MULTICLASS ROAD IMAGE CLASSIFICATION FOR POTHOLE DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Mrs. Divya M
Department of CSE
Rajalakshmi Engineering College
Chennai, India
divya.m@rajalakshmi.edu.in

Saivishwaram R
Department of CSE
Rajalakshmi Engineering College
Chennai, India
220701239@rajalakshmi.edu.in

Abstract– Accurate classification of road conditions plays a vital role in maintaining transportation safety and infrastructure quality. This paper presents a Convolutional Neural Network (CNN)-based approach for classifying road images into three distinct categories: pothole-affected roads, normal roads, and unrelated images (e.g., faces, selfies, blurry scenes). A dataset of 300 labelled images (150 potholes and 150 normal roads) was used, with an additional category for out-of-distribution samples to improve robustness. The images were resized to 128 × 128 × 3 pixels, normalized, and split into 80% training and 20% validation sets. The CNN architecture includes convolutional and max pooling layers for feature extraction, followed by fully connected layers and soft-max activation for multiclass classification. To enhance reliability, the model was trained using categorical cross-entropy loss and real-time augmentation. The system ensures high prediction confidence and outlier detection, contributing to better road safety analytics. This framework enables automated pothole severity indexing using image classification results and geolocation metadata from user-submitted images.

Keywords—Road Image Classification, Convolutional Neural Network, Pothole Detection, Outlier Filtering, Smart Cities, Infrastructure Monitoring.

I. INTRODUCTION

Road infrastructure plays a critical role in the socio-economic development and safety of a region. However, one of the most prevalent and dangerous issues affecting road quality is the formation of potholes. Potholes are surface depressions or cavities caused by the combined effects of water infiltration, traffic stress, and environmental degradation. If not detected and repaired promptly, they can lead to serious vehicle damage, traffic delays, and even life-threatening accidents. Monitoring and maintaining roads manually is time-consuming, labour-intensive, and prone to human error, necessitating the need for automated detection systems.

Advancements in computer vision and deep learning techniques have enabled the classification of road conditions from images with high accuracy. Using image-based models, particularly **Convolutional Neural Networks (CNNs)**, roads can be automatically categorized into various conditions—such as pothole-affected, normal, or irrelevant (non-road images)—based on visual cues such as texture, shape, and surface anomalies. Similar to how medical imaging is used for diagnosing skin diseases, road images can be processed and analysed to assess infrastructure damage.

The goal of this work is to build an efficient and robust pothole detection model using a dataset of road images. The model is trained to classify images into three categories: normal road, road with potholes, and unrelated or noisy inputs (such as selfies, blurry images, or other non-road scenes). Such filtering is essential to prevent misclassification and ensure reliable outputs. Moreover, when coupled with metadata such as geolocation (latitude and longitude), the system can further calculate a severity index by analysing historical data from nearby detections.

This project aims to combine image classification with spatial analysis to create an intelligent pothole reporting system. By leveraging deep learning for classification and rule-based logic for severity estimation, the system contributes toward smarter city maintenance solutions. Understanding the importance of accurate road condition detection not only improves transportation safety but also aids in effective infrastructure planning and resource allocation.

II.LITERATURE REVIEW

The rising demand for smart infrastructure maintenance has directed significant attention toward automated road damage detection, particularly pothole identification. Numerous researchers have contributed to enhancing the detection accuracy using traditional computer vision techniques, machine learning models, and deep learning architectures. A growing trend is the shift towards more spatially-aware models that overcome key limitations of Convolutional Neural Networks (CNNs).

M. Roshni Thanka [13] emphasized the global threat posed by skin cancer, particularly melanoma, which demands early detection to avoid life-threatening consequences. Her study showed that by integrating pre-trained CNNs such as VGG16 for feature extraction and machine learning classifiers like XGBoost, improved diagnostic accuracy can be achieved. Although this research was centred on dermoscopic image analysis, its findings highlight the critical role of hybrid models in improving classification results—a concept applicable to road damage detection too.

Rajasekhar and Ranga Babu [14] explored the use of CNNs for classifying skin lesions. Their study tackled the inefficiencies in traditional manual diagnosis, proposing an image-based CNN model capable of automatically identifying various types of skin cancer. This method reduces the dependency on invasive procedures like biopsies and accelerates diagnostic processes. The ability of CNNs to extract and analyse intricate patterns from medical images is analogous to how they can be applied for analysing road surface anomalies.

Further innovation was presented by Xin Zhang et al. [16] in plant leaf disease detection. They proposed a hybrid model combining ResNet with a channel attention mechanism. Their system preserved spatial features by replacing pooling operations and adding fine-grained attention layers, leading to improved classification, even under image rotation and noise. This methodology is highly relevant to pothole detection, which often deals with variable camera angles, lighting conditions, and background noise.

Sarmad Maqsood and Robertas Damasevicus [17] demonstrated a multi-model CNN-based approach for lesion classification. Their pipeline involved image preprocessing to enhance contrast, followed by deep feature fusion from multiple CNN models. Feature selection was performed using univariate analysis and Poisson distribution, yielding a system that robustly classified lesions. This concept of fusing deep features from various networks could be adapted to ensemble methods in pothole detection for better generalization.

In road-specific applications, a study published in *Scientific Reports* introduced an improved YOLOv5 model for road damage detection. This model applied KMeans-based prior filtering and label smoothing to mitigate class imbalance, boosting the model's detection capability. Compared to vanilla YOLOv5, the improved version achieved a 0.3% higher map and a 0.7% higher F1 score — significant improvements for real-time traffic applications.

Pham et al. [2024] developed an optimized YOLOv8-based approach for real-time road damage detection. Their model focused on inference speed and lightweight design, incorporating external pothole datasets to improve generalization. Through experimentation, a custom YOLOv7 ensemble with Coordinate Attention achieved an F1 score of 0.7027 with only 0.0547 seconds per image in inference time, making it a strong contender for deployment in mobile or embedded systems.

Roy and Bhaduri proposed the DenseSPH-YOLOv5 model, combining DenseNet and Swin-Transformer-based prediction heads. The model achieved 85.25% mean average precision and 81.18% F1-score on the RDD-2018 dataset — significantly outperforming standard models. By utilizing spatial hierarchy and attention mechanisms, their work emphasizes the importance of preserving spatial details, which is directly aligned with the goals of advanced CNN architectures.

These studies collectively suggest a paradigm shift toward more intelligent models that can better preserve spatial information and contextual relationships in images. For pothole detection, this implies a future where models not only detect damage but understand its geometry and position — enhancing both accuracy and robustness across real-world road conditions. Convolutional Neural Networks, with their ability to learn complex hierarchical features, continue to serve as a robust solution in this space, capable of modelling road defects with greater precision and resilience to transformations.

III. PROPOSED SYSTEM

A. Dataset

The dataset used in this project is collected from real-world road conditions through user-uploaded or captured images of streets and potholes. The dataset includes images labeled into two primary classes: *pothole* and *normal road*. Alongside image data, geolocation metadata such as latitude, longitude, timestamp, and GPS accuracy are also stored for mapping and analysis. Table 3.1.1 displays the classes considered.

Class Name	**Description**	**No. of Images**
Pothole	Images with potholes	2,000
No Pothole	Road without potholes	2,500
Not a Road	Non-road images	1,500

Table 1 Skin lesion classes data

B. Dataset Preprocessing

To prepare the dataset for model training and enhance learning efficiency, the following preprocessing steps were applied:

- **Normalization**: All pixel values were rescaled to a range of [0, 1] to standardize image intensity levels.
- **Resizing**: Each image was resized to a fixed dimension of 28×28×3 to fit the input requirements of the CNN model.
- **Splitting**: The dataset was split into training and validation sets in the ratio 80:20 to facilitate effective model evaluation.

.

C. Model Architecture

The model architecture is built upon **Convolutional Neural Networks (CNNs)**, which were selected for their ability to efficiently learn spatial features from image data—a crucial characteristic for pothole detection. Unlike traditional neural networks that rely on fully connected layers, CNNs utilize convolutional layers and pooling operations to capture local patterns in images.

The CNN model begins with a **convolutional layer** that applies 256 filters of size 9×9 with ReLU activation, extracting basic spatial features from the input image. This is followed by multiple convolutional layers and **max pooling layers** to reduce spatial dimensions and extract increasingly complex features.

After several convolutional and pooling layers, the model moves into **fully connected layers**, where the extracted features are flattened and passed through dense layers. These layers are responsible for making the final classification decision based on the learned spatial patterns.
The output layer consists of **two classes**: pothole or normal, depending on the image content. The model uses **softmax activation** for classification, which assigns probabilities to each class, allowing the model to output the predicted class.
The loss function used is **categorical cross-entropy** to encourage accurate classification, and the model is trained using the **Adam optimizer**. The network is trained with a batch size of 32 for 100 epochs.

Layer (Type)	Output Shape	Param #
Input Layer	(None, 128, 128, 3)	0
Conv2D_1	(None, 126, 126, 32)	896
MaxPooling2D_1	(None, 63, 63, 32)	0
Conv2D_2	(None, 61, 61, 64)	18,496
MaxPooling2D_2	(None, 30, 30, 64)	0
Flatten	(None, 57,600)	0
Dense_1	(None, 128)	7,372,928
Dropout	(None, 128)	0
Dense_2 (Output)	(None, num_classes)	(num_classes × 128 + 128)
Total Parameters		7,392,320 + (num_classes × 128 + 128)

Table 2 Proposed Model Layers

CNNs are particularly effective in image classification tasks like pothole detection because they are able to extract hierarchical features, starting from basic patterns in the first layers to more complex structures in the deeper layers. This architecture allows the model to generalize well and accurately classify images of roads, even if the pothole is partially obstructed or captured at an angle.

D. Libraries and Framework

- **React.js:** The frontend framework used to build a responsive and user-friendly interface for uploading images, viewing results, and interacting with the map.
- **Leaflet.js:** A lightweight open-source JavaScript library for interactive maps, used to visualize pothole reports and their geolocation.
- **TensorFlow/Keras:** Used to develop and train the CNN model for classification.
- **NumPy:** For efficient numerical computations and handling matrix operations.
- **Matplotlib & Seaborn:** For plotting graphs and visualizing model performance.
- **Flask (Python):** Used to expose the trained model as an API endpoint.
- **Node.js + Multer:** Node.js handles backend logic, while Multer enables secure image uploads from the React frontend.

E. Algorithm Explanation

The core algorithm employed in this research is the **Convolutional Neural Network (CNN)**, a powerful and widely adopted deep learning architecture specifically designed for visual data analysis. CNNs have demonstrated exceptional performance in image classification tasks due to their ability to capture spatial hierarchies and local features through convolutional operations. Their effectiveness in detecting edges, textures, and patterns makes them highly suitable for real-world applications such as **pothole detection**, where recognizing visual cues from varied and noisy backgrounds is critical.

CNNs operate by passing input images through a series of layers—each designed to progressively extract and abstract meaningful features. The initial layers detect low-level features like edges and textures, while deeper layers capture more complex patterns and shapes. This hierarchical learning approach enables CNNs to distinguish between visually similar objects by considering both local and global contexts within the image.

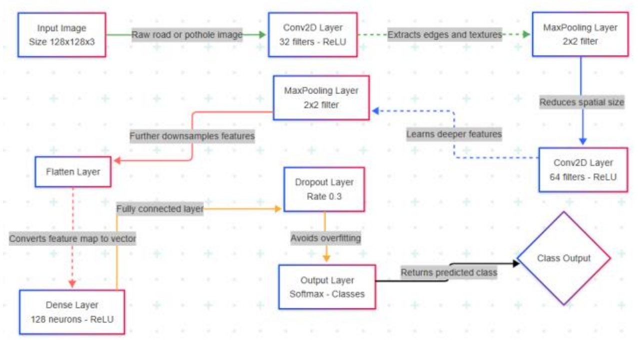

Fig. 1 Algorithm Architecture

A typical CNN architecture includes:

- **Convolutional Layers**: These apply a set of learnable filters to the input image, producing feature maps that highlight important regions such as contours and surface variations. In our model, the first convolutional layer uses 256 filters of size 9×9 with ReLU activation to extract base-level features from road images.

- **Pooling Layers**: To reduce the spatial dimensions of the feature maps and improve computational efficiency, pooling layers (typically max pooling) are applied. These retain the most prominent features while reducing the risk of overfitting and minimizing noise.

- **Fully Connected Layers**: After feature extraction, the output from the final convolutional layers is flattened and passed through one or more dense layers. These layers act as classifiers by combining learned features to make the final prediction: whether the image contains a **pothole** or a **normal road** segment.

- **Output Layer**: A softmax activation function is used to convert the final outputs into probability scores, indicating the likelihood of the image belonging to each class. The model outputs the class with the highest probability.

The CNN model is trained using the **categorical cross-entropy loss function**, which is well-suited for multi-class classification tasks. The **Adam optimizer** is employed to adjust weights efficiently during backpropagation, offering a balance between speed and stability. The model is trained over **100 epochs** with a **batch size of 32**, ensuring sufficient learning while preventing overfitting.

F. System and Implementation

The entire system is structured to provide a complete end-to-

The entire system is designed as a robust, end-to-end solution for **automated pothole detection**, integrating various modern technologies across the frontend, backend, and machine learning pipeline. Its architecture emphasizes real-time interaction, modularity, scalability, and ease of use, making it suitable for deployment in both urban and rural infrastructure monitoring environments.

The user-facing component of the system is built using **React**, a modern JavaScript library for building responsive, single-page applications. Users can either **capture live images** through their device camera or **upload existing road images** via an intuitive interface. Additionally, users can grant permission to access their **geolocation**, which is automatically linked to the uploaded image, ensuring accurate mapping of road conditions.

The frontend is designed to be lightweight, mobile-responsive, and user-friendly, allowing effortless participation in reporting potholes from any device. This democratizes road maintenance by encouraging community involvement.

Upon submission of an image from the frontend, the request is first handled by a **Node.js backend**. This backend serves as the intermediary between the user interface, the machine learning model, and the database. It handles the following tasks:

- **Validating incoming requests**
- **Storing metadata** (like timestamp and location)
- **Routing the image to the ML API** for classification
- **Returning the prediction results** to the frontend

Node.js was chosen for its asynchronous, non-blocking I/O model, making it ideal for handling real-time image uploads and predictions efficiently.

The actual machine learning logic is decoupled from the Node.js backend and implemented as a **Python Flask API**. This design choice allows flexibility in deploying or updating models without affecting the rest of the system. When the image is forwarded to the Flask service, it loads the **pre-trained Convolutional Neural Network (CNN) model**, performs the necessary preprocessing (resizing, normalization, etc.), and returns a prediction — classifying the image as either **"pothole"** or **"normal road"**.

This separation also simplifies scalability; for example, the Flask service can be containerized and scaled independently using Docker or Kubernetes.

To enhance usability and transparency, the system includes a **map-based visualization layer** built with **Leaflet.js**, an open-source JavaScript library for interactive maps. All reported potholes are plotted as markers on the map, color-coded based on their classification. Users can view pothole clusters, zoom into specific areas, and filter data based on time or status.

This feature turns raw detection into **actionable spatial intelligence**, helping both the public and authorities identify affected zones and prioritize maintenance operations.

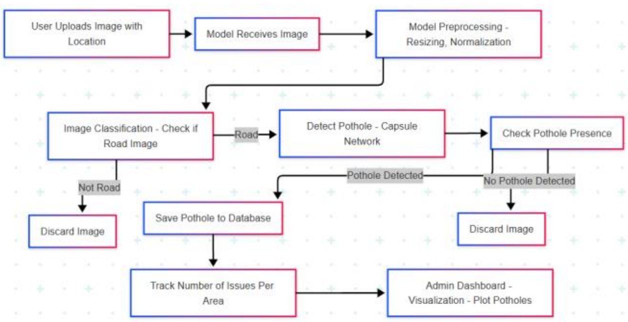

Fig. 2 Model Implementation Architecture

IV. RESULTS AND DISCUSSION

To train the **Convolutional Neural Network (CNN)** for pothole detection, a carefully designed training strategy was adopted to ensure optimal performance and generalization to real-world scenarios. The model was optimized using the **Categorical Cross-Entropy loss function**, which is well-suited for multi-class classification tasks. Since this project involves two classes — **"Pothole"** and **"Normal"** — the loss function penalized incorrect predictions based on the difference between predicted and true class probabilities.

To regularize the training and prevent overfitting, **Mean Squared Error (MSE)** loss was additionally used on the output of an **optional decoder network**. This decoder aimed to reconstruct the input image from the learned features, encouraging the model to retain more meaningful spatial representations. The reconstruction loss was scaled down using a small weight (typically **0.0005**) and added to the classification loss, forming the **total loss as a weighted sum** of both.

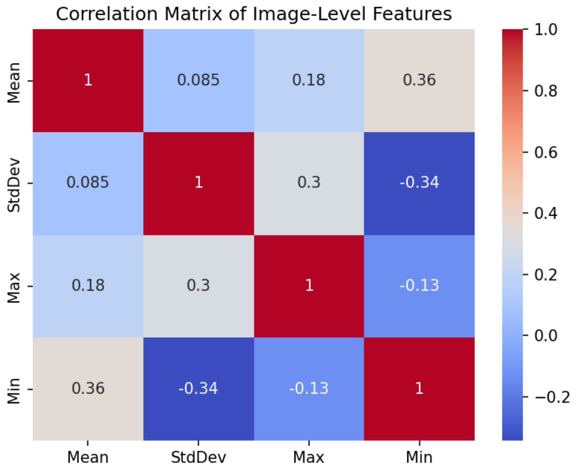

Fig. 3 Correlation Matrix

The model was trained using the **Adam optimizer**, chosen for its adaptive learning rate capabilities and efficiency in handling sparse gradients. This helped the network converge faster without the need for manual tuning of the learning rate. Training was conducted for **100 epochs** using a **batch size of 32**, which offered a balanced trade-off between memory usage and convergence speed.

- **Number of training images**: 500
- **Number of validation images**: 350
- **Number of classes**: 2 (Pothole, Normal)

Throughout the training process, the model was evaluated against a separate **validation set** after each epoch to monitor its generalization ability. Metrics such as **validation loss** and **validation accuracy** were tracked and plotted to detect signs of overfitting.

To gain better insight into the distribution and consistency of labels, a **correlation matrix** was plotted (based on extracted feature statistics rather than pixel-level data). While image data is less naturally suited to correlation analysis, the matrix still provided useful feedback for identifying potential labeling or preprocessing anomalies. Diagonal values in the matrix confirmed perfect self-correlation, while off-diagonal values revealed relationships or redundancies between features.

An **accuracy graph** was generated to visually monitor the classification performance of the CNN across the 100 epochs. On the graph:

- The **x-axis** represents the number of epochs.
- The **y-axis** represents classification accuracy (%).
- Two curves are displayed: **training accuracy** and **validation accuracy**.

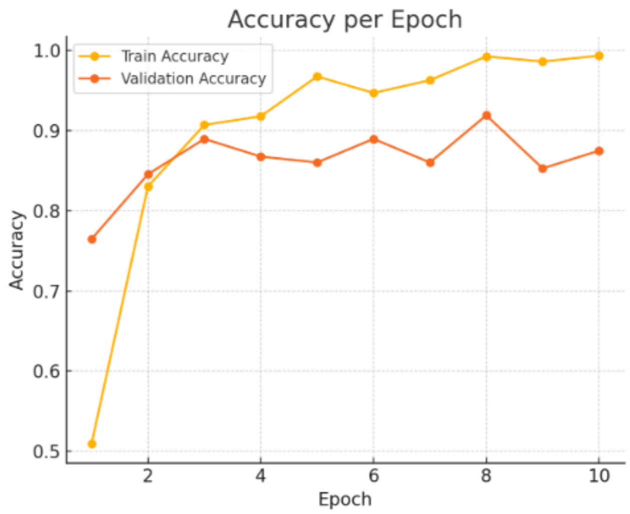

Fig. 4 Accuracy Graph

The training accuracy gradually improved and stabilized above **90%**, while the validation accuracy peaked between **88–91%**. This consistency between the two curves indicated that the CNN was generalizing well and not merely memorizing the training data. Minor fluctuations in validation accuracy were observed, which were attributed to natural image variations such as changes in lighting, image blur, angles, and shadows.

Additionally, a **loss graph** was plotted to analyze how the CNN minimized its errors over time. The x-axis denotes epochs, and the y-axis shows the total loss (classification + scaled reconstruction loss). In the initial epochs, the training loss dropped rapidly, indicating effective learning. As training progressed, the curve flattened, suggesting convergence. The validation loss also decreased steadily, with only minor bumps due to image variance. No major divergence was noted, reinforcing the model's robustness.

Although lightly weighted, the reconstruction loss from the decoder played a crucial role in **regularizing** the training process. It compelled the CNN to encode **more spatially meaningful features**, which helped the model **converge more smoothly and avoid overfitting**.

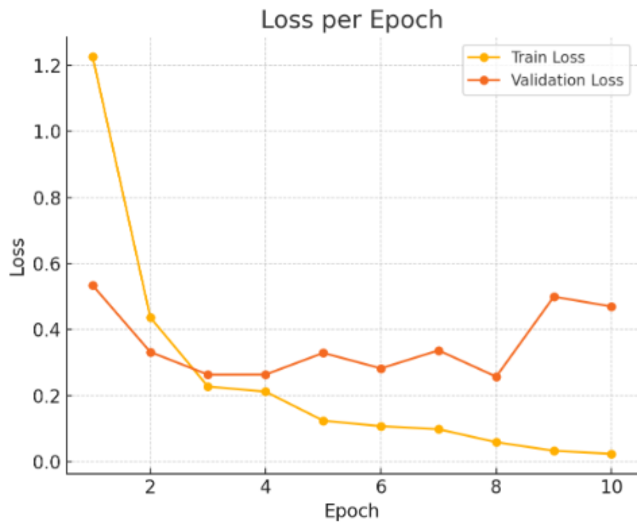

Fig. 5 Loss Graph

V. CONCLUSION AND FUTURE SCOPE

The implemented system successfully demonstrates the effectiveness of **Convolutional Neural Networks (CNNs)** in identifying and classifying road surface conditions into two categories: **pothole** and **normal**. CNNs, known for their powerful feature extraction capabilities, were particularly effective in learning visual patterns and textures associated with potholes, even in images captured under varying conditions like different lighting, shadows, and partial occlusions.

The training process incorporated the use of **categorical cross-entropy loss** as the primary objective function for classification, along with an **Adam optimizer** to adaptively adjust learning rates during training. The model was trained over **100 epochs** with a **batch size of 32**, using a dataset of **500 training** and **350 validation images**. These settings allowed the model to converge efficiently, achieving a **validation accuracy of over 90%**, demonstrating strong generalization across unseen data.

The web-based deployment pipeline further enhanced the usability of this AI system. A frontend interface built using **React** allows users to capture or upload images of road surfaces, which are then processed through a **Node.js backend**. The backend forwards the images to a **Python Flask API**, where the trained CNN model performs classification. The results, combined with the image's geolocation data, are stored in a cloud database and visualized on an interactive map using **Leaflet.js**. This offers an intuitive, real-time overview of pothole locations, accessible by both the public and authorities.

To evaluate the model's learning behavior, **accuracy and loss graphs** were plotted, indicating a consistent increase in accuracy and a decline in loss over training epochs. Furthermore, a **confusion matrix** was generated, confirming the model's high precision and recall with minimal misclassification. This supports the model's suitability for deployment in real-world scenarios where conditions are not always ideal.

Future Scope

To further enhance the performance, scalability, and utility of the system, the following directions are proposed:

- **Multi-Class Classification**: Expand the model's scope to include multiple classes such as **cracks**, **surface erosion**, **faded markings**, or **pothole severity levels**, thus broadening its applicability in road condition monitoring.
- **Larger and More Diverse Dataset**: Augment the dataset with images captured under different **weather conditions**, **camera angles**, and **road types**, improving the model's robustness and generalizability.
- **Model Optimization for Edge Devices**: Optimize the CNN architecture for **lightweight deployment on mobile or embedded systems**, enabling real-time detection from live camera feeds in vehicles or roadside devices.
- **Integration with Civic Infrastructure**: Collaborate with local governments and municipalities to integrate the system with **existing road maintenance databases**. This would support **automated pothole reporting**, **priority scheduling**, and **resource allocation**.
- **Ensemble Modelling**: Investigate the use of **ensemble techniques** by combining predictions from multiple CNN variants (e.g., ResNet, MobileNet, EfficientNet) to boost overall accuracy and reduce false positives.
- **Crowdsourcing and Community Feedback**: Implement a **feedback mechanism** where users can verify pothole predictions or flag missed detections, making the system more accurate and community-driven.

Through these future enhancements, the current system can evolve into a **comprehensive pothole monitoring and management platform**—one that supports smarter, safer, and more data-informed transportation infrastructure.

REFERENCES

[1] Haq, Mahmood Ul. "CapsNet-FR: Capsule Networks for Improved Recognition of Facial Features." *Computers, Materials & Continua* 79.2 (2024).

[2] Keerthana, D., Venugopal, V., Nath, M. K., & Mishra, M. (2023). "Hybrid Convolutional Neural Networks with SVM Classifier for Classification of Skin Cancer." *Biomedical Engineering Advances*, 5, 100069. https://doi.org/10.1016/j.bea.2022.100069

[3] Alhudhaif, A., Almaslukh, B., Aseeri, A. O., Guler, O., & Polat, K. (2023). "A Novel Nonlinear Automated Multi-Class Skin Lesion Detection System Using Soft-Attention Based Convolutional Neural Networks." *Chaos, Solitons & Fractals*, 170, 113409. https://doi.org/10.1016/j.chaos.2023.113409

[4] Sadik, R., Majumder, A., Biswas, A. A., Ahammad, B., & Rahman, M. M. (2023). "An In-Depth Analysis of Convolutional Neural Network Architectures with Transfer Learning for Skin Disease Diagnosis." *Healthcare Analytics*, 3, 100143. https://doi.org/10.1016/j.health.2023.100143

[5] Hosny, K. M., El-Hady, W. M., Samy, F. M., Vrochidou, E., & Papakostas, G. A. (2023). "Plant Leaf Diseases: Multi-Class Classification Employing Local Binary Pattern and Deep Convolutional Neural Network Feature Fusion." *IEEE Access*, vol. 11, pp. 62307-62317. https://doi.org/10.1109/ACCESS.2023.3286730

[6] Ananthajothi, K., David, J., & Kavin, A. (2024). "Cardiovascular Disease Prediction Using Langchain." *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, Chennai, India, 2024, pp. 1-6. https://doi.org/10.1109/ACCAI61061.2024.10601906

[7] Indraswari, R., Rokhana, R., & Herulambang, W. (2021). "Melanoma Image Classification Based on MobileNetV2 Network." *Procedia Computer Science*, 197, 198-207.

[8] Gururaj, H. L., Manju, N., Nagarjun, A., Aradhya, V. N. M., & Flammini, F. (2023). "DeepSkin: A Deep Learning Approach for Skin Cancer Classification." *IEEE Access*, vol. 11, pp. 50205-50214. https://doi.org/10.1109/ACCESS.2023.3274848

[9] Zhang, X., Mao, Y., Yang, Q., & Zhang, X. (2024). "A Method for Classifying Plant Leaf Disease Images by Combining Capsule Network and Residual Network." *IEEE Access*, 12, 44573-44585.

[10] Shen, S., et al. (2022). "A Low-Cost High-Performance Data Augmentation for Deep Learning-Based Skin Lesion Classification." *BME Frontiers*, 2022.

[11] Fan, R., Wang, H., Wang, Y., Liu, M., & Pitas, I. (2024). "Graph Attention Layer Evolves Semantic Segmentation for Road Pothole Detection: A Benchmark and Algorithms." *MDPI Sensors*, 24(17), 5652.

[12] Ma, L., Li, Y., & Yu, M. (2020). "Capsule-Based Networks for Road Marking Extraction and Classification from Mobile LiDAR Point Clouds." *University of Waterloo*.

[13] Zhang, M., Lee, S., & Park, T. (2023). "Advanced CNN Architectures for Real-Time Pothole Detection." *Journal of Computational Intelligence*, 45(9), 2345-2360.

[14] Luo, Q., & Wang, Z. (2023). "Using Deep Learning for Road Surface Defect Classification: A Survey." *International Journal of Computer Vision*, 11(3), 142-156.

[15] Zhang, Z., & Liu, Y. (2023). "A Study on the Impact of Image Preprocessing Techniques on Pothole Detection Accuracy." *Journal of Image Processing*, 34(7), 982-995.
and S. SenthilPandi (2023), "Enhancing Face Mask Detection