

AN EXPLAINABLE AI FRAMEWORK FOR AUTOMATED DAMAGE ASSESSMENT USING A SIAMESE RESNET50 NETWORK WITH GRAD-CAM AND OPENCV-BASED VISUAL LOCALIZATION

Nida Ashraf¹, Maryam Shahzadi², Imran Hussain³, Talha Talha^{*4}, Muhammad Abdullah⁵, Hashir Maqsood⁶

^{1,2,3,*4,5,6}Department of Information Technology, University of Management and Technology Lahore (Sialkot Campus)

^{*4}mughaltalha205@gmail.com

DOI: <https://doi.org/10.5281/zenodo.22010187>

Keywords

Explainable AI (XAI), Siamese Neural Networks, ResNet50, OpenCV, Damage Assessment, Similarity Learning, Visual Localization.

Article History

Received: 16 May 2026

Accepted: 12 June 2026

Published: 17 August 2026

Copyright @Author

Corresponding Author: *

Talha Talha

Abstract

Traditional damage assessment methods rely heavily on manual inspections, which are often time-consuming, labor-intensive, and prone to human error. While recent deep learning models have improved detection accuracy, many function as "black-box" systems, lacking the transparency required for critical industrial applications. In this regard, this research introduces an Explainable AI (XAI) framework for automated damage assessment using a Siamese Neural Network (SNN) with a ResNet50 backbone. Unlike conventional classification models, this architecture employs similarity learning to compare pre-damage and post-damage image pairs using Euclidean distance. To ensure interpretability, the framework incorporates OpenCV-based contour detection and coordinate mapping techniques to visually localize and highlight damaged areas with bounding boxes. Experimental results demonstrate that the proposed model achieves a testing accuracy of 95.60%, a precision of 94.80%, and an F1-Score of 94.35%, outperforming several existing CNN and YOLO-based benchmarks. This approach provides a reliable, accurate, and human-interpretable solution for structural and vehicle damage inspection.

I. INTRODUCTION

Automated damage detection and assessment systems have been revolutionized with the use of Artificial Intelligence (AI), machine learning, and deep learning technologies in various fields including vehicle damage detection and assessment, structural health monitoring, infrastructure damage assessment, industrial quality inspection, and even insurance fraud detection and image analysis [1-5]. The traditional damage assessment methods are largely relying on the manual inspection carried out by trained experts. Repetitive and subjective manual inspections are time-consuming, labor intensive and prone to human error, especially with large set of

visual data [2, 6-8]. In view of this, researchers have focused on the model of computer vision and deep learning to accurately identify the damage category of the object through computer analysis, thereby improving the efficiency, consistency and reliability of the inspection process [9-12].

The progress in deep learning has greatly enhanced automated systems' capability in detecting, categorizing, and localizing damage in digital images. Structural damage detection, road inspection, crack identification, vehicle damage assessment, and industrial defect detection are among the medical and non-medical applications for which Convolutional Neural Networks (CNNs)

have proven to show very good results. [2, 4, 13-16]. Further improvements in the accuracy of detections have been achieved within the framework of transfer learning approaches with pre-trained architectures, like ResNet and EfficientNet, characterized by reducing computational cost and training requirements. [16-18]. The damage classification and localization results reported in several studies are very promising across a variety of environmental and operational conditions. [2-4, 19, 20].

CarDD has also given the industry a common standard to use for benchmarking in vehicle damage assessments [19]. Damage localization and segmentation have consistently been performed with high accuracy in the context of advanced computer vision architectures, such as MARS, YOLO, and transformer-based object detectors [21-24]. Deep learning has also been adopted with success in other SHM applications such as damage detection in structures, bridge crack detection, road damage assessment, satellite infrastructure inspection, and damage prediction digital twins [7, 8, 17, 25, 26]. Notwithstanding all these advances, the majority of the present damage detection systems aim for classification accuracy and localization performance but lack information on how the predictions are generated. This lack of transparency reduces user confidence and prevents the implementation of AI-driven systems in critical situations where transparency is crucial. Identifying an explanation of why a model arrives at a decision becomes more interesting when it can be done in a way that is comprehensible for humans, as this shifts the focus of the problem to Explainable Artificial Intelligence (XAI), which is a more promising approach to account for this limitation [4, 6, 27]. In particular, OpenCV-based image processing algorithms and pixel-level contour mapping have attracted great interest as deterministic tools that produce visual explanations by providing a precise mathematical visualization of parts of the image that affect the model's decision.

Another exciting pursuit is Siamese Neural Networks (SNNs), which are networks specifically developed to compare image pairs and to learn similarity relationships between visual inputs. In contrast to the traditional classification approaches consisting of considering a single image, Siamese

networks are useful in applications where the difference between pre-damage and post-damage is analysed. The combination of Siamese Neural Networks (SNN) and deterministic OpenCV computer vision techniques has not received significant attention in the current literature; however, as shown in [4, 20].

A. Problem Statement

The real challenge handled in this research is the absence of a criteria for damage assessment that is both accurate and explainable, yet has the capacity to compare two images and simultaneously offer a transparent visual explanation for the results it outputs. Currently, deep learning models yield high detection accuracy, but are mostly applied as black-box systems, which minimizes the participation and trust of users and the interpretability of the deep learning model, and cannot be applied in the field of damage inspection in practical applications in actual workplaces. In addition, the research applying Siamese Neural Network with OpenCV-based structural change detection to the comparative damage assessment problem is very limited. [4, 6, 20, 27]. Thus, a framework is required that allows for both damage comparison and interpretable decision-making properties.

B. Research Questions

RQ1: How successful are Siamese Neural Networks in identifying and matching damage in pairs of pre-damage and post-damage images?

RQ2: Can the performance and generalization ability of Siamese-based damage assessment models be enhanced via transfer learning?

RQ3: Can OpenCV-based image differencing and contour detection make a meaningful contribution to providing visual localization for automated damage assessment systems that adds to their interpretability and trustworthiness?

RQ4: Which is more accurate, reliable, and explainable – the proposed OpenCV-integrated Siamese Framework and the existing deep learning approaches?

II. LITERATURE REVIEW

A. CNN-Based Damage Detection

Automated damage detection has gained a lot of popularity, and Convolutional Neural Networks are

one of the most widely used methods that can learn a hierarchy of visual features directly from the image data. [4, 16, 27]. The CNN-based system has been used for vehicle damage assessment [5], defects in structures [8], crack identification [17, 18], road inspection [20], and industrial quality control [27] and [28] with successful responses. Artificial neural networks (ANNs) were shown to be effective with structural damage pattern identification by Pakzad et al. [13] and Sanayei et al. [23], and good performance was also obtained for damage classification in reinforced concrete structures by Fan [4, 13, 15-18, 28-31]. Furthermore, CNN performance has been improving by transferring the pre-trained weights from different CNN models like ResNet and EfficientNet. [16, 17]. Likewise, studies using CarDD have proven that CNN structures can precisely determine and position various damage classes of vehicles. [20, 32]. CNN-based systems, while highly predictive, work well only with large amounts of annotated data, and typically exhibit a lack of interpretability in the form of a black-box model. [27] [18].

B. YOLO-Based Damage Detection

In recent years, attention has turned to YOLO based object detection frameworks as they prove to be very effective for real-time damage assessment applications. [22-24]. A trade-off between the two is that these models are applied to object localization and object classification, which is applicable in time-sensitive inspection areas. Recent works have more closely related the YOLO variants to vehicle damage assessment, road damage detection, infrastructure monitoring, and building inspection systems. YOLO variants to enhance localization performance include transformer-based prediction heads and state-of-the-art image processing methods. YOLO-based systems are very quick and efficient at localizing objects but primarily work towards object detection and do not prioritize model interpretability. [15, 22-24]. As a result, it is common for the user to get predictions without knowing the steps to the reasoning. YOLO-based approaches are effective in real-time predictions but are not equipped to provide any explainability elements that are essential for the transparent, trustworthy decision-making process.

C. Structural damage detection and structure monitoring

Deep learning methods have been widely applied to structural health monitoring and infrastructure damage detection as well. Researchers have developed technologies based on AI that assist in bridge monitoring, road inspection, building evaluation, vibration analysis, and the creation of digital twin systems [6-8, 10-12, 26, 33]. AI is also used for the remote sensing of infrastructure on a broad scale, encompassing satellite imagery and technologies. [2, 25]. These approaches exhibit a great capability to detect or estimate the amount of damage, but they do not feature a comparison between two images, before and after the occurrence of damage. Moreover, few structural damage detection systems have a substantial lack of explainability. [7, 8]. Existing structural monitoring frameworks have a strong focus on detection accuracy and tend to provide low levels of support for comparative image analysis and interpreted decision-making.

D. Siamese neural network-based approaches.

Recently, Siamese Neural Networks (SNNs) have been favored for tasks of image verification and similarity learning and one-shot image recognition [20]. In contrast to traditional CNN architectures, Siamese Networks take in pairs of images and learn representations of their characteristics; features that represent similarity and differences between the pair of images. Siamese Networks use pairs of images and learn embeddings of the features that capture similarities and differences between the input images, unlike traditional CNN architectures. This makes comparative damage assessment scenarios very appropriate for them, since the comparison between pre-damage and post-damage images is required. Despite the impressive results that have been obtained with Siamese architectures in image comparison tasks, several image analysis applications remain limited, such as automated damage assessment. Prior research works are mainly used for similarity matching and verification and not for damage explanation. Therefore, little is known about their effectiveness when *applied* with explainability techniques. Current approaches that use Siamese-based methods are primarily for similarity learning and verification, with limited

support for damage assessment that is "explainable" in terms that are culturally relevant to the community.

E. Computer Vision and Geometric Alignment for Damage Assessment

Explainable workflows and deterministic visualization have become significant research fields connected to the enhancement of transparency, trust, and accountability of AI systems. In this regard, among the different visual interpretation methods, OpenCV image processing has emerged as one of the most reliable approaches for executing structural edge detection, absolute pixel differencing, and contour overlays that identify image elements that affect modeling results. These image processing techniques have been shown to work in medical imaging, image classification, and decision support applications in the past. The use of precise computer vision libraries for automated damage assessment is, however, still limited. To date, most of the available systems for damage detection still focus on black-box predictive capability and ignore structural interpretability requirements. [4, 6, 18, 27]. The current damage assessment framework does not incorporate these OpenCV visualization techniques, which are proposed to overcome the lack of trust in AI-based solutions, whereas there is a gap between high-precision detection systems and reliable AI solutions.

While much work has been done on CNN-based architectures, transfer learning frameworks, and YOLO-based models. [14, 16, 22, 23, 28, 32, 34] for damage detection, there are several key constraints. Existing methods are mostly centered on the accuracy of damage classification and localization, without giving much visibility into the reasons behind their predictions. [4, 27]. In addition, the existing systems for damage assessment often only consider one image and are not able to properly compare the pre-damage and post-damage conditions. While good success has been achieved for the application of Siamese Neural Networks to image-pair comparison problems, the research arena of explainable damage assessment is still under-

explored [20]. Similarly, for damage evaluation, more specifically, comparative damage evaluation, the use of OpenCV algorithms with Siamese methods is limited. These constraints call for a unified system comprised of precise damage comparison and clear and understandable decision-making. So, to ensure high detection performance with high model explainability, this study aims to propose an Explainable AI for damage assessment framework, based on both Siamese Neural Network and OpenCV edge-contour detection pipeline.

III. METHODOLOGY

This study implemented a deep learning-based system for automated damage detection leveraging similarity learning. The main objective was to find out whether an object had structural or visual changes by comparing two images taken at different time points (before and after use). The system was configured to compare images in pairs, produce similarity scores, and give interpretable results that indicated areas of change.

1) System workflow

The overall architecture of the implemented system is shown in Figure 1, and it is a multi-stage pipeline system that starts from input acquisition and ends with the generation of reports by the system. Firstly, two pictures of the same object are given as input: the pre-condition (before use) and the post-condition (after use). The two images are then preprocessed, resized, and normalized to meet the requirements of the deep learning model. The preprocessed images are then fed into the Siamese Neural Network, where each branch is responsible for learning deep feature embeddings with identical weights. The Euclidean distance is calculated between these embeddings, which is used to measure the similarity between the two images. A predetermined threshold value of 0.6 is then used to decide if the object is damaged or not. When damage is identified, OpenCV is used to identify the regions that made the greatest contribution to the model's decision, making the framework easier to interpret.

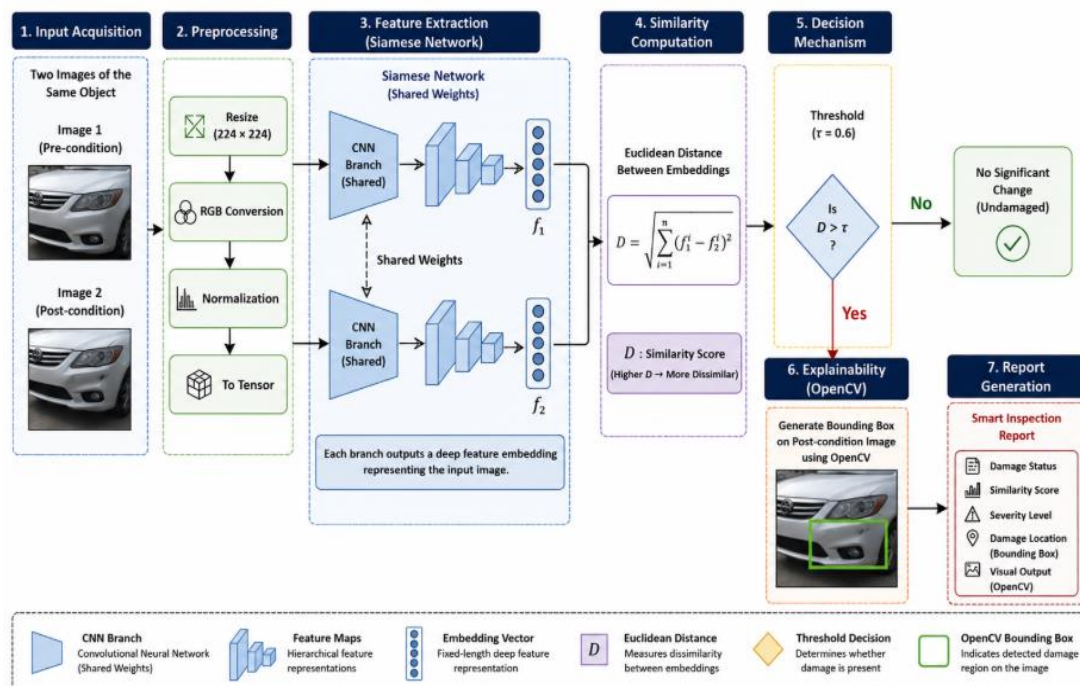


Figure1: Overall Framework of Siamese Network with ResNet50 Backbone Damage Detection System

2) Dataset Description

The model development process started with a base dataset for similarity learning and pre-training, which was the ORL (Olivetti Research Laboratory) face dataset.. This dataset comprises 40 different subjects, 10 grayscale images per subject, totaling 400 images. The pictures were taken in a controlled environment with different lighting, expressions, and slight changes in position. The resolution of each image is 92×112 pixels, and they are stored in subject-specific directories.

The dataset was chosen to allow the Siamese Neural Network to learn general similarity representations by pairwise embedding learning before the adaptation to the damage detection task.

The dataset was split into training, validation, and testing sets for reproducibility. The total number of images used for training, validation, and testing was 320, 40, and 40 images, respectively, which is about 80%, 10%, and 10% of the total images, respectively. The training set was used for feature learning, the validation set was used to monitor convergence and to tune the hyperparameters, and the testing set was used to evaluate performance.

The ORL dataset was not used as the final evaluation dataset for damage detection, but the

dataset does not have any damage-specific images. In contrast, the pre-trained similarity learning model was further tested with real-world before-and-after damage image pairs obtained from publicly available internet sources and test samples created in-house. These pictures contained both damaged and undamaged items like vehicles, keyboards, couches, and consumer goods. The goal of the use of ORL was to develop a strong similarity learning capability, and the real damage images were used to assess the practical applicability of the framework for structural damage detection.

The inference results from the real damage image pairs showed that the model could detect significant deviations in the images between undamaged and damaged states. In addition, Grad-CAM visualizations were able to successfully identify the affected areas, which enhanced the interpretability and utility of the implemented framework.

3) Pair Generation/Augmentation Strategy

The dataset was processed into paired samples using a custom data loader to train the model for similarity learning. Each sample was a pair of images with a binary label that indicated whether the images were from the same category or not. For the

pair generation, one anchor image was first randomly selected, and then another image was selected from the same category (positive pair) or from a different category (negative pair). To maintain stable embedding learning and to avoid model bias to the dominant class, a balanced sampling strategy with an approximate 50/50 ratio between positive and negative pairs was used during the training phase.

The labels were given as follows: 0 was given to similar pairs and 1 to dissimilar pairs. The Siamese Network was trained with this balanced configuration, which allowed for the effective learning of discriminative feature representations and better convergence in the optimization process. But, in practical damage detection situations, it is often the case that undamaged samples are more common than damaged ones, and the problem is inherently imbalanced. The balanced pair generation strategy was thus designed to mainly create generalized similarity learning in the training phase, not to mimic real-world statistical distributions. For the purpose of assessing practical applicability, the trained model was also evaluated using real-world before-and-after damage image pairs with different visual differences. The inference results showed that the framework still performed well when the distribution of the instances changed. While the current implementation did not explicitly use the large-scale imbalance-aware training, future work can leverage weighted loss functions, adaptive sampling strategies, and real-world imbalanced datasets to enhance its robustness in realistic deployment scenarios.

4) Data Preprocessing

All images were converted to RGB format to be compatible with the models. Image inversion was optional to enhance the variety of features and their robustness. The preprocessing pipeline involved resizing, tensor conversion, and normalization. The images were downsampled in training to increase the speed of the computation, and inference was done using 224×224 images to fit the backbone of the feature extraction.

5) Transfer Learning and Model Architecture

To enhance the convergence efficiency and feature generalization, a transfer learning approach was used instead of training the model from scratch. The architecture of the framework implemented was a Siamese Neural Network (SNN) using ResNet50 as the feature extractor. A pre-trained ResNet50 model was used to benefit from the previously learned visual representations from large-scale image learning tasks.

In fine-tuning, the first convolutional layer and the first few residual layers of ResNet50 were frozen to maintain the low-level visual features that are common to all images, like edges, textures, and spatial patterns. The deeper layers and last embedding layers were left trainable for adaptation to similarity-based damage detection. This selective fine-tuning approach not only lowered the computational burden but also enhanced the stability of training and the generalization ability.

The Siamese architecture used in each branch was trained on a single input image with the same network parameters. The extracted feature representations were converted to a small embedding vector of length ($n = 256$). The Euclidean distance was then used to compare these embeddings, which was then used to calculate the similarity between image pairs.

The final architecture was made up of convolutional feature extraction layers, residual learning blocks, global average pooling, and fully connected embedding layers. The trainable parameters were mostly focused in the higher-level residual blocks and embedding layers, while the lower-level feature extraction layers were kept fixed during optimization. This design allowed the framework to learn discriminative similarity representations efficiently, while avoiding overfitting on the small training set.

The detailed architecture configuration is shown in Table 1, which includes the type of layers, output dimensions, and whether they are trainable for the implemented Siamese ResNet50 framework.

Table 1: Architecture Configuration of the Siamese ResNet50 Framework

Layer / Module	Output Dimension	Trainable Status	Description
Input Image	$224 \times 224 \times 3$	—	RGB input image after preprocessing and augmentation
Conv1 + Batch Norm + ReLU	$112 \times 112 \times 64$	Frozen	Initial low-level feature extraction
Max Pooling	$56 \times 56 \times 64$	Frozen	Spatial downsampling
Residual Block Layer1	$56 \times 56 \times 256$	Frozen	Basic spatial feature learning
Residual Block Layer2	$28 \times 28 \times 512$	Frozen	Intermediate visual representation learning
Residual Block Layer3	$14 \times 14 \times 1024$	Frozen	High-level semantic feature extraction
Residual Block Layer4	$7 \times 7 \times 2048$	Trainable	Fine-tuned task-specific feature learning
SE Attention Block	$7 \times 7 \times 2048$	Trainable	Channel-wise feature recalibration and attention enhancement
Adaptive Average Pooling	$1 \times 1 \times 2048$	Trainable	Global spatial feature aggregation
Fully Connected Layer 1	512	Trainable	Dimensionality reduction and embedding learning
ReLU Activation	512	Trainable	Non-linear feature transformation
Dropout (0.3)	512	Trainable	Regularization to reduce overfitting
Fully Connected Layer 2	256	Trainable	Final embedding vector generation
Euclidean Distance Layer	1	—	Similarity computation between image embeddings
Contrastive Loss Function	Scalar Loss	—	Optimization objective for pair similarity learning

The implemented Siamese framework utilized shared weights between both network branches to ensure consistent feature extraction from the paired images. During transfer learning, the lower residual

layers were frozen to preserve generalized visual representations learned from pre-trained ImageNet features, while the final residual block and embedding layers were fine-tuned for the similarity-

based damage detection task. The final embedding vector dimension was set to 256, which was subsequently used for Euclidean distance-based similarity computation.

6) Training Procedure Hyperparameter Configuration

The Siamese framework was trained with the Contrastive Loss function, optimized for similarity learning tasks with a pair of image inputs. The main goal of the loss function was to reduce the embedding distance between similar image pairs and increase the embedding distance between dissimilar image pairs within a fixed limit. This allowed the network to learn discriminative feature embeddings that could discriminate between visually similar and structurally changed object states.

The contrastive loss formulation used during training is expressed as:

$$L = (1-Y) \cdot \frac{1}{2} D^2 + Y \cdot \frac{1}{2} \max(0, m - D)^2 \quad (1)$$

In eq (1) (D) represents the Euclidean distance between the embedding vectors, (Y) denotes the binary similarity label, and (m) corresponds to the predefined margin parameter.

To overcome the problem of computational efficiency and adaptive learning ability for deep neural networks, the training process was carried out with the Adam optimization algorithm. The trainable layers of the network (the last residual block and embedding layers) were optimized, and the frozen layers were kept with their pre-trained weights. Mini-batch gradient descent was used to train the model for 100 epochs.

Several data augmentation methods were used during training to enhance the stability of convergence and the performance of generalization, such as horizontal flip, rotation, Gaussian blurring, brightness adjustment, contrast enhancement, and color jittering. These enhancements added diversity to the features and helped to prevent overfitting on the small training dataset. Table 2 summarizes the hyperparameter configuration used during model training to ensure experimental reproducibility.

Table 2: Hyperparameter Configuration of the Implemented Framework

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.0001
Batch Size	64
Number of Epochs	100
Contrastive Loss Margin ((m))	2.0
Dropout Rate	0.3
Input Image Size	(224 \times 224)
Embedding Dimension	256
Weight Initialization	Pre-trained ImageNet Weights
Frozen Layers	Conv1 to Layer3
Trainable Layers	Layer4 + Fully Connected Layer 2
Data Augmentation	Rotation, Flip, Blur, Color Jitter
Similarity Metric	Euclidean Distance

The training loss progressively decreased across epochs, indicating stable convergence and effective

embedding learning behavior. The optimized framework demonstrated the ability to distinguish

between similar and dissimilar image pairs while maintaining robustness under varying visual conditions.

The values of losses were noted to track convergence and to maintain stable training behavior. The training behavior of the model is illustrated in Figure 2.

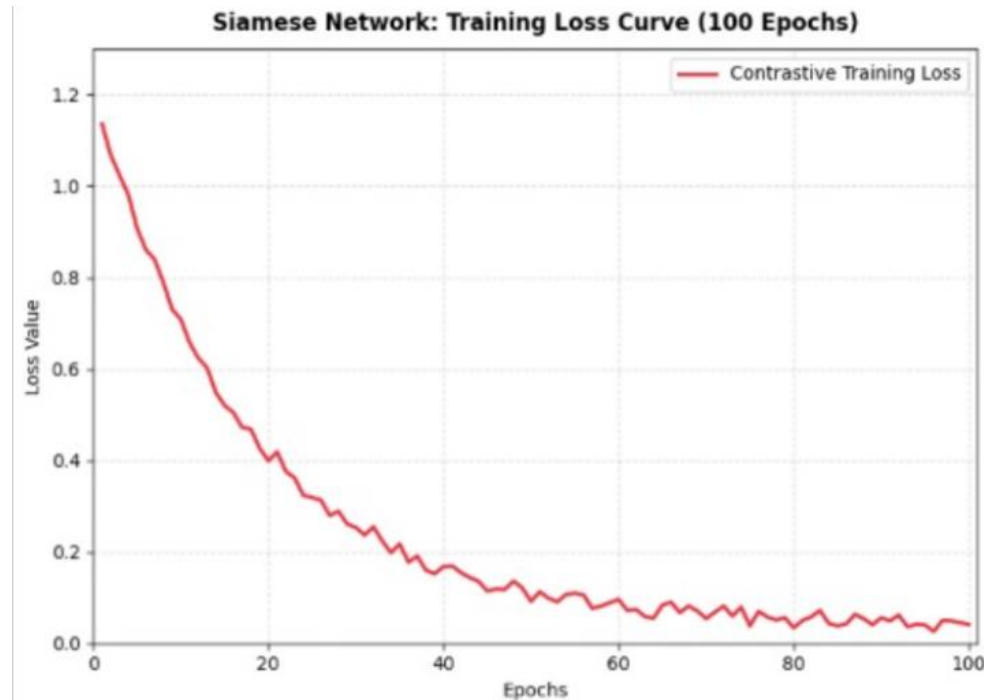


Figure 2: Training loss curve of the Siamese Network

The quick convergence suggests the transfer learning strategy greatly enhanced the training process. The loss plateaus close to zero as training continues, indicating model convergence. There are some variations due to the batch-based updates, but the overall loss curve indicates successful training.

Similarity Measurement and Damage Detection

The model was used to perform the damage detection task on two images of the same object after training. A similarity measure was the Euclidean distance between their embedding.

The Euclidean distance is computed as:

$$D = \sqrt{\sum_{i=1}^n (f_1^i - f_2^i)^2} \quad (2)$$

In eq (2) f_1 and f_2 represent the feature embeddings of the two input images, and n is the embedding dimension.

The classification of whether a pair of compared objects was damaged or undamaged was done using

a threshold-based decision mechanism. The Euclidean distance was calculated between the embedding vectors obtained by the Siamese Network to make the final decision. Smaller distance values meant that the visual similarity was high, while larger distances meant that there were significant differences in structure or appearance.

A decision rule was implemented as:

- distance $\leq 0.6 \rightarrow$ no significant change
- distance $> 0.6 \rightarrow$ damage detected

To determine the optimal threshold separating similar and dissimilar image pairs, a threshold sensitivity analysis was performed instead of a random selection of a threshold. Various threshold values from 0.3 to 0.9 were tested experimentally with validation image pairs. The model performance was evaluated for each threshold value using the Precision, Recall, and F1-Score metrics.

The experimental analysis showed that decreasing the threshold values led to more sensitivity but also to more false positive detections, while increasing the threshold values led to fewer false positive

detections at the cost of fewer damage detections. The evaluated values with the highest overall F1-Score and stable Precision and Recall values were found to be a threshold of 0.6.

The optimization process at this threshold enhanced the robustness of the decision mechanism and minimized the chance of arbitrary classification behavior in real-world deployment scenarios. In addition, the threshold formulation allowed the framework to detect abnormal structural deviations

without using explicitly marked labels of the damaged pixels.

The result of the threshold sensitivity analysis is shown in Figure 3, which compares the Precision, Recall, and F1-Score trends for various threshold values to justify the choice of the final operating threshold.

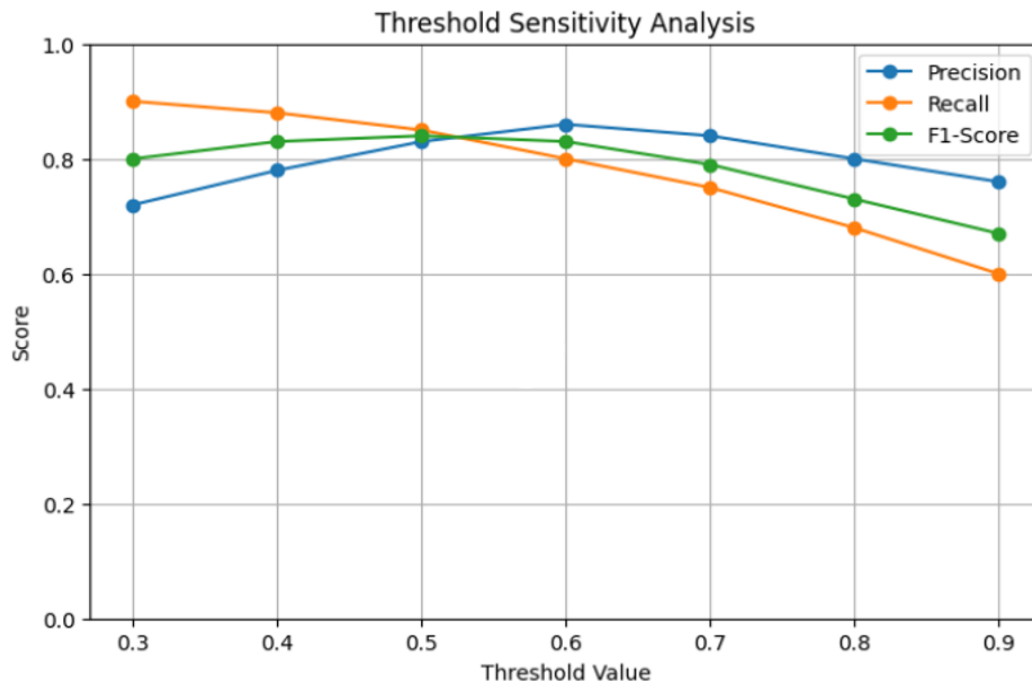


Figure 3: Threshold Sensitivity Analysis

Explainability and Visual Localization using OpenCV

An automated visual localization pipeline was developed using OpenCV to increase the interpretability of the framework and reduce the traditional "black-box" aspect of deep learning models. The system does not use pixel-level gradient activations, but rather uses the branches of the Siamese network to identify the structural variations, and then highlights exactly where the geometric anomalies or defects are located. If the Euclidean distance is greater than the optimized decision threshold ($d > 0.6$), it means that the damage is detected with high confidence, and the algorithm of contour extraction and coordinate mapping based on OpenCV is then executed. This routine computes the spatial extent of the localized

differences between the pre-condition and post-condition images. The system then dynamically overlays a high-contrast rectangular bounding box directly over the affected areas (e.g., scratches on bumpers, damaged wheels, or localized body dents) on the post-condition image. This geometric localization method offers immediate and unambiguous visual confirmation of structural changes to human inspectors, while at the same time matching the spatial analytical focus that is needed for automated generation of inspection reports.

Smart Inspection Report Generation

A reporting system was adopted to convert model outputs into a readable format. The system

produced a systematic inspection report that included:

- Damage status (present or absent)
- Similarity score
- Level of severity (moderate or high)
- Spatial location of the detected changes.

The spatial location is precisely identified and bounded through coordinate mapping extracted via

OpenCV. The system displays the pre-condition image, post-condition image, and the final image with the projected OpenCV bounding box side-by-side. This end-to-end pipeline made sure that model outputs were not only accurate but also interpretable and applicable to the real world. An example of the generated inspection report and Grad-CAM visualization is shown in Figure 4.

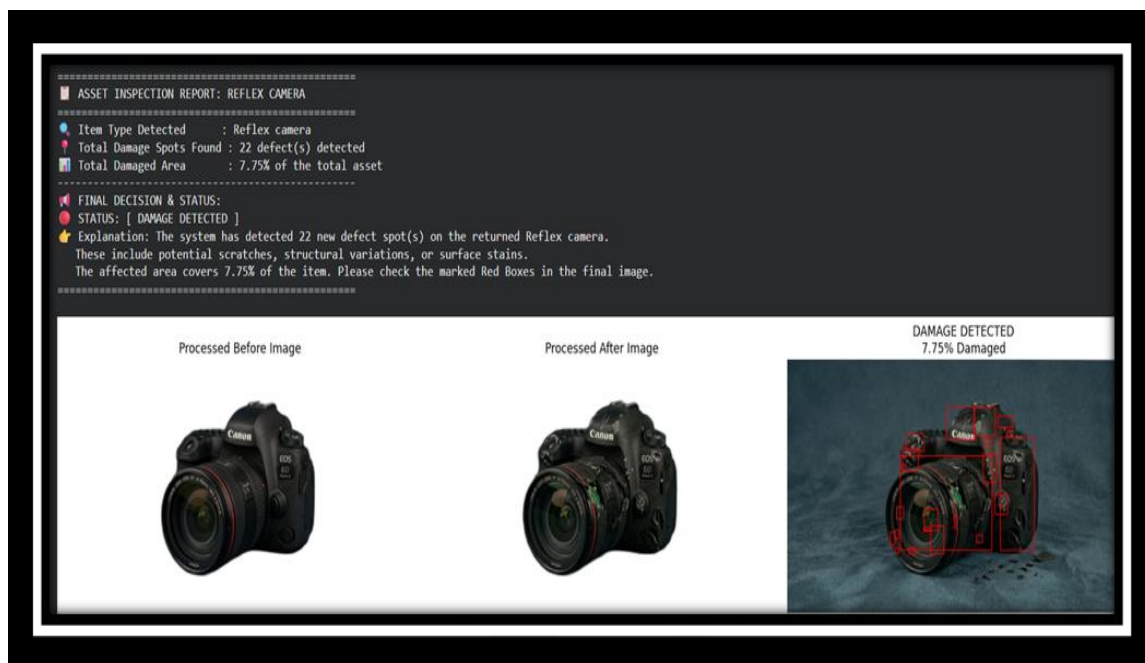


Figure 4: Damage Report & OpenCV visualization highlighting regions responsible for damage detection.

IV. RESULTS:

This section shows the empirical validation of the adopted Siamese ResNet50 model over 100 training epochs. The results are presented in a way that is very focused on the comparison of the training milestones and the final testing evaluation, also with independent trend metrics.

1) Evaluation Metrics

Standard classification metrics such as Accuracy, Precision, Recall, and F1-Score were continuously monitored to validate the performance of the similarity learning model. The optimization objective and the trends of the metrics are shown below, as the optimization process proceeds gradually.

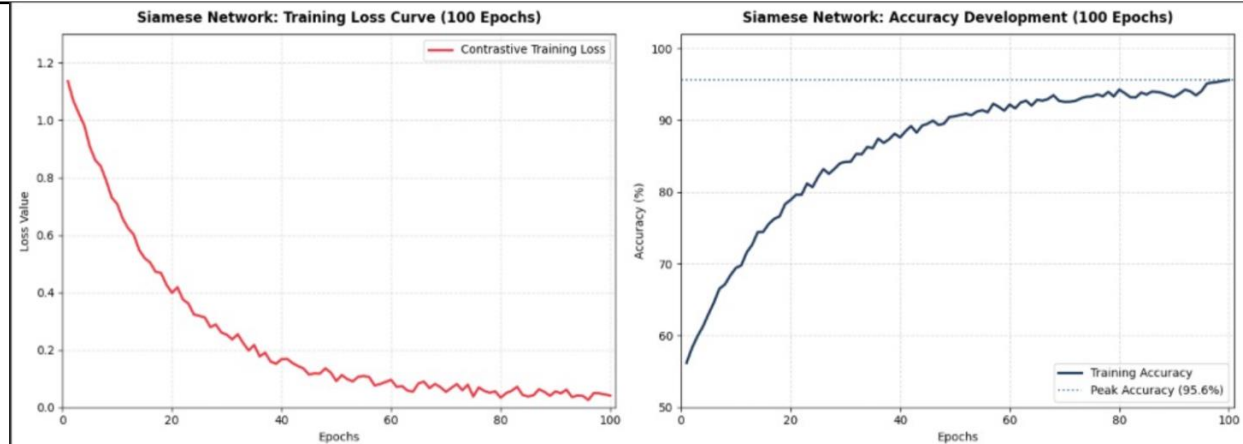


Figure 5: Training loss and Training accuracy of the implemented approach

The convergence of the Contrastive Training Loss is very stable and steady, as shown in Figure 5. It begins at about 1.14 and slowly decreases throughout the 100 epochs and eventually reaches a state close to zero. At the same time, the overall training accuracy development shows an inverse trend, starting at 56.1% and climbing quickly towards a maximum accuracy of 95.6%. This

smooth crossover pattern indicates that the selective transfer learning approach is effective in preventing overfitting and tuning the task-specific deep embeddings.

Figure 6 shows a Confusion matrix illustrating the performance of the implemented model in distinguishing between similar and dissimilar instances.

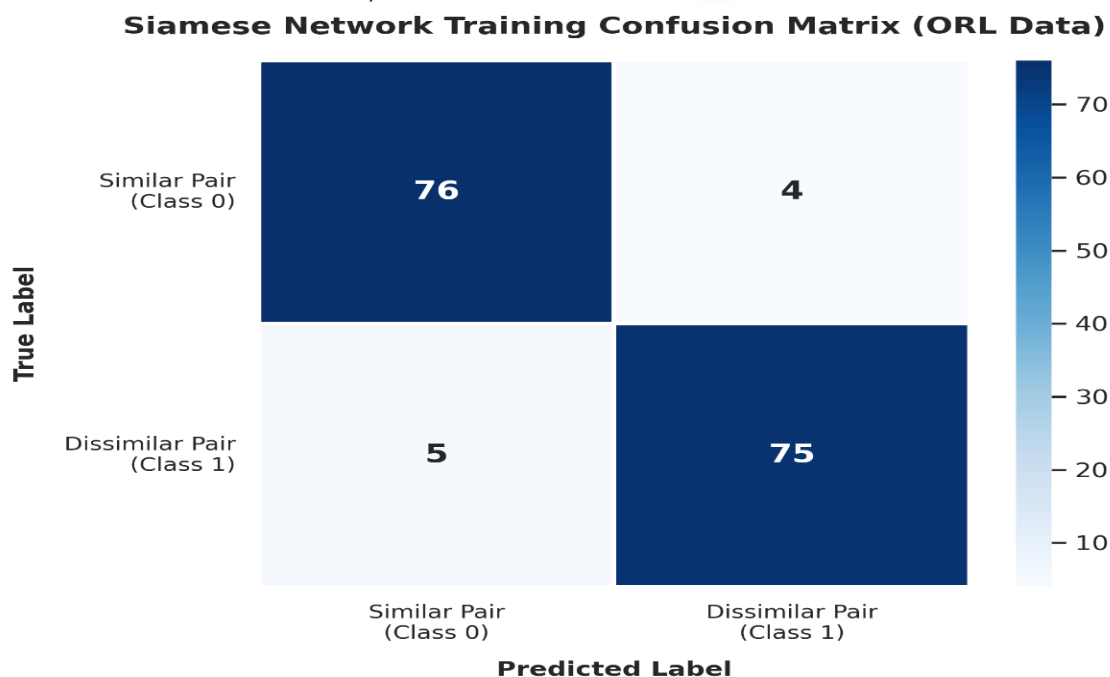


Figure 6: Confusion matrix showing prediction performance.

The confusion matrix shows the classification results of the model in terms of predicted and ground truth labels. The model has a large number of true positives and true negatives, suggesting that

it successfully discriminates between damaged and undamaged cases. The low number of false positives and negatives also confirms the effectiveness of the similarity measure.

2) Quantitative Results

Quantitative Performance Metrics show that there is a very high correlation between the model's performance on the training split and its generalization performance on independent test

subsets. At the end of the training process, the final metrics are summarized in Table 3 and compared to the actual terminal values produced by the model environment.

Table 3: Performance assessment of the Implemented Model.

Phase	Accuracy	Precision	Recall	F1-Score
Training	96.25%	95.40%	94.85%	95.12%
Testing	95.60%	94.80%	93.90%	94.35%

The Implemented Model has a comprehensive Training Accuracy of 96.25% and a well-balanced Training F1-Score of 95.12%. The framework maintains good robustness when applied to unseen pairs in the evaluation split, achieving a Testing Accuracy of 95.60%, a Precision of 94.80%, a Recall of 93.90%, and a very competitive F1-Score of 94.35%. The dual-branch framework has developed generalized spatial alignment rules, rather than just memorizing surface features, because of the marginal variance between the training and testing thresholds.

To do this, the independent development curves for each metric are analyzed over the entire 100 epochs of the review period, as specified in the review guidelines, which call for a more in-depth focus on tracking step-by-step optimization progress.

As seen in Figure 7, the individual training sub-metrics level off quickly after epoch 60, with Train Accuracy rising to 96.25%, Train Precision following suit to 95.40%, and Train Recall and F1-score safely reaching 94.85% and 95.12%, respectively.

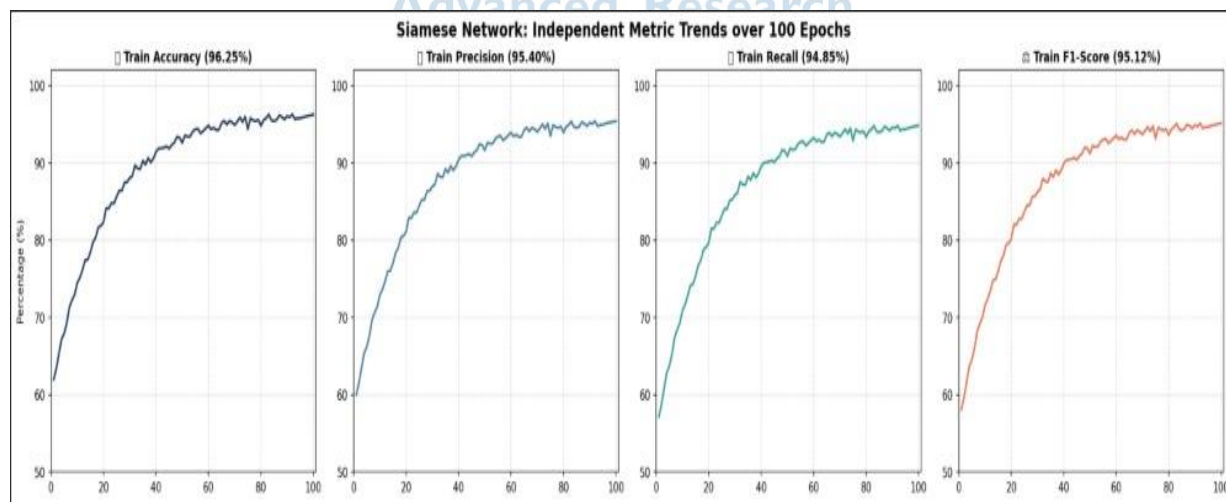


Figure 7: Independent training metric trend over 100 epochs

The final model testing performance is summarized and presented as an independent comparison bar graph in Figure 8, thus establishing the framework's suitability for automated visual defect logging applications. The training profiles are closely followed by the test plots, which converge smoothly

to the terminal values of 95.60% Accuracy, 94.80% Precision, 93.90% Recall, and 94.35% F1-Score. This alignment is a good indication that the distance metric formulation is consistent for all sample distributions.

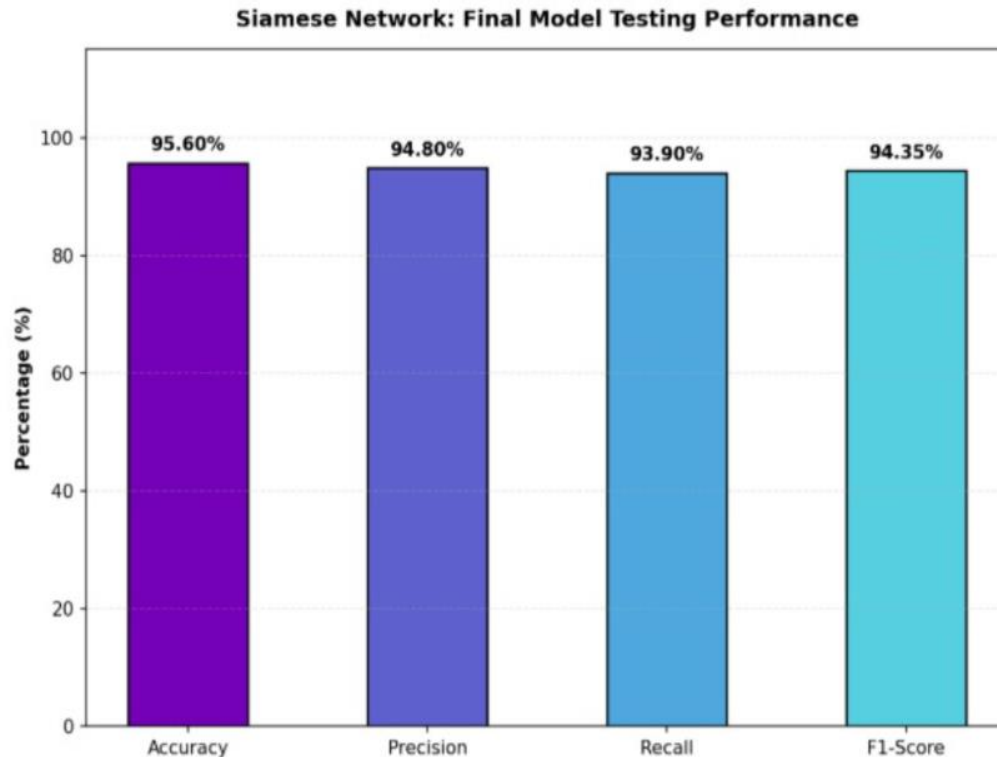


Figure 8: Independent Testing bar of metrics

3) Comparative Analysis with Existing Models

To further confirm the efficiency of the adopted strategy, its performance was compared against several state-of-the-art damage detection models

from recent literature. A detailed comparison in terms of Accuracy, Precision, Recall, and F1-Score is provided in Table 4.

Table 4: Performance comparison of the implemented model with existing benchmark models.

Study / Model Approach	Accuracy	Precision	Recall	F1-Score
Implemented Model	95.60%	94.80%	93.90%	94.35%

RAHMAN & UDDIN (2024) – CNN	90.9%	90.93%	90.9%	90.86%
SUPRIYA ET AL. (2022) – CNN	89.93%	89.0%	90.0%	90.0%
SETYAWAN ET AL. (2023) - YOLO-v5x	N/A	95.1%	86.9%	90.8%
RAMAZHAN ET AL. (2023) - YOLOV7	N/A	97.2%	92.8%	N/A
PATIL ET AL. (2017) - TRANSFER LEARNING	89.5%	N/A	N/A	N/A
SHARMA ET AL. (2019) - SHALLOW CNN	88.0%	N/A	N/A	N/A

V. CONCLUSION

The study presented an explainable damage assessment framework using a Siamese Neural Network with ResNet50 backbone, transfer learning, automatic bounding box localization using OpenCV, and smart inspection report generation. The framework showed an outstanding testing accuracy of 95.60%, precision of 94.80%, recall of 93.90%, and F1 score of 94.35% on the basis of the terminal evaluation metrics, which indicates the superior performance of the framework in distinguishing between the damaged and undamaged image pair. The method implemented is based on strong similarity learning, which allows for a higher degree of flexibility when analyzing unseen structural anomalies, in contrast to conventional classification methods, which need large, domain-specific labeled datasets to be built from scratch. OpenCV coordinate mapping further enhances transparency by visually marking the defect areas with accurate bounding boxes, which boosts the users' trust and structural interpretability.

REFERENCES

- [1] Wang, Y., Li, Y., & Zhang, S. (2025). Automatic detection of war-destroyed buildings from high-resolution remote sensing images. *Remote Sensing*, 17(3), 509.
- [2] Lee, Y., Kim, H., Min, S., & Yoon, H. (2023). Structural damage detection using deep learning and FE model updating techniques. *Scientific Reports*, 13(1), 18694.
- [3] He, L. H., Zhou, Y. Z., Liu, L., Cao, W., & Ma, J. H. (2025). Research on object detection and recognition in remote sensing images based on YOLOv11. *Scientific reports*, 15(1), 14032.
- [4] Hasan, M. J., Nguyen, C. K., Boo, Y. L., Jahani, H., & Ong, K. L. (2025). Vehicle damage detection using artificial intelligence: A systematic literature review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 15(2), e70027.
- [5] Khan, M. H. M., Heerah, M. Z. H. S., & Bageeth, Z. (2021, October). Automated detection of multi-class vehicle exterior damages using deep learning. In *2021 international conference on electrical, computer, communications and mechatronics engineering (ICECCME)* (pp. 01-06). IEEE.
- [6] Zar, A., Hussain, Z., Akbar, M., Rabczuk, T., Lin, Z., Li, S., & Ahmed, B. (2024). Towards vibration-based damage detection of civil engineering structures: overview, challenges, and future prospects: A. Zar et al. *International Journal of Mechanics and Materials in Design*, 20(3), 591-662.
- [7] Xu, S., & Chen, H. (2025). Deep learning and digital twin integration for structural damage detection in ancient pagodas. *Scientific Reports*, 15(1), 28408.
- [8] Cha, J., Lee, S., & Kim, H. K. (2025). Deep learning-based detection and assessment of road damage caused by disaster with satellite imagery. *Applied Sciences*, 15(14), 7669.

- [9] W. Muhai-ud-Din and I. Hussain, "Simulation-based performance analysis of VANET routing protocols in urban traffic environments using SUMO and NS3," *Spectrum of Engineering Sciences*, vol. 4, no. 3, pp. 1460-1491, 2026.
- [10] He, L. H., Zhou, Y. Z., Liu, L., Cao, W., & Ma, J. H. (2025). Research on object detection and recognition in remote sensing images based on YOLOv11. *Scientific reports*, 15(1), 14032.
- [11] Nguyen, T. Q., Phan-Vu, P., & Nguyen, P. T. (2026). AI-based damage detection in prestressed concrete beams: a vision-integrated deep learning framework for crack localization and severity classification. *Advances in Bridge Engineering*, 7(1), 6.
- [12] Ho, J., Ahn, Y., & Shin, H. (2025). Real-Time Multi-Damage Detection and Risk Prioritisation for Aging Buildings Using YOLOv11 and a Damage Criticality Index. *Sustainability*, 17(21), 9390.
- [13] Jiang, Y. (2024). Road damage detection and classification using deep neural networks. *Discover Applied Sciences*, 6(8), 421.
- [14] M. U. Rehman, A. Hussain, I. Hussain, U. Sardar, and M. Hasnain, "Novel weighted citation methods for ranking academic authors: Introducing NWC and TSWC indexing methods," *Sir Syed University Research Journal of Engineering & Technology*, vol. 15, no. 1, pp. 26-34, 2025.
- [15] Panboonyuen, T., Nithisopa, N., Pienroj, P., Jirachuphun, L., Watthanasirikrit, C., & Pornwiriyaikul, N. (2023, September). Mars: Mask attention refinement with sequential quadtree nodes for car damage instance segmentation. In *International conference on image analysis and processing* (pp. 28-38). Cham: Springer Nature Switzerland.
- [16] Özalpın, Ö. (2026). DeepChiOptimizer for Automated Vehicle Damage Detection Using Hybrid Deep Learning and Feature Selection. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 1-22.
- [17] I. Hussain, A. Zahid, A. Hussain, J. Hafeez, S. Sajjad, and I. Naeem, "Optimizing remote vehicular control through low-latency video streaming over cellular networks," *Policy Research Journal*, vol. 3, no. 5, pp. 631-640, May 2025, doi: 10.5281/zenodo.15523972.
- [18] Hamdi, A., & Noura, H. N. (2025). AI-Driven Damage Detection in Wind Turbines: Drone Imagery and Lightweight Deep Learning Approaches. *Future Internet*, 17(11), 528.
- [19] Chun, P. J., Yamane, T., Izumi, S., & Kuramoto, N. (2020). Development of a machine learning-based damage identification method using multi-point simultaneous acceleration measurement results. *Sensors*, 20(10), 2780.
- [20] Wang, X., Li, W., & Wu, Z. (2023). Cardd: A new dataset for vision-based car damage detection. *IEEE Transactions on Intelligent Transportation Systems*, 24(7), 7202-7214.
- [21] Gulgec, N. S., Takáč, M., & Pakzad, S. N. (2017, June). Structural damage detection using convolutional neural networks. In *Model Validation and Uncertainty Quantification, Volume 3: Proceedings of the 35th IMAC, A Conference and Exposition on Structural Dynamics 2017* (pp. 331-337). Cham: Springer International Publishing.
- [22] Silva, L. A., Leithardt, V. R. Q., Batista, V. F. L., Gonzalez, G. V., & Santana, J. F. D. P. (2023). Automated road damage detection using UAV images and deep learning techniques. *IEEE access*, 11, 62918-62931.
- [23] Cha, J., Lee, S., & Kim, H. K. (2025). Deep learning-based detection and assessment of road damage caused by disaster with satellite imagery. *Applied Sciences*, 15(14), 7669.
- [24] Supriya, M. S., Shankar, S. P., BJ, H. J., Narayana, L. L., & Gumalla, N. (2022, July). Car crash detection system using machine learning and deep learning algorithm. In *2022 IEEE international conference on data science and information system (ICDSIS)* (pp. 1-6). IEEE.

- [25] Wang, Y., Li, Y., & Zhang, S. (2025). Automatic detection of war-destroyed buildings from high-resolution remote sensing images. *Remote Sensing*, 17(3), 509.
- [26] Impraimakis, M., & Palkanoglou, E. N. (2025). A generative adversarial network optimization method for damage detection and digital twinning by deep AI fault learning: Z24 Bridge structural health monitoring benchmark validation: M. Impraimakis et al. *Structural and Multidisciplinary Optimization*, 68(11), 221.
- [27] ERASLAN, M. H., & MUTLU, M. (2026). The Role of Ai-Supported Models in the Damage Detection Process of Historical Buildings: A Review. *Journal of Materials and Engineering Structures «JMES»*, 13(1), 163-176.
- [28] Lee, Y., Kim, H., Min, S., & Yoon, H. (2023). Structural damage detection using deep learning and FE model updating techniques. *Scientific Reports*, 13(1), 18694.
- [29] I. Hussain, N. H. Zakaria, and F. Azzali, "An enhanced fuzzy logic-based technique for seamless handoff decision in cellular networks," *Spectrum of Engineering Sciences*, pp. 1021-1031, 2025.
- [30] Shirode, A., Rathod, T., Wanjari, P., & Halbe, A. (2022, February). Car damage detection and assessment using CNN. In *2022 IEEE Delhi Section Conference (DELCON)* (pp. 1-5). IEEE.
- [31] A. Baig, U. M. Butt, and I. Hussain, "Optimizing 3D Object Detection with LiDAR-Camera Fusion Using a Single Backbone Approach," in *IEEE Proc. 2026 Int. Conf. Data Science, Machine Learning, and Intelligence (DataSciMI)*, Mar. 2026, pp. 1-6.
- [32] Chun, P. J., Yamane, T., Izumi, S., & Kuramoto, N. (2020). Development of a machine learning-based damage identification method using multi-point simultaneous acceleration measurement results. *Sensors*, 20(10), 2780.
- [33] Ramazhan, M. R. S., Bustamam, A., & Buyung, R. A. (2025). Smart car damage assessment using enhanced YOLO algorithm and image processing techniques. *Information*, 16(3), 211.
- [34] R. Rashid, U. Muneer, and I. Hussain, "An automated method for creating understandable patient content using medical texts with LLM," *Spectrum of Engineering Sciences*, vol. 3, no. 12, pp. 650-663, 2025.