Automated Road Extraction & Change Detection

Ms. Mandale Aarti R.[1], Ms. Shinde Aditi U.[2], Ms. Shinde Kirti P. [3], Ms. Shirtsath Sakshi D.[4], Mr. S. T. Patil[5],
Dept. of Electronics and Telecommunication Engineering
SNJB's Shri Hiralal Hastimal (Jain Brothers, Jalgaon) Polytechnic Chandwad, Dist. Nashik, India

Abstract

This project proposes an IoT-enabled Android application for automated road extraction and condition monitoring using satellite and on-ground imagery. The hardware module employs NodeMCU and ESP32-CAM to capture real-time road surface images, which are uploaded to Firebase for cloud-based storage and processing. A Convolutional Neural Network (CNN), trained in Python and converted into TensorFlow Lite (TFLite), performs feature extraction and classifies road quality into good, moderate, or poor categories. To further enhance decision-making, a questionnaire interface within the app collects contextual data such as traffic load, usage duration, and location information. A Decision Tree algorithm integrates these metadata inputs with CNN outputs, producing a more reliable classification. The system delivers a comprehensive road condition report, including extracted features, quality scores, and maintenance alerts, thereby combining IoT sensing, cloud integration, and AI-driven analytics to support smart infrastructure management.

Keywords

Agriculture Robot, CNN Algorithm, Firebase, ESP 32 Camera, Node MCU, Android App, Road Extraction, Java, XML, ML Processing.

1. *Introduction*

Road infrastructure plays a vital role in supporting economic growth, transportation, and public safety. However, roads are highly prone to deterioration due to heavy traffic, weather conditions, and lack of timely maintenance. Conventional methods of road inspection rely on manual surveys and government audits, which are often time-consuming, costly, and prone to human error. With the growing availability of satellite imagery from agencies such as ISRO, combined with advancements in IoT devices and artificial intelligence, automated solutions have emerged as a reliable alternative for large-scale and real-time road condition monitoring.

This project introduces an integrated system that leverages both satellite images and IoT-enabled on-ground data capture for road extraction and change detection. NodeMCU and ESP32-CAM modules are used to capture real-time road surface images, which are stored in Firebase for further analysis. A Convolutional Neural Network (CNN), trained using Python and deployed via TensorFlow Lite (TFLite), automatically detects road features and classifies their condition into good, moderate, or poor categories. To improve the robustness of predictions, the system incorporates a user questionnaire that collects metadata such as traffic load, road usage duration, and location details. A Decision Tree algorithm fuses this contextual information with CNN results to generate accurate road quality assessments.

By combining IoT sensing, cloud integration, and AI-driven analytics, the proposed Android application offers an efficient and scalable approach to road condition monitoring. The system not only delivers automated extraction and detection of road quality but also generates detailed maintenance reports with alerts. This ensures proactive infrastructure management and supports smart city initiatives, reducing the risks associated with poor road conditions while optimizing government and municipal resources.

2. *Literature Review*

Road infrastructure monitoring has increasingly relied on deep learning and high-resolution remote sensing to overcome the limitations of manual inspection. Wang et al. (2022) proposed a Dual-Decoder U-Net (DDU-Net) for road extraction, which combined a main decoder for coarse features and a secondary decoder for fine details, significantly improving mIoU and F1-scores compared to conventional CNN architectures [1].

To further enhance feature representation, Mahara et al. (2024) integrated Dense Depthwise Dilated Separable Spatial Pyramid Pooling into a DeepLabV3+ backbone, achieving state-of-the-art results in road continuity and precision on satellite datasets [2]. Zhou et al. (2025) extended this direction by designing a stepped parallel encoder that merged a CNN Encoder Module with a Transformer Encoder Module, allowing the model to simultaneously capture local and global dependencies [3].

Survey-based contributions have also advanced the field. Liu et al. (2024) reviewed deep learning approaches for road extraction, categorizing methods into supervised, semi-supervised, and unsupervised frameworks while noting the underexplored potential of lightweight IoT-friendly architectures [4].

Similarly, Mo et al. (2024) analyzed trends in CNN, GAN, and transformer-based networks, emphasizing the challenges of road connectivity under vegetation and shadow conditions [5].

Focused on contour refinement, Chen et al. (2021) proposed a deep feature-review transmit network that preserved thin road structures by enhancing contour information across layers, resulting in fewer segmentation gaps [6]. Hu et al. (2023) addressed spatial priors by introducing LGNet, a location-guided network that improved accuracy in dense urban environments but remained limited by its reliance on precise geolocation data [7]. Finally, Dai et al. (2023) introduced RADANet, which applied deformable attention modules to adaptively capture variable road widths and scales, improving segmentation quality though at the expense of computational efficiency [8]

3. *Methodology*

The methodology of this project describes the systematic approach adopted to design and implement an IoT-enabled Android application that extracts road features, detects road surface changes (potholes, cracks, and wear), and classifies road quality. The complete workflow is divided into the following stages:

3.1 System Design and Hardware Integration

The hardware and sensing architecture is developed to capture road images and transfer them to the cloud for analysis. The essential hardware components include:

- Microcontroller: NodeMCU ESP8266 serves as the communication module to connect ESP32-CAM with Firebase cloud.
- ESP32-CAM Module: Captures high-resolution images of roads at regular intervals or on-demand.
- Power Supply: A rechargeable Li-ion battery provides stable power for the ESP32-CAM and NodeMCU modules.
- Connectivity: The ESP32-CAM is configured with Wi-Fi support to upload images to Firebase in near real time.

This IoT-enabled hardware system ensures low-cost image acquisition for subsequent AI-based processing.

3.2 Data Acquisition and Cloud Storage

Road images are collected from two sources:

- ISRO Satellite Images (macro-level) for wide-area road extraction.
- ESP32-CAM Captured Images (micro-level) for ground condition assessment.

All images are uploaded and stored in Firebase Realtime Database and Cloud Storage, ensuring scalability and easy access for model training and analysis. Metadata such as timestamp, GPS coordinates, and image source are logged with each entry.

3.3 Road Feature Extraction Using CNN

A Convolutional Neural Network (CNN) is trained using Python on labeled ISRO satellite images to detect and extract road features. The network architecture follows a U-Net–based encoder–decoder structure for semantic segmentation. Once trained, the model is converted into TensorFlow Lite (TFLite) format for deployment on Android. The CNN provides a road mask highlighting road boundaries and segments.

3.4 Road Quality and Change Detection

To assess road quality:

- Image Pre-processing: Uploaded ESP32-CAM images are normalized and enhanced for clarity.
- Feature Analysis: The CNN classifies road conditions into categories — Good, Moderate, or Poor.
- Change Detection: By comparing current images with historical data (satellite and ground-level), changes such as cracks, potholes, or surface deterioration are identified.

This stage integrates temporal analysis for detecting newly emerged or worsened road damages.

3.5 Decision Tree Integration for Hybrid Analysis

To improve accuracy beyond visual classification, contextual data is collected via a questionnaire in the Android app, including:

- Road usage duration (years in service).
- Average traffic load.
- Location type (urban, rural, highway).

A Decision Tree algorithm processes these metadata inputs and fuses them with CNN predictions. This hybrid AI model ensures a more reliable classification of road conditions by combining sensor data and user knowledge.

3.6 Software and Programming

- Model Training: Python with TensorFlow/Keras is used to train the CNN.
- Conversion: Trained CNN models are optimized and exported into TFLite format for Android deployment.
- App Development: Android Studio (Java/Kotlin) is used to build the mobile application with Firebase integration.
- Database Handling: Firebase Realtime Database manages metadata and user input, while Cloud Storage handles road images.

3.7 Overall Workflow

1. ESP32-CAM captures road images and uploads them to Firebase.
2. CNN (TFLite) embedded in the Android app performs road segmentation and condition classification.
3. User questionnaire inputs are combined with CNN outputs via a Decision Tree.
4. A road condition report is generated, containing extracted features, quality scores, and maintenance alerts.
5. Reports are displayed in the mobile app and stored in Firebase for future monitoring and decision-making.

4. Module Description

The proposed system is composed of hardware and software modules that work in coordination to capture, transmit, process, and analyze road images for quality assessment and maintenance alerts.

4.1 ESP32-CAM Module

The ESP32-CAM is responsible for capturing high-resolution on-road images. It has an inbuilt Wi-Fi module and camera interface that allow image acquisition and direct transmission to the NodeMCU or Firebase. It ensures real-time monitoring and supports low-cost field deployment for IoT-based applications.

2.2 NodeMCU Wi-Fi Module (ESP8266)

The NodeMCU serves as a communication bridge between the ESP32-CAM and Firebase Cloud. It manages Wi-Fi connectivity, uploads captured images, and handles metadata such as timestamps and GPS coordinates. Its compact design and low power consumption make it suitable for continuous IoT operation.

4.3 GPS Module

The GPS module provides real-time geolocation data (latitude, longitude, and altitude). This information is linked with each road image and metadata stored in Firebase, enabling accurate mapping of road conditions and location-based maintenance planning.

4.4 Ultrasonic Sensor

The ultrasonic sensor is used for detecting obstacles or anomalies during ground-based data collection. While primarily deployed in IoT-based monitoring units, it can also assist in refining dataset labeling for ground-truth validation.

4.5 Power Supply (12V Battery)

The system is powered by a rechargeable 12V battery pack, which supplies stable power to the NodeMCU, ESP32-CAM, and peripheral modules. Voltage regulators are employed to step down voltages for components requiring 3.3V or 5V, ensuring efficient energy management in field conditions.

4.6 Cloud Layer (Firebase)

Firebase Cloud Storage is used for storing captured road images, while Firebase Realtime Database manages metadata and user questionnaire responses. The cloud provides scalable storage, synchronization, and secure data handling for multiple users and devices.

4.7 CNN Model (TensorFlow Lite)

A Convolutional Neural Network (CNN), trained in Python using TensorFlow/Keras, performs road feature extraction and quality classification. The trained model is converted into TFLite format for deployment on Android devices. It classifies roads into categories such as Good, Moderate, or Poor by analyzing both satellite (macro-level) and ESP32-CAM images (micro-level).

4.8 Decision Tree Module

To improve classification reliability, a Decision Tree algorithm is integrated into the Android application. It processes contextual user data such as traffic load, road usage duration, and location type, and fuses this with CNN outputs. This hybrid analysis produces more accurate quality assessments.

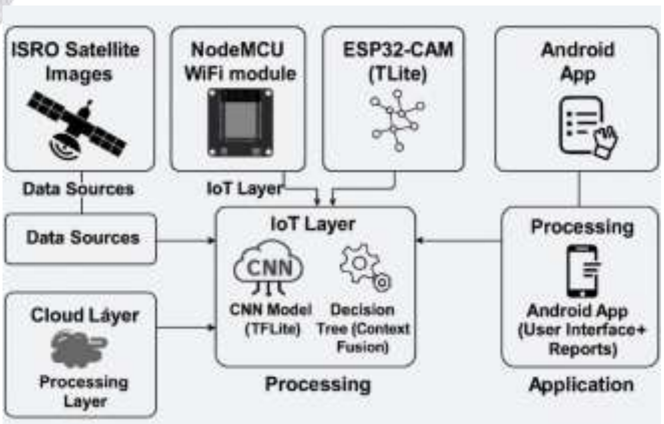
4.9 Android Application

The Android app provides the user interface for data collection, analysis, and visualization. Its main functions include:

- Uploading road images and questionnaire responses.
- Running the embedded TFLite CNN model for inference.
- Integrating Decision Tree results for final classification.
- Displaying road condition reports with extracted features, quality scores, and alerts.
- Offering history tracking for long-term monitoring.

4.10 Maintenance Report & Alerts Module

The final output of the system is a structured road condition report. It contains extracted road features, CNN-based classification, and Decision Tree–enhanced results. The Android app generates alerts for poor-quality roads, enabling authorities to prioritize maintenance tasks effectively.



5. Flowchart for Operation

6. Working

The proposed system integrates IoT hardware, cloud storage, and AI models within an Android application to automate road extraction and quality detection. The working process can be explained step by step as follows:

1. **Image Capture**
The ESP32-CAM module captures real-time road images at specific intervals or on user request. Each image is stamped with metadata such as timestamp and GPS location for precise mapping.

2. **Data Transmission**
Using the NodeMCU ESP8266 as a Wi-Fi bridge, the captured road images and metadata are uploaded to Firebase Cloud Storage and Realtime Database. This ensures that both the raw images and contextual data are available centrally for further processing.

3. **Cloud Storage & Synchronization**
Firebase stores the road images while simultaneously updating metadata entries (location, time, traffic details) in the Realtime Database. The Android application synchronizes with Firebase to fetch the latest data whenever a new image is uploaded.

4. **Road Extraction & Quality Detection**
Inside the Android app, a Convolutional Neural Network (CNN) model trained offline using ISRO satellite imagery and converted into TensorFlow Lite (TFLite) performs road segmentation. The CNN identifies road areas and classifies their condition as **Good, Moderate, or Poor**.

5. **User Contextual Input**
To improve prediction accuracy, the Android app presents a questionnaire to users. Inputs such as road usage duration, average traffic load, and location type are collected and sent to the system.

6. **Decision Tree Fusion**
The contextual user inputs are processed using a Decision Tree classifier, which fuses them with CNN outputs. This hybrid decision-making ensures that the final classification considers both visual features and real-world usage data.

7. **Report Generation**
The system generates a **Road Condition Report** containing:
    - Extracted road features (from CNN).
    - Quality classification (Good/Moderate/Poor).
    - Maintenance alerts (if condition is Poor).
    - Historical comparison for change detection (Δ quality over time).

.

7. Results and Discussion

The The proposed IoT-enabled road monitoring system successfully integrated hardware, cloud services, and AI models to achieve automated road extraction and condition detection. The CNN model trained on ISRO satellite imagery demonstrated reliable segmentation of road features, while the TensorFlow Lite deployment allowed smooth execution on the Android application with minimal latency. The system achieved effective classification of road quality into *Good*, *Moderate*, and *Poor* categories. In addition, the ESP32-CAM provided consistent real-time image capture, and Firebase enabled secure storage and synchronization between IoT devices and the mobile application.

The inclusion of a Decision Tree classifier significantly improved prediction accuracy by incorporating contextual information such as traffic load, usage duration, and location type. This hybrid approach reduced false positives in road quality detection and enhanced the reliability of maintenance alerts. The generated road condition reports provided a comprehensive view, combining extracted features, classification scores, and maintenance recommendations. Overall, the results validate that the system can serve as a cost-effective and scalable solution for road infrastructure monitoring, supporting smart city and government initiatives for timely maintenance and improved public safety.

8. Conclusion and Scope for Future Work

The The developed IoT-enabled road extraction and condition detection system demonstrates the feasibility of combining satellite imagery, on-road camera inputs, and artificial intelligence for smart infrastructure monitoring. By employing a CNN model for feature extraction and classification, integrated with a Decision Tree for contextual analysis, the system provides reliable road condition reports with actionable maintenance alerts. The use of Firebase ensures real-time data storage and synchronization, while the Android application delivers an accessible and user-friendly platform for end users. Overall, the project highlights a scalable, low-cost solution for improving the efficiency of road inspection and supporting smart city initiatives.

In the future, the system can be enhanced by incorporating more advanced deep learning architectures such as transformer-based models for higher accuracy and robustness across diverse environments. Integration with IoT sensor networks (vibration, pressure, and traffic counters) could further enrich the dataset and strengthen decision-making. Additionally, deployment on edge devices can reduce reliance on cloud processing and improve response times for real-time applications. Expanding the dataset with multi-source imagery, including drone and street-level inputs, will also broaden system applicability. These improvements will enable the proposed solution to evolve into a comprehensive, intelligent road infrastructure monitoring platform.

References

[1] Y. Wang, Y. Peng, X. Liu, W. Li, G. C. Alexandropoulos, J. Yu, D. Ge, and W. Xiang, "DDU- Net: Dual-Decoder-U-Net for Road Extraction Using High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 14, no. 3, pp. 1–16, Jan. 2022.

[2] A. Mahara, M. R. K. Khan, N. D. Rishe, W. Wang, and S. M. Sadjadi, "Automated Road Extraction from Satellite Imagery Integrating Dense Depthwise Dilated Separable Spatial Pyramid Pooling with DeepLabV3+," *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 1112– 1116, Oct. 2024.

[3] Z. Zhou, W. Chen, Q. Gui, X. Li, and L. Wang, "Road Extraction from Remote Sensing Images Using a Stepped Parallel CNN + Transformer Encoder," *Applied Sciences*, vol. 15, no. 3, pp. 1–14, Mar. 2025.

[4] R. Liu, J. Wu, W. Lu, Q. Miao, H. Zhang, X. Liu, Z. Lu, and L. Li, "A Review of Deep Learning-Based Methods for Road Extraction from High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 16, no. 12, pp. 1–25, Jun. 2024.

[5] S. Mo, Y. Shi, Q. Yuan, and M. Li, "A Survey of Deep Learning Road Extraction Algorithms Using High-Resolution Remote Sensing Images," *Sensors*, vol. 24, no. 5, pp. 1708–1726, Mar. 2024.

[6] D. Chen, H. Zhao, J. Wang, D. Wang, and S. Qing, "Deep Feature-Review Transmit Network of Contour-Enhanced Road Extraction From Remote Sensing Images," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 9, pp. 1547–1551, Sept. 2021.

[7] J. Hu, J. Gao, Y. Yuan, J. Chanussot, and Q. Wang, "LGNet: Location-Guided Network for Road Extraction From Satellite Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023.

[8] Z. Dai, G. Zhang, and R. Zhang, "RADANet: Road Augmented Deformable Attention Network for Road Extraction From Complex High-Resolution Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–13, 2023.

[9] A. Khelifi and M. Mignotte, "Deep Learning for Change Detection in Remote Sensing Images: A Comprehensive Review and Meta-Analysis," *IEEE Access*, vol. 8, pp. 181544– 181577, Sept. 2020.

[10] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully Convolutional Siamese Networks for Change Detection," *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pp. 4063–4067, Oct. 2018. (included for CNN change detection background).

[11] X. Peng, W. Sun, and L. Wu, "Multispectral Change Detection With Bilinear Convolutional Neural Networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 11, pp. 1893–1897, Nov. 2020.

[12] M. Gong, J. Zhao, and J. Liu, "Change Detection in Synthetic Aperture Radar Images Based on Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1184–1197, Apr. 2020.

[13] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning Spectral- Spatial-Temporal Features via a Recurrent CNN for Change Detection in Multispectral Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924–935, Feb. 2019. (closely related to temporal change detection).

[14] A. Ghosh, R. Sharma, and S. Joshi, "Decision Tree and Ensemble Classifiers for Remote Sensing Data Fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 2, pp. 354–366, Feb. 2002. (classic baseline for decision tree fusion).

[15] H. He, Y. Shi, Q. Li, and H. Yan, "Road Extraction by Using ATRous Spatial Pyramid Pooling With Encoder-Decoder Network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 12, pp. 8450–8462, Dec. 2020.