [11] Arun Mohan , Sumathi Poobal, "Crack detection using image processing: A critical review and analysis", Alexandria Engineering Journal, Volume 57, Issue 2, Pages 787-798, June 2018,.
- [12] Butcher, J.B.; Day, C.R.; Austin, J.C.; Haycock, P.W.; Verstraeten, D.; Schrauwen, B. (2014). Defect Detection in Reinforced Concrete Using Random Neural Architectures. Computer-Aided Civil and Infrastructure Engineering, 2014.
- [13] Hongxia Li, Weixing Wang, Mengfai Wang, Limin Li, A review of deep learning methods for pixel-level crack detection, Journal of Traffic and Transportation Engineering, Vol 9, Issue 6, pages 945-968, 2022.

