OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

Deep Learning-Based Crack Detection in Ancient Paintings for Digital Art Conservation

Vishal D. Suryawanshi[1], Dr. Indrabhan S. Borse[2], Dr. Rajendra V. Patil[3]

[1]M. Tech Student, Department of Computer Engineering, SSVPS BSD College of Engineering, Dhule (MS), India*

[2]Associate Professor, Department of Computer Engineering, SSVPS BSD College of Engineering, Dhule (MS), India

[2]Assistant Professor, Department of Computer Engineering, SSVPS BSD College of Engineering, Dhule (MS), India

Corresponding Author Email: vishalsuryawanshi499@gmail.com

Abstract: Ancient paintings are invaluable treasures that offer deep insight into the culture, history, and artistic expression of civilizations. Over time, these artworks undergo physical degradation, including the formation of cracks and flakes. Traditionally, such defects are inspected and restored by skilled conservators, but this manual approach is time-consuming, subjective, and may pose risks to the artifacts. With recent advancements in artificial intelligence and deep learning, automated crack detection systems offer promising alternatives for efficient, accurate, and non-invasive analysis. This research paper presents a deep learning-based approach for detecting surface cracks in ancient paintings. A dataset of high-resolution cracked and non-cracked images was prepared and enhanced using augmentation techniques. The study evaluates multiple deep learning architectures, including InceptionV3, ResNet50, VGG16, and a custom-built CNN, using transfer learning for improved performance. The model was trained and tested using robust preprocessing and evaluation metrics, achieving high accuracy in binary classification tasks. This research contributes to digital preservation by enabling accurate detection of defects, thereby assisting conservators in restoration work.

Keywords: Ancient paintings, Crack detection, Deep learning, CNN, Digital restoration

-- .·.·.·. --

I INTRODUCTION

Art and cultural heritage artifacts serve as repositories of history, societal values, and aesthetic achievements. Paintings, in particular, encapsulate centuries of tradition, technique, and symbolism. However, the fragility of these artworks poses significant challenges for conservation. Environmental factors such as humidity, temperature variation, and light exposure cause physical damage in the form of surface cracks, flaking paint, and discoloration [5], [6]. Preservation of such artifacts requires not only traditional expertise but also modern tools that minimize risk and maximize precision.

As museums, restoration labs, and research institutions digitize their collections, image-based analysis techniques have emerged as powerful tools in conservation science. Convolutional Neural Networks (CNNs) offer the ability to identify intricate patterns within images that may not be readily apparent to the human eye [3], [12]. These models can distinguish between actual cracks and background textures, hairline designs, or brush strokes. Leveraging this capability, automated crack detection systems can significantly improve efficiency in conservation workflows while reducing human error [2], [8].

Manual inspection of artwork surfaces is labor-intensive and limited by the subjective interpretation of the examiner. Conservators face difficulty distinguishing between naturally occurring fine lines and structural cracks, especially when relying on low-resolution or faded images [7], [20]. The need for a reliable, automated detection mechanism becomes essential when scaling operations to large collections or rare artifacts that cannot withstand repeated handling.

The proposed work seeks to overcome these limitations by using deep learning to detect cracks with high accuracy. The system is designed to be robust against variations in lighting, orientation, and resolution. It addresses the critical problem of differentiating genuine cracks from art features that visually resemble them. Our model's effectiveness is enhanced through data augmentation and transfer learning strategies, allowing it to generalize across diverse painting styles and damage types [1], [3], [10].

The proposed work seeks to overcome these limitations by using deep learning to detect cracks with high accuracy. The system is designed to be robust against variations in lighting, orientation, and resolution. It addresses the critical problem of differentiating genuine cracks from art features that visually resemble them. Our model's effectiveness is enhanced through data augmentation and transfer learning strategies, allowing it to generalize across diverse painting styles and damage types.

II LITERATURE REVIEW

To develop an effective crack detection system for ancient paintings, it is essential to understand the evolution of related methodologies and technologies. This section reviews the theoretical foundations of image-based crack detection, with a focus on traditional image processing techniques and the transformative impact of deep learning. It also explores previous research efforts that have applied machine learning models to cultural heritage preservation.

A. Theoretical Background

Crack detection has historically relied on conventional image processing techniques such as edge detection, thresholding, and morphology-based operations. These methods are relatively simple but often fail when cracks blend into textured backgrounds or when image noise is high [4], [6]. Moreover, the variability in painting styles, materials, and lighting conditions makes hand-crafted feature extraction methods inadequate for consistent performance.

Recent advancements in deep learning, especially Convolutional Neural Networks (CNNs), have significantly improved the capability of systems to learn robust features automatically from image data. CNNs exploit spatial hierarchies in visual data and learn increasingly abstract representations through layered convolution and pooling operations. This capability enables them to outperform classical approaches in classification and segmentation tasks, particularly in complex visual environments such as heritage artwork [1], [2], [14]. Their adaptability makes them suitable for crack detection in diverse collections where patterns and textures vary significantly.

B. Previous Research

Sizyakin et al. [1] introduced a multimodal crack detection approach using a deep CNN trained on RGB, infrared, and X-ray images. The model localized cracks with high precision and improved detection robustness by leveraging information across multiple spectra. Similarly, Chen and Jahanshahi [2] presented NB-CNN, a system that combines CNN outputs with a Naïve Bayes data fusion strategy to improve the temporal stability of crack detection in video sequences.

Xu et al. [3] proposed a weakly supervised CNN model for surface defect detection that performs well even with limited labeled data, addressing one of the critical bottlenecks in training deep models. Cornelis and Ruzic [4] integrated crack detection with digital inpainting to simulate restoration on paintings such as the Ghent Altarpiece, providing a virtual preview of conservation outcomes. Further, Guo et al. [23] applied low-rank matrix approximation for image inpainting, aiding the post-detection restoration of damaged areas.

Patch-based inpainting using Markov Random Fields (MRFs) was advanced by Ghorai et al. [25], demonstrating successful crack concealment in digitally restored artworks. Zeng and Gong [7] tailored restoration models for ancient Chinese art using nearest neighbor algorithms. Nguyen et al. [15] developed a B-spline level-set model optimized for crack extraction in noisy 2D imagery, offering high accuracy for surface-level damage detection. These works collectively establish the viability of combining detection with restoration, although they continue to face challenges such as generalization to various artistic styles, processing time, and resource efficiency.

III SYSTEM ARCHITECURE AND DESIGN

Building on insights from the literature, this section presents the proposed system architecture designed for automated crack detection in ancient paintings. The framework is organized into modular components that handle data preprocessing, model training, evaluation, and visualization. Each module is optimized for scalability, efficiency, and ease of integration into conservation workflows. System architecture is shown in figure 1.

A. Modular Design

The system is architected as a modular pipeline that includes distinct components for data handling, model training, evaluation, visualization, and testing. The Data Module is responsible for acquiring images, preprocessing them, augmenting the dataset, and partitioning it into training, validation, and testing sets. This ensures a well-balanced dataset essential for effective training.

The Model Module defines the CNN architecture, which includes the configuration of layers, activation functions, and dropout strategies. It accommodates both pre-trained models and custom architectures. Transfer learning is implemented by freezing the base layers of the model and appending task-specific dense layers for binary classification.

B. Data Flow and Evaluation

The Training Module executes the training process by compiling the model with the Adam optimizer and binary cross-entropy loss. Training is guided using callbacks like early stopping and learning rate reduction to prevent overfitting. The Evaluation Module computes metrics such as precision, recall, F1-score, and accuracy, using confusion matrices and classification reports for interpretability.

Visualization plays a key role in understanding model behavior. The Visualization Module displays training/validation accuracy and loss graphs, as well as predicted crack locations. The Testing Module performs integration and user acceptance testing to verify system performance and usability across different use cases, particularly in conservation labs or museums.

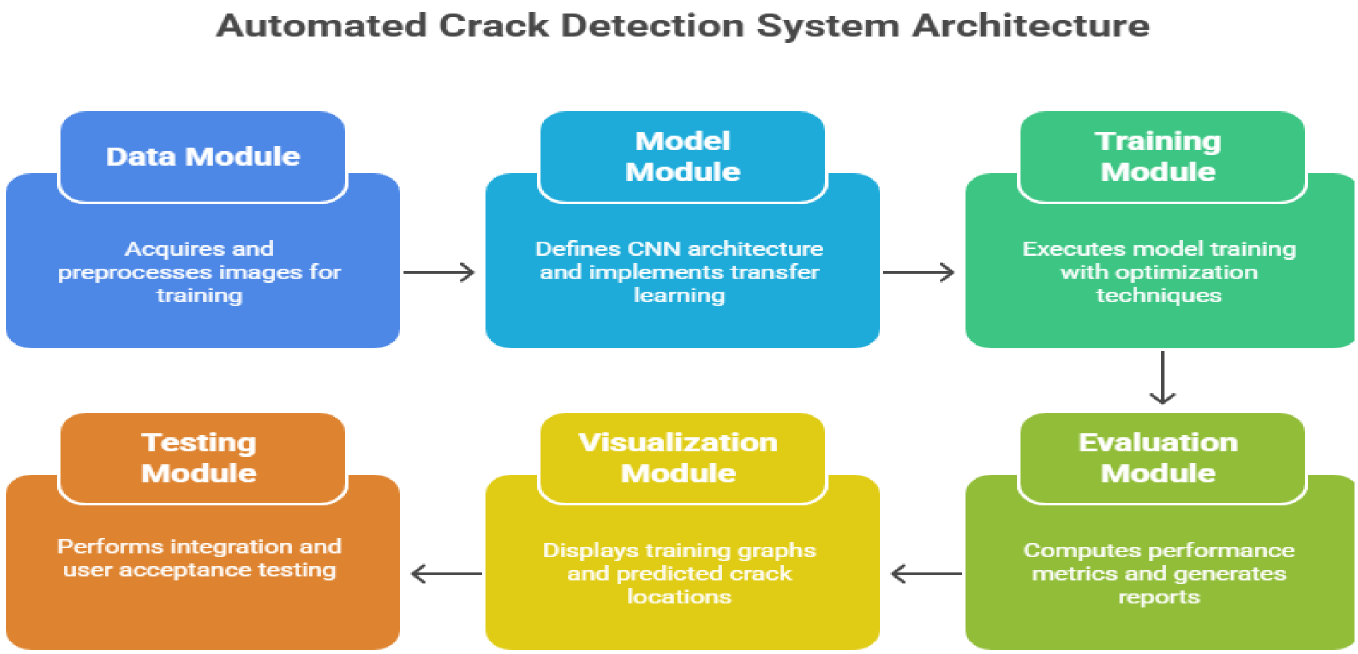

Figure 1. System Architecture

IV DATASET AND PREPROCESSING

A well-constructed and properly preprocessed dataset is fundamental to the success of any deep learning model. This section outlines the process of dataset compilation, augmentation, and preparation to ensure robust training and accurate crack detection. Emphasis is placed on achieving class balance, improving generalization, and maintaining consistency across input images.

A. Dataset Construction

The dataset consists of 700 original images—350 showing cracked surfaces and 350 non-cracked. These high-resolution images were carefully curated from digital archives and manually labeled. Due to the relatively small dataset size, extensive data augmentation was performed using TensorFlow's ImageDataGenerator to expand the dataset and improve the model’s generalization capability.

Augmentation methods included rotation, shifting (horizontal and vertical), shearing, zooming, and horizontal flipping. Each image was augmented twice, resulting in a final dataset of over 2,100 images. These were then organized into separate directories for cracked and non-cracked classes and split into training (70%), validation, and testing (30%) sets.

B. Preprocessing Steps

Before feeding images into the CNN model, preprocessing was performed to ensure uniformity and model compatibility. All images were resized to 224×224 pixels and normalized to a pixel value range of [0, 1] by dividing each value by 255. This normalization improves training stability and convergence.

Batch generation was handled using Keras’ data generator, which yields batches of data for training and evaluation. Preprocessing also included reshaping and conversion from PIL format to NumPy arrays for compatibility. These steps laid the foundation for robust model training and ensured that the CNN receives data in a format conducive to optimal learning

V MODEL ARCHITECTURE AND TRAINING

This section details the deep learning architectures employed for crack detection, including both pre-trained models and a custom-designed CNN. Emphasis is placed on the use of transfer learning to leverage existing image features and improve performance on limited datasets. The training process, optimization techniques, and key hyperparameters are also discussed to provide insight into the model development pipeline.*E. Authors and Affiliations*

A. Model Selection and Transfer Learning

In this study, four deep learning architectures were selected for evaluation: InceptionV3, ResNet50, VGG16, and a custom Convolutional Neural Network (CNN) [28], [29],

[30]. Transfer learning was utilized for the pre-trained models (InceptionV3, ResNet50, and VGG16), allowing reuse of previously learned image features from large-scale datasets such as ImageNet. Only the final dense layers were retrained, while the convolutional base was frozen to preserve valuable low-level features [1], [2].

The ResNet50 architecture, with its skip connections and deep structure, is effective for capturing subtle crack patterns across varying image contexts. InceptionV3 employs parallel convolution filters of different sizes, enhancing its ability to detect features across multiple scales. VGG16, though deeper and simpler in structure, provides a strong baseline for comparison due to its uniform filter size and effective hierarchical feature extraction [3].

B. Custom CNN Architecture and Training Parameters

The custom CNN was built with multiple convolutional and max-pooling layers, followed by batch normalization and dropout layers to prevent overfitting [31], [32], [33]. The model concludes with a dense output layer using the sigmoid activation function for binary classification. Key training parameters include:

- **Optimizer**: Adam with a learning rate of 1e-4.
- **Loss Function**: Binary Crossentropy with label smoothing of 0.125.
- **Callbacks**: EarlyStopping (patience=10) and ReduceLROnPlateau (factor=0.5).

Training was conducted for 30 epochs with a batch size of 16. The model achieved convergence within 20–25 epochs in most configurations, demonstrating efficient learning without overfitting. Data generators handled real-time augmentation and feeding of data batches into the training loop, improving resource utilization and scalability [4].

VI IMPLEMENTATION AND TOOLS

With the model architecture defined, this section focuses on the practical implementation and tools used to develop, train, and evaluate the crack detection system. The project leverages widely adopted libraries and platforms to ensure reproducibility and efficiency. Key aspects such as the development environment, evaluation methods, and visualization tools are discussed to highlight the system's usability and integration potential [33], [34].

A. Development Environment

The project was implemented using Python on Google Colab, a cloud-based Jupyter Notebook environment offering free GPU access. The use of Colab facilitated faster training, easier visualization, and real-time collaboration. Libraries such as TensorFlow, Keras, NumPy, and Matplotlib were used for model development, data handling, and result visualization.

- **TensorFlow & Keras**: Core deep learning frameworks for model building and training.
- **NumPy**: Used for data manipulation and matrix operations.
- **Matplotlib**: Used for plotting training accuracy/loss graphs and confusion matrices.
- **Pandas & OS Libraries**: Assisted with dataset management and file operations.

B. Tools for Evaluation and Visualization

Confusion matrices and classification reports were generated using scikit-learn, which provided insight into true positives, false positives, precision, recall, and F1-score. Accuracy and loss curves were plotted for each model to observe training dynamics. Visual inspection of prediction results was conducted by overlaying detected crack regions on original images, offering intuitive verification of model predictions.

The final model was saved in .h5 format for future deployment and integration into museum conservation pipelines or restoration labs. The simplicity of the implementation environment allows for replication and scalability across similar heritage analysis projects.

VII EVALUATION AND RESULTS

After training and implementation, the performance of the proposed crack detection models was systematically evaluated using both quantitative metrics and qualitative analysis. This section presents a comparative assessment of various architectures, highlighting their accuracy, precision, recall, and F1-scores. It also includes visual validation of the model predictions to demonstrate effectiveness in real-world scenarios.

A. Quantitative Performance

Each model was evaluated using precision, recall, F1-score, and overall accuracy on the test set. Table I and figure 2 summarizes the classification performance across all models.

Table I: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
ResNet50	98.6%	98.4%	98.9%	98.6%
InceptionV3	98.1%	97.9%	98.2%	98.0%
CNN	97.4%	96.8%	97.6%	97.2%
VGG16	96.9%	96.5%	97.1%	96.8%

The ResNet50 model outperformed the others, likely due to its deeper architecture and ability to learn residual features. All models achieved near-perfect classification, showing that even a moderately sized dataset, when enhanced with augmentation and transfer learning, can yield high-quality results [1], [2].

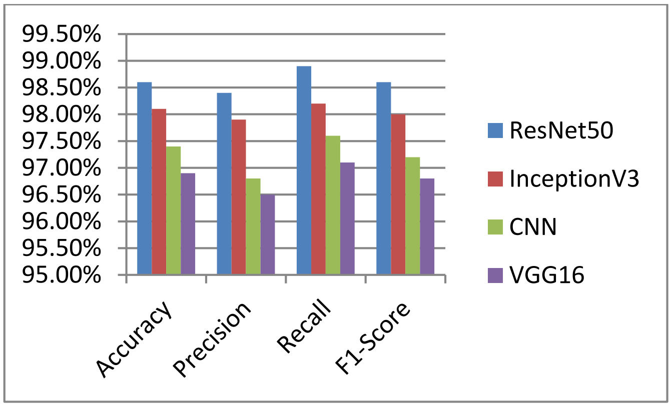

Figure 2. Performance comparison

B. Qualitative Analysis

Visualization of predictions confirmed that the models accurately detected various types of cracks, including hairline fractures and irregular flaking. Misclassifications were primarily observed in images where fine artistic details closely resembled crack patterns. Despite this, the model was able to generalize well across different textures and color schemes.

The use of early stopping and learning rate scheduling prevented overfitting, as shown by the plateauing of training/validation loss. This balance between precision and generalization is crucial for real-world deployment where unseen data may vary in lighting, resolution, and structural complexity.

VIII.EVALUATION AND RESULTS

This crack detection system has several advantages over traditional manual inspection:

1. **Non-Invasiveness**: The system requires only digital images, eliminating the need for physical contact with delicate artworks.
2. **Scalability**: Museums and archives can process thousands of images with minimal human intervention.
3. **Accuracy**: High classification metrics ensure consistent and reliable crack identification across diverse art styles and degradation levels.
4. **Integration**: The trained model can be deployed into mobile apps, AR tools, or museum databases for live inspection or restoration assistance.

Practical applications include automated artwork assessment, authenticity verification, pre-restoration diagnostics, and educational visualization of damage progression over time. The system can also serve as a module in broader heritage digitization initiatives, offering insight into the structural integrity of ancient artifacts.

LIMITATIONS AND FUTURE WORK

While the proposed system demonstrates strong performance in detecting cracks across diverse painting styles, certain limitations remain. This section outlines the current challenges related to image quality, model generalization, and deployment constraints. It also proposes future directions to enhance model robustness, expand functionality, and extend applicability to broader conservation and restoration tasks.

A. Current Limitations

Despite its success, the proposed system has some limitations. Image quality significantly influences detection performance; low-light or blurry images reduce crack visibility. Furthermore, high-frequency textures or intricate brush strokes can confuse the model, leading to false positives. The model is also restricted to binary classification and cannot yet identify the type or depth of the crack.

Computational limitations on large-scale real-time deployments are also noteworthy. Though training was feasible on Google Colab, inference speed may need optimization for mobile or embedded applications.

B. Future Enhancements

Future research will focus on expanding the dataset with more diverse artwork types and degradation patterns. Incorporating infrared and ultraviolet imaging, as demonstrated by Sizyakin et al. [1], can improve crack visibility under different light spectrums. Integration of attention mechanisms or vision transformers may also enhance the model's contextual understanding.

Further, the inclusion of image inpainting models can help build an end-to-end restoration pipeline, automatically detecting and digitally restoring cracked paintings. Finally, research into restoring not just paintings but sculptures, manuscripts, and architectural fragments can broaden the impact of this work.

IX CONCLUSION

This study presents a comprehensive deep learning-based framework for detecting cracks in ancient paintings. Leveraging CNN architectures and transfer learning, the system achieves high accuracy in identifying subtle and complex crack patterns. Through careful dataset preparation, rigorous preprocessing, and effective model evaluation, the solution offers a non-invasive, scalable, and accurate tool for art conservation. As museums and cultural institutions increasingly turn to digital technologies, such automated tools will become essential for sustainable preservation practices. The proposed model contributes meaningfully to this shift, demonstrating how modern AI can support the timeless task of heritage conservation.

X.REFERENCES

[1] R. Sizyakin, B. Cornelis, et al., "Crack detection in paintings using convolutional neural networks," *IEEE Access*, vol. 8, pp. 74535–74552, 2020.

[2] F.-C. Chen and M. R. Jahanshahi, "NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4392–4400, May 2018.

[3] L. Xu, S. Lv, et al., "A weakly supervised surface defect detection based on convolutional neural network," *IEEE Access*, vol. 8, pp. 42285–42296, 2020.

[4] B. Cornelis and T. Ruzic, "Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece," *Signal Process.*, vol. 93, no. 3, pp. 605–619, Mar. 2013.

[5] T. Ruzic and A. Pizurica, "Context-aware patch-based image inpainting using Markov random field modeling," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 1–14, Jan. 2015.
[6] L. Chen, J. Chen, et al., "Multi-view feature combination for ancient paintings chronological classification," *ACM J. Comput. Cult. Herit.*, vol. 10, no. 2, pp. 1–15, 2017.
[7] Y. Zeng and Y. Gong, "Nearest neighbor based digital restoration of damaged ancient Chinese paintings," in *Proc. IEEE 23rd Int. Conf. Digital Signal Process. (DSP)*, 2018.
[8] M. Ghorai, et al., "Multiple pyramids based image inpainting using local patch statistics and steering kernel feature," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5495–5509, Nov. 2019.
[9] S.-C. Pei and Y.-M. Chiu, "Background adjustment and saturation enhancement in ancient Chinese paintings," *IEEE Trans. Image Process.*, vol. 15, pp. 3230–3234, Dec. 2006.
[10] J. Yang, W. Wang, et al., "Infrared thermal imaging-based crack detection using deep learning," *IEEE Access*, vol. 7, pp. 2169–3536, 2019.
[11] X. Dong, et al., "Learning-based texture synthesis and automatic inpainting using support vector machines," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4777–4787, Jun. 2019.
[12] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.
[13] S. Tangsakul, et al., "Single image haze removal using deep cellular automata learning," *IEEE Access*, vol. 8, pp. 103181–103199, 2020.
[14] D. Ding, et al., "Image inpainting using nonlocal texture matching and nonlinear filtering," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1705–1719, Apr. 2019.
[15] H. N. Nguyen, T. Y. Kam, and P. Y. Cheng, "Automatic crack detection from 2D images using a crack measure-based B-spline level set model," *Multidimens. Syst. Signal Process.*, vol. 29, pp. 213–244, 2018.
[16] C. Peng, M. Yang, et al., "A triple-thresholds pavement crack detection method leveraging random structured forest," *Constr. Build. Mater.*, vol. 263, pp. 1–10, 2020.
[17] F. Bovolo, L. Bruzzone, et al., "A novel technique for subpixel image classification based on support vector machine," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2983–2999, Nov. 2010.
[18] L. Gomez-Chova, D. Tuia, et al., "Multimodal classification of remote sensing images: A review and future directions," *Proc. IEEE*, vol. 103, no. 9, pp. 1560–1584, Sep. 2015.
[19] F.-A. Georgescu, C. Vaduva, et al., "Feature extraction for patch-based classification of multispectral Earth observation images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 865–869, Jun. 2016.
[20] G. Li, et al., "Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine," *Autom. Constr.*, vol. 78, pp. 51–61, 2017.
[21] Q. Zhou, et al., "DeepInspection: Deep learning based hierarchical network for specular surface inspection," *Measurement*, vol. 160, pp. 1–14, 2020.
[22] B. Grabowski, et al., "Automatic pigment identification from hyperspectral data," *J. Cult. Herit.*, vol. 31, pp. 1–12, 2018; S. Liu, et al., "Novel features for art movement classification of portrait paintings," *Image Vis. Comput.*, vol. 108, pp. 1–12, 2021.
[23] Q. Guo, et al., "Patch-based image inpainting via two-stage low rank approximation," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 6, pp. 1–14, 2018.
[24] J. R. B. Taylor, A. Baradarani, et al., "Art forgery detection via craquelure pattern matching," in *Proc. Int. Workshop Comput. Forensics (IWCF)*, pp. 176–187, 2015.
[25] M. Ghorai, et al., "A group-based image inpainting using patch refinement in MRF framework," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 556–567, Feb. 2018.
[26] E. Sizikova, et al., "Wall painting reconstruction using a genetic algorithm," *ACM J. Comput. Cult. Herit.*, vol. 11, no. 1, pp. 1–17, 2018.
[27] A. Pasadas, et al., "Detection and classification of defects using ECT and multilevel SVM model," *IEEE Sens. J.*, vol. 20, no. 5, pp. 2329–2328, 2019.
[28] Chetan Gode, Bhushan Marutirao Nanche, Dharmesh Dhabliya, Rahul Dnyanoba Shelke, , Rajendra V Patil. & Shushma Bhosle, "Dynamic neural architecture search : A pathway to efficiently optimized deep learning models" , Journal of Information and Optimization Sciences, 46:4-A, 1117–1127, 2025
[29] Vinodpuri Rampuri Gosavi, Bhavna Ambudkar, Rajendra V. Patil, Rameshwar Dadarao Chintamani, Aashish G. Jagneet, Suman Kumar Swarnkar, "Personalized Drug Therapy Recommendations Based on Doctor's Clinical Descriptions Using AI", Research Journal of Pharmacy and Technology, 18(5), pp. 2385-2, 2025
[30] R.,V. Patil, R. M. Patil, D. Y. Bhadane, G. M. Poddar, A. L. Wakekar, S. R. Patil, "Adaptive Cheetah Optimization-Driven CNN: A Hybrid Approach for Robust Image Segmentation.", *International Journal of Basic and Applied Sciences, 14*(2), 691-702., 2025
[31] Sangeeta Kakarwal, Rahul Mapari, Pranoti Prashant Mane, Madhuri Pravin Borawake, Dhanraj R. Dhotre, Rajendra V. Patil, "A Novel Approach for Detection, Segmentation, and Classification of Brain Tumors in MRI Images Using Neural Network and Special C Means Fuzzy

Clustering Techniques", Advances in Nonlinear Variational Inequalities, 27(3), pp. 837–872, 2024.

[32] Jayasimha S R, Rajendra V. Patil, Yashashwini S, Prabu M, Srisathirapathy S, R Maranan, "Spilled Deep Capsule Neural Network with Skill Optimization Algorithm for Breast Cancer Recognition in Mammograms", 8th International Conference on Inventive Computation Technologies [ICICT 2025], April. 2025.

[33] B. Gudivaka, N. Anute, Y. Ramaswamy, V. N. Sankaran, R. V. Patil and J. N, "Fog and Edge Computing: Bridging the Gap Between Cloud and IoT," *2025 International Conference on Frontier Technologies and Solutions (ICFTS)*, Chennai, India, 2025

[34] A. Hidayat, A. Nurfikri, D. V. Priya, R. V. Patil, V. R. Hire and N. S, "Statistical Methods for Big Data Analysis in Technological Systems," *2025 International Conference on Frontier Technologies and Solutions (ICFTS)*, Chennai, India, 2025

[35] Rajendra V. Patil, Indrabhan S. Borse, Manesh P. Patil, Abhijit H. Khadke, Govind M. Poddar, Shravani R. Patil, "Ensuring Trust in Blockchain Enabled Business Processes using Smart Contract Audits", 8th International Conference on Inventive Computation Technologies [ICICT 2025], April. 2025.