Journal of Artificial Intelligence and Technology, 2026, 6, 236-256
https://doi.org/10.37965/jait.2026.0920

ISTP
RESEARCH ARTICLE

CropResMoE-50: A Region-Aware Mixture-of-Experts Framework for Fine-Grained Vehicle Damage Detection and Semi-Automated Annotation

Kamching Cheng, Yongpoh Yu, and Tongming Lim
Tunku Abdul Rahman University of Management and Technology, Jalan Genting Kelang, Setapak, 53300, Kuala Lumpur, Malaysia

(Received 07 September 2025; Revised 28 November 2025; Accepted 14 January 2026; Published online 10 February 2026)

Abstract: Accurate detection of vehicle damages such as dents, scratches, and cracks is essential for improving the efficiency, consistency, and scalability of insurance claim assessment. Conventional inspection procedures rely heavily on manual evaluation, making them time-consuming, subjective, and costly. To address these limitations, this paper presents a three-stage progression of mixture-of-experts (MoE)–based classification models trained on the CarDD dataset, which contains 4,000 COCO-annotated vehicle damage images. The study begins with a baseline RawMoE model operating on flattened image representations, followed by ResMoE-50, which incorporates ResNet-50 for deep feature extraction. Building upon these foundations, we propose CropResMoE-50, a region-aware hybrid architecture that integrates object-level cropping with scale-specific analysis to enhance spatial localization and classification accuracy. Extensive experiments on the CarDD benchmark, together with comparisons against established baselines including ResNet-50, EfficientNet-B0, and Swin-T, demonstrate that CropResMoE-50 achieves strong performance with favorable computational efficiency (24.8M parameters, 4.13 GFLOPs, and approximately 0.006 s inference latency). The model attains 89.30% test accuracy and average precision scores of 87.70%, 93.45%, and 98.12% for small, medium, and large objects, respectively. To extend practical applicability, a semi-automated labeling pipeline integrating ChromaDB is introduced to support retrieval-augmented learning and pseudo-labeling under uncertainty. Additional validation on a real-world insurance dataset from a real-world insurance company confirms robust generalization, achieving 83.33% accuracy. Overall, the proposed framework offers a scalable, interpretable, and deployment-ready solution for automated vehicle damage assessment.

Keywords: ChromaDB; insurance claims; mixture of experts; ResNet-50; vehicle damage detection

I. INTRODUCTION

Manual inspection of vehicle damage remains a persistent bottleneck in automotive insurance workflows, often introducing delays, inconsistencies, and subjective evaluations. These shortcomings are exacerbated by the limited nature of existing datasets, which typically suffer from class imbalance and constrained perspectives and factors that undermine fine-grained recognition of damage types such as dents, scratches, and cracks. Although machine learning has been widely applied to object detection tasks, the integration of mixture of experts (MoE) with hybrid embedding strategies for specialized and imbalanced datasets remains underexplored.

The lack of robust multi-angle benchmarks further hinders the generalizability of current models in real-world scenarios. While the CarDD dataset provides a standardized basis for damage classification research, few studies have fully utilized its potential to assess MoE architectures in tandem with deep visual embeddings. Moreover, there is a critical gap in data augmentation strategies tailored to low-frequency classes, limiting detection accuracy for underrepresented damage types.

Conventional deep learning models, such as YOLO and ResNet, have shown promising results in generic object detection tasks [1,2]. However, these models tend to underperform when applied to domain-specific challenges like vehicle damage classification, particularly in the presence of small-object occlusion, low-contrast features, and rare class instances [3,4]. This limitation necessitates the design of adaptive architectures that can capture localized visual patterns and handle long-tail class distributions with greater precision [5,6].

To address these challenges, this research proposes a hierarchical and hybrid solution grounded in MoE architectures. Specifically, we introduce three progressively refined models: RawMoE, ResMoE-50, and CropResMoE-50. The RawMoE model serves as a foundational benchmark, operating directly on raw pixel inputs to explore MoE capabilities in isolation [7,8]. Building upon this, ResMoE-50 integrates a ResNet-50 backbone for feature extraction, enabling the model to learn from spatially rich, pretrained embeddings [9]. The most advanced version, CropResMoE-50, further enhances performance by incorporating object-cropped regions and size-aware analysis through bounding box metadata, facilitating fine-grained focus on damaged areas.

This paper's key contributions are threefold:

1. Application of MoE in Damage Detection: We demonstrate the suitability of MoE architectures in navigating the

Corresponding author: Kamching Cheng (e-mail: chengkam.ching@gmail.com).

complexity and imbalance of real-world car damage datasets, a direction not adequately explored in prior work [2,10].

2. Proposal of CropResMoE-50: By coupling a refined routing mechanism with object-level cropping, the model significantly boosts classification accuracy for challenging damage types, particularly small-size or obscured defects [11,12].
3. Deployment-Oriented Validation: We validate the model's real-world applicability by automating damage triage in insurance claim scenarios, effectively reducing the need for manual visual assessment [1,5,12,13]. Furthermore, we introduce a semi-automated labeling pipeline that integrates retrieval-augmented memory via ChromaDB, enabling confidence-aware pseudo-labeling which is a novel extension that bridges prediction with transparent decision support for low-certainty cases.

These contributions fill notable gaps in the literature. While previous research has evaluated MoE on large-scale datasets, its application to highly specialized classification tasks—particularly those involving rare or fine-grained classes—remains limited [3,5,6,14]. Similarly, few studies have addressed the implications of class imbalance using task-specific preprocessing or expert selection mechanisms.

Within this research framework, we pose two guiding questions: (1) under what conditions do MoE-based models surpass conventional deep networks in complex visual classification tasks? and (2) Can a hybrid architecture combining MoE and deep embeddings meaningfully improve detection performance in highly imbalanced, real-world datasets? To answer these, we design and evaluate hybrid MoE architectures tailored for automotive damage classification, setting a target performance benchmark of at least 85% test accuracy. In doing so, this study aims to bridge a critical research–practice divide by delivering scalable, deployable models for next-generation damage assessment systems.

The rest of the paper is organized as follows. Section II reviews related work. Section III outlines the methodology. Section IV details our three MoE models: RawMoE (baseline), ResMoE-50, and CropResMoE-50. Section V describes dataset preparation and localized region extraction strategy. Section VI presents the experimental setup with state-of-the-art baselines. Section VII analyzes results showing progressive model improvements and competitive performance. Section VIII discusses practical implications. Section IX introduces the semi-automated labeling pipeline with ChromaDB. Section X concludes the paper, and Section XI proposes future work.

II. RELATED WORK

Recent developments in MoE and model fusion strategies have demonstrated notable progress in addressing multi-modal and class-imbalanced classification challenges. For instance, [15] introduced a multi-task MoE-based fusion method that unifies diverse models within a cohesive structure, allowing domain-specific specialization without compromising generalization. Such flexible architectures have proven highly effective in tasks requiring precision across heterogeneous inputs [11,12,16–19].

One of the key datasets in this domain, the CarDD dataset introduced by Wang *et al.* [14], provides a foundational benchmark for vehicle damage detection research. With over 4,000 high-resolution images and more than 9,000 annotated instances across six damage categories, CarDD is the first publicly available dataset tailored for fine-grained vehicle damage classification and segmentation. Nevertheless, this methodology presents considerable challenges, notably class imbalance and scale variability, with these problems being most acute for less common defect categories, including dents, cracks, and scratches. [4,5,20,21, 21–26]. While CarDD enables classification benchmarking, most existing models prioritize bounding box accuracy over holistic workflows such as auto-labeling or semantic confidence filtering which are limitations that motivate the integration of retrieval-augmented tools like ChromaDB. These conditions necessitate specialized architectures that go beyond generic object detectors.

Traditional models like YOLO, U-Net, and Mask R-CNN have served as baseline solutions in this space. However, their fixed-capacity architectures struggle with allocating focus dynamically, especially when identifying fine-grained, small-sized damages in noisy or cluttered scenes [1,4,7,13,27]. Additionally, many detection-focused architectures prioritize bounding box precision over semantic clarity, often producing labels that are spatially correct but semantically ambiguous. This limitation is especially problematic for classes like "dent" or "paint scratch," where spatial cues alone may be insufficient. MoE architectures, on the other hand, are designed to allocate computational focus based on input characteristics through a dynamic routing mechanism. This selective activation enables the model to better handle heterogeneous data distributions and visually subtle variations across classes [6,10].

Expanding on this, [10] explored Gated MoE (GMoE) configurations with domain-specific constraints to improve model interpretability and modular task decomposition. This flexibility is particularly vital in insurance contexts, where trust and auditability of predictions play a crucial role in operational deployment [5,7,28]. Additionally, the adaptability of MoE becomes even more evident when extended to domains beyond vision, providing valuable design analogs for vehicle damage assessment systems. While GMoE models effectively showcase the interpretability and robustness of MoE architectures, they are designed for static or uniform data conditions, making them poorly suited for vehicle damage datasets with their inherently high intra-class variance. In contrast, vehicle damage datasets exhibit high intra-class variance, illumination noise, and overlapping object scales, requiring more dynamic and scale-aware MoE implementations. Existing literature rarely explores the impact of focal loss or SoftMax calibration within MoE gates, particularly when applied to long-tail distributions like cracks or scratches. This study empirically investigates how tuning these loss functions can reshape expert attention in imbalanced domains.

Indeed, recent studies on natural language inference (NLI) have introduced the mixture of prompt experts (MOPE) framework to dynamically handle prompt-based classification using an MoE-inspired structure [8,29–33]. These approaches show how expert specialization can increase sensitivity to domain-specific features, a principle that holds true in detecting subtle visual damage cues.

In real-time sorting systems, [11,12] applied MoE to integrate neural networks with Kalman filtering, enabling enhanced prediction accuracy in stochastic environments. This hybrid learning formulation supports the hypothesis that MoE architectures are particularly suited for dynamic, noisy, or high-variance datasets which is a condition similar to car damage detection in uncontrolled environments [1,2].

Few studies have examined how MoE-based vision pipelines respond under constraints such as partial occlusion, inconsistent lighting, or non-standard viewpoints, all of which are common in

real-world car inspection scenarios. While Mask R-CNN and YOLO offer spatial detection, they often lack class-level certainty estimates needed for downstream auto-labeling or confidence-based triage.

Moreover, hybrid MoE approaches in environmental sound classification have leveraged attention-based routing to improve expert selection. As shown in [30], selectively attending to discriminative temporal features allowed better generalization in noisy conditions. This method of conditional expert activation offers a viable direction for fine-tuning vision-based models to detect infrequent or ambiguous damage types like hairline cracks or light scratches [3,6,23,34].

Another critical factor in fine-grained classification is deep feature extraction. Architectures such as ResNet remain vital due to their residual connections and ability to learn robust representations, especially for localized features in small-object detection [5,7,9,35,36]. However, YOLO-based pipelines, while optimized for real-time processing, are less effective in detecting minor anomalies, given their coarse-grained bounding strategies [4,6,28,37–40].

In conclusion, the reviewed literature strongly advocates for hybrid, expert-routed models in domains characterized by class imbalance and spatial variability. This body of work forms the foundation upon which this study builds its MoE-ResNet hybrid pipeline, targeting both improved performance and scalability in real-world car damage classification applications. While prior work has explored MoE in various domains, including NLP and sensor fusion, few studies have applied MoE to vision tasks requiring fine-grained, localized attention within unstructured environments like vehicle damage. This highlights a critical gap that this study seeks to fill and not merely by applying MoE but by tailoring the expert routing and fusion logic to accommodate real-world insurance inspection constraints. This study distinguishes itself not only through architectural innovation but also by proposing a complete semi-automated annotation framework that bridges the gap between isolated classification models and deployable, explainable systems fit for industry-scale insurance workflows.

III. METHODOLOGY

This study follows the Analytics Solutions Unified Method for Data Mining (ASUM-DM) framework [41], which extends the classic CRISP-DM methodology [42–44] with enhancements for iterative delivery, model reuse, and integration with modern analytics platforms. ASUM-DM is structured into seven process groups: Analyze, Design, Configure & Build, Deploy, Operate & Optimize, and Project Management. These phases are adapted to guide the lifecycle of a deep learning-based vehicle damage classification system from planning through deployment.

Analyze Phase: The project commences with defining the business objective of automating fine-grained car damage detection for insurance workflows. The key challenges identified include class imbalance, small-object detection, and the need for scalable pseudo-labeling. A risk assessment is performed to ensure data privacy, especially given the inclusion of real-world claims photos.

Design Phase: Data understanding and feature design are driven by the Car Damage Dataset (CarDD), a public benchmark containing over 4,000 high-resolution images and six annotated damage types. To improve feature discrimination, the original COCO-format dataset is supplemented with manually cropped damage patches, organized into size classes: small, medium, and large. Preprocessing pipelines are developed using OpenCV and PIL to normalize image resolution (224×224) and prepare class-consistent crops. All images are converted to RGB, and augmentation (rotation, noise, and brightness adjustment) is applied to address class imbalance in particularly for underrepresented damage types like cracks and scratches.

Configure & Build Phase: Three MoE-based models are developed: (1) RawMoE: A baseline MoE model using raw pixel vectors as input. (2) ResMoE-50: Integrates ResNet-50 as a fixed feature extractor before the MoE gating layer. (3) CropResMoE-50: Builds on ResMoE-50 with a region-based input strategy, enabling scale-aware learning across size-diverse damage types. All models are trained using focal loss with various α and γ configurations to address label imbalance. Training and validation follow an 80/20 split, and performance was evaluated using accuracy, AP, and confusion matrices.

Deploy Phase: Although full-scale deployment is beyond the scope of this journal, a simulated deployment environment is established. This involved packaging the trained model using PyTorch and ONNX formats and integrating it with a semi-automated annotation pipeline that supports fallback retrieval via ChromaDB. Inference APIs were designed to process new crops and return class predictions alongside retrieval-supported confidence overlays.

Operate & Optimize Phase: System performance is continuously monitored during pseudo-labeling runs on unseen data batches. Outputs were compared against expert labels, and high-confidence predictions ($\tau > 0.6$) were retained for dataset expansion. Feedback loops are proposed to enable future integration of active learning mechanisms.

Project Management: The research employed iterative development cycles aligned with ASUM-DM phases, with regular supervisor consultations guiding progress and refinements. Key performance indicators included labeling accuracy, model throughput, and usability of the annotation interface.

IV. PROPOSED MODELS

This study proposes three deep learning models based on the MoE paradigm: RawMoE, ResMoE-50, and the enhanced CropResMoE-50. These models progressively integrate embedding techniques and spatial localization to improve the detection of car damage across varying sizes and categories. Table I presents a high-level comparison of the proposed models. The underlying hypothesis is that combining MoE with deep feature extractors and scale-aware preprocessing enables better specialization and discrimination, especially under class imbalance.

A. RAWMOE: FLATTENED IMAGE APPROACH

The RawMoE model (Fig. 1) uses an MoE architecture to classify vehicle damage based on images from the CarDD dataset. To reduce computational complexity, the images are resized to 64×64 pixels and then flattened into a 12,288-dimensional feature vector ($64 \times 64 \times 3$), sacrificing spatial locality in favor of computational simplicity. This design highlights the baseline behavior of MoE without spatial inductive bias, forming a contrast with the later models. The architecture of RawMoE consists of five experts, each with distinct components:

1). INPUT LAYER. The input to the MoE model is a flattened RGB image ($x \epsilon R^{12288}$), representing each pixel value of a resized

Table I. Comparison of the proposed models

Feature	RawMoE	ResMoE-50	CropResMoE-50
Input representation	Flattened pixel intensities (12,288)	Global features vectors (2048)	Localized feature vectors (2048)
Feature richness	Low (none)	High (entire image)	Very high (localized damage regions)
Learning complexity	Limited to raw data	Enhanced by hierarchical feature extraction	Focused on localized damage regions
Accuracy	Lower	Higher	Highest
Expert specialization	Less effective	More focused due to richer features	Highly specialized and aligned with real-world scenarios
Noise handling	None	Moderate	High: noise-free inputs

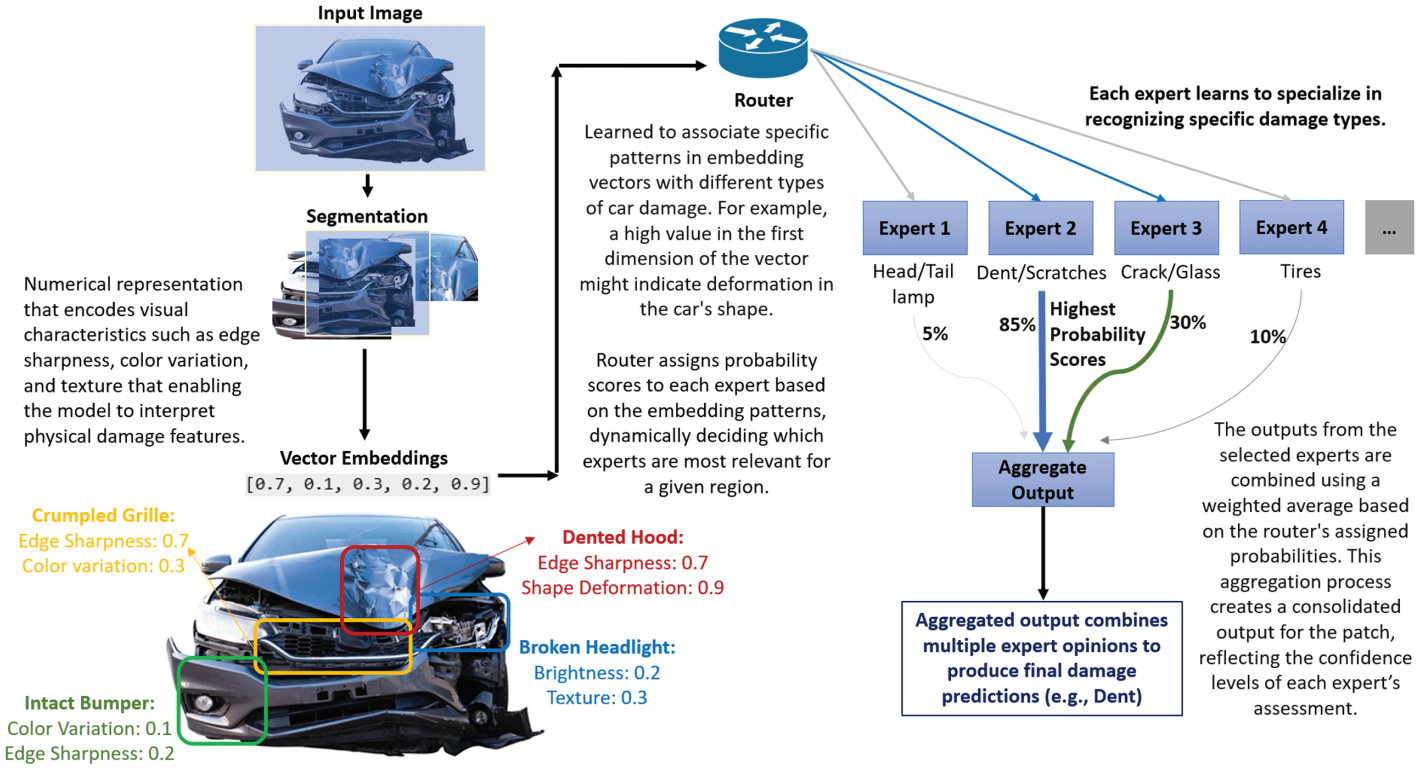

Fig. 1. Conceptual overview of MoE-based damage detection showing expert specialization.

64×64 image with three color channels (Red, Green, and Blue). Flattening the image allows us to convert the spatial pixel information into a single vector, making it suitable for feeding into a fully connected neural network. Hidden Layer (Expert Output): Each expert processes the input through a hidden layer, which can be mathematically represented as:

$$h_i = ReLU(W_{1,i} \cdot x + b_{1,i})$$

- x represents the input vector to the expert model where each element of x contains the pixel intensity value for one of the RGB color channels at a specific pixel location of the image.
- $W_{1,i}$ represents the weight matrix of learned parameters specific for expert i which transforms the input vector. It has a size that is compatible with the input vector.
- $b_{1,i}$ is the bias vector added to ensure flexibility in learning. Each expert in the MoE architecture has its own unique bias. It has the same dimensionality as the output of the hidden layer which is 128 units.
- ReLU (rectified linear unit) is used as the activation function to introduce nonlinearity. This helps the model capture more complex patterns in the data by retaining positive values and setting negative values to zero, using 128 units in this case.

2). OUTPUT LAYER (EXPERT PREDICTION). The output from each expert, denoted by y_i, is calculated using:

$$y_i = W_{2,i} \cdot h_i + b_{2,i}$$

- $W_{2,i}$ and $b_{2,i}$ are the weight and bias for the output layer of the expert i, respectively.

- The output y_i corresponds to the prediction made by the expert i for the given input, specifically predicting the class label for the vehicle damage categories (Dent, Scratch, Crack, Tire Flat, Lamp Broken, and Glass Shatter).

3). ROUTER (SOFTMAX-BASED ROUTING NETWORK). A crucial aspect of the MoE model is the routing network, which determines which expert or combination of experts should process a given input. The router assigns inputs based on learned weights through a SoftMax function:

$$g = softmax(W_r \cdot x + b_r)$$

- W_r and b_r are the weights and bias of the router, respectively.
- The SoftMax function converts the weighted input into a probability distribution over the experts, where each component g_i represents the probability assigned to expert i. These probabilities ensure that each expert contributes to the final prediction based on the relevance of their specialization to the given input.
- Here, r denotes the router or routing layer, which dynamically assigns probabilities for selecting experts in the MoE architecture. It implies that these weights and biases belong specifically to the router mechanism.

4). FINAL OUTPUT. The final output of the model is obtained by aggregating the predictions from all experts, weighted by the router's outputs:

$$y = \sum_{i=1}^{N} g_i y_i$$

- Here, N represents the total number of experts (which is five in this case).
- Each g_i is the probability assigned by the router to expert i, and y_i is the corresponding prediction by that expert.
- The final output y is a weighted sum of these predictions, providing the model's overall classification for the vehicle damage type.

The RawMoE model effectively uses an MoE to classify vehicle damage by distributing the workload across multiple specialized "experts." The router dynamically assigns the input to one or more experts, allowing the model to adapt to different types of damage by leveraging specialized processing pathways. This mechanism is particularly advantageous for complex tasks like vehicle damage detection, where different experts can focus on different aspects of the image, ultimately leading to a more robust and accurate prediction.

B. RESMOE-50: A HYBRID APPROACH

While RawMoE demonstrates baseline classification through simplified inputs, its limitation in capturing spatial hierarchies prompted the introduction of ResMoE-50. The ResMoE-50 model is a hybrid approach that combines the strengths of ResNet-50 and the MoE architecture. This model aims to address the limitations of raw pixel-based processing in the RawMoE model by utilizing deep feature extraction to provide a semantically richer and more abstract representation of vehicle damage. Specifically, ResNet-50, a state-of-the-art convolutional neural network (CNN), is used as the backbone for feature extraction, while the MoE module dynamically assigns these features to specialized experts for precise classification. This section explores the integration of ResNet-50 into the MoE framework and how this combination addresses key challenges in vehicle damage detection.

1). RESNET-50 AS FEATURE EXTRACTOR. The final fully connected layer of ResNet-50 is removed, and the penultimate layer outputs a 2048-dimensional embedding for each 224×224, ImageNet-normalized input image. These embeddings provide semantically rich representations of vehicle structures and surface textures, which are then routed through an MoE head.

The MoE module consists of multiple expert subnetworks and a gating router. The router operates on the 2048-dimensional feature vector and assigns input-dependent weights to each expert, enabling specialization for different damage characteristics (e.g., edges and scratches vs structural deformation). Compared to RawMoE, which operates directly on flattened pixels, ResMoE-50 leverages pretrained convolutional features to improve convergence stability, robustness to noise, and discriminative power for fine-grained damage classes.

2). FEATURE EXTRACTION IN RESMOE-50. In ResMoE-50, the final classification layer of ResNet-50 is removed to use its second-to-last layer as a feature extractor. This layer outputs a 2048-dimensional vector, which encapsulates high-level features summarizing the critical aspects of the input image.

The preprocessing phase involves essential transformations applied to the dataset images. Each image is resized to a standardized dimension of 224×224 pixels to ensure compatibility with the ResNet-50 architecture. This resizing is accompanied by normalization, which scales pixel values consistently across the dataset. These steps are critical for achieving efficient training dynamics, enabling stable gradient propagation, and improving the convergence behavior during neural network training.

During forward propagation, ResNet-50 employs a hierarchical feature extraction mechanism. The input image progresses through a network of convolutional layers and residual blocks, which collaboratively extract and refine features at multiple abstraction levels. The process begins with identifying basic visual elements, such as edges and textures, and advances to detecting complex patterns and shapes. This hierarchical feature extraction ultimately produces a compact yet information-rich feature vector that effectively represents the key characteristics of the input image.

Transfer learning plays a pivotal role in this approach. ResNet-50, which is a model that was pretrained on the ImageNet dataset, serves as a solid foundation with robust visual features. These pretrained features are adapted to meet the specific requirements of damage detection tasks, leveraging the network's prior knowledge to enhance performance. This strategy accelerates the training process and improves the model's accuracy, especially in ambiguous cases where label overlap or occlusion occurs, an advantage particularly valuable in complex insurance scenarios.

3). INTEGRATION WITH MOE. The MoE module in ResMoE-50 is tasked with interpreting the 2048-dimensional feature vector extracted by ResNet-50. It consists of multiple "experts," each specializing in certain patterns or damage types, such as dents or scratches. The router dynamically assigns input features to these experts based on their learned specialization.

The dynamic routing mechanism is an integral aspect of the MoE architecture, enabling adaptive decision-making in response to diverse inputs. At the heart of this mechanism lies the router,

which serves as a computational decision-maker (Fig. 2). Its primary function is to evaluate the input features extracted by ResNet-50 and determine the optimal allocation of these features to a set of specialized experts. By assigning probabilities to each expert, the router quantifies the suitability of each expert for processing the given input. This probabilistic allocation ensures that the network dynamically directs computational resources to the most relevant components, thereby optimizing task-specific performance.

Central to the routing process is the generation of the gating vector, which encapsulates the probabilistic contributions of each expert. The router computes this vector by analyzing the input features, with the constraints that all probabilities collectively sum to one. This ensures a weighted distribution of responsibility among the experts, aligning their contributions with the demands of the input. For instance, in scenarios where an expert exhibits a high degree of specialization in detecting scratches, and the input features strongly align with such a task, the router assigns a correspondingly higher probability to that expert. This dynamic weighting mechanism allows the model to prioritize relevant experts while minimizing redundancy, leading to efficient and targeted processing.

Following the generation of the gating vector, the input features are forwarded to all experts for parallel processing. Each expert operates independently, leveraging its distinct learned parameters to generate intermediate predictions based on its area of specialization. This framework enables the network to decompose complex inputs into manageable components, with each expert addressing specific aspects of the data. For example, an expert trained on identifying subtle surface irregularities focuses on fine-grained details, while another, specializing in structural damages, evaluates broader spatial patterns. This modular design ensures a comprehensive analysis of the input, even in scenarios characterized by high variability or ambiguity.

The final stage of the mechanism involves the aggregation of expert predictions, weighted according to the probabilities defined by the gating vector. Experts assigned higher probabilities exert greater influence on the aggregated output, resulting in a composite prediction that encapsulates the network's collective expertise. This dynamic and adaptive behavior enables the MoE architecture to effectively handle a wide range of input scenarios, from fine scratches to extensive structural damages. The approach not only enhances the model's predictive accuracy but also underscores its efficiency and flexibility. By leveraging the strengths of specialization, adaptability, and resource optimization, the dynamic routing mechanism positions the MoE architecture as a robust solution for complex tasks in damage detection and beyond.

ResMoE-50 offers several distinct advantages that enhance its effectiveness in vehicle damage detection. Leveraging ResNet-50, the model benefits from semantically rich feature representations, capturing both local and global patterns in damage images, which significantly outperforms RawMoE's reliance on raw pixel intensities. The MoE module further elevates performance through dynamic specialization, assigning specific experts based on the input features. For instance, experts focusing on fine-grained edge patterns handle small scratches, while those analyzing broader deformations address large dents. Combining ResNet-50's pre-trained weights provides a robust foundation, enabling faster convergence and improved generalization on the car damage dataset. These attributes collectively position ResMoE-50 as a highly adaptable and efficient solution for vehicle damage detection tasks.

C. CROPRESMOE-50: REFINEMENT WITH LOCALIZED REGION

To further enhance spatial specificity, particularly for fine-grained damages, we introduced CropResMoE-50, which localizes and

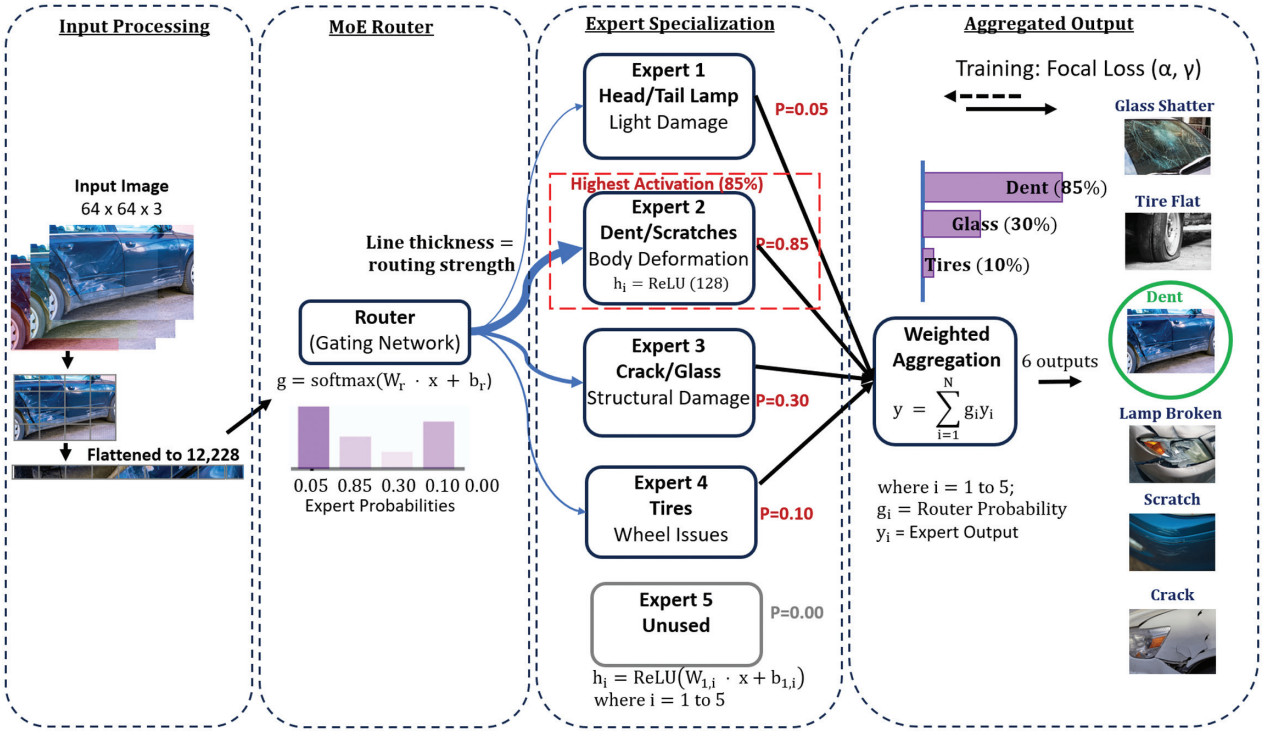

Fig. 2. RawMoE architecture with dynamic expert routing for vehicle damage classification.

isolates annotated regions before classification. The CropResMoE-50 model represents a pivotal advancement in vehicle damage detection by concentrating on localized damage regions. Unlike ResMoE-50, which processes entire images, CropResMoE-50 utilizes bounding box annotations from the dataset to extract regions of interest (ROIs) that correspond to specific damage instances. This targeted region-based approach enables a more refined and focused analysis by reducing the impact of irrelevant image areas, ensuring that the model prioritizes critical features directly associated with the damage.

By combining region-focused processing with ResNet-50's robust feature extraction and the dynamic specialization of the MoE architecture, CropResMoE-50 excels at detecting fine-grained damage types such as scratches, dents, and cracks. This approach also effectively mitigates challenges such as class imbalance and the detection of small or subtle damage, providing enhanced accuracy for nuanced damage scenarios.

Additionally, the model incorporates size-aware analysis to evaluate performance metrics like accuracy and precision across varying object scales. This ensures consistent and reliable detection, regardless of the size or complexity of the damage type. With its focus on localized analysis and specialized processing, CropResMoE-50 stands as a highly efficient and accurate solution for vehicle damage detection, addressing the practical demands of real-world applications. This approach also supports our semi-automated labeling pipeline by enabling ChromaDB-assisted confidence scoring on cropped regions, which serves as an efficient triage mechanism for downstream human validation.

1). DAMAGE REGION LOCALIZATION. Damage region localization is a critical preprocessing step in the CropResMoE-50 pipeline. Instead of relying on automated detection systems like YOLOv5, this study leverages pre-annotated bounding boxes embedded in the dataset to isolate specific areas corresponding to visible damage such as dents, scratches, or cracks. By extracting these predefined regions, the model avoids interference from irrelevant background content, thereby enhancing its ability to detect and classify fine-grained damage types with greater precision and consistency.

2). BOUNDING BOX ANNOTATIONS AND DEFINITION. Bounding box annotations are an essential component of the car damage dataset, which represent the spatial regions within each image that correspond to specific damage types, such as dents, scratches, or cracks. These annotations enable precise isolation of relevant regions, facilitating targeted analysis. A bounding box is defined by four critical parameters:

- x_{min}: The horizontal coordinate of the top-left corner of the bounding box.
- y_{min} : The vertical coordinate of the top-left corner of the bounding box.
- $Width(w)$: The horizontal span of the bounding box.
- $Height(h)$: The vertical span of the bounding box.

A bounding box is expressed as: $Bounding\,Box = (x_{min}, y_{min}, w, h)$. The range of the bounding box is governed by the conditions:

$$x_{min} \leq x \leq x_{min} + w \quad \text{AND} \quad y_{min} \leq y \leq y_{min} + h$$

This detailed marking of regions is foundational for isolating specific damage types during preprocessing. For example, a bounding box acts as a virtual rectangle around a damaged area, helping the model focus exclusively on this region without being distracted by unrelated parts of the image, like the sky or license plate. These annotations are carefully created, often through a combination of expert input and automated tools, ensuring that all damage instances are accurately marked for further analysis.

3). DAMAGE REGION EXTRACTION AND PREPROCESSING. Once the bounding boxes are defined, the next step is to extract the ROIs from the images. This ensures that the model processes only the areas containing damage, improving efficiency and accuracy. The preprocessing pipeline ensures consistency and compatibility with the ResNet-50 feature extractor. This process includes the following steps: Region Selection, Resizing, and Normalization.

Region Selection: For a given input image I with dimensions $H \times W$, the corresponding bounding box defines the pixel range that isolates the damage region. The cropped region is expressed as $I_{crop} = I[y_{min} : y_{min} + h, x_{min} : x_{min} + w]$, where I_{crop} represents the sub-image containing the damage. Using these bounding box coordinates, the specified portion of the image is extracted, akin to digitally cutting out a rectangle that encapsulates the damaged area. This step ensures that the analysis is focused solely on the relevant region, eliminating distractions from unrelated parts of the image.

Resizing: Once cropped, the extracted region is resized to a uniform dimension of 224×224 pixels to align with the input requirements of ResNet-50. The resizing process, expressed as $I_{resized} = resize(I_{crop}, 224 \times 224)$, standardizes the input dimensions while preserving critical features of the damage. This ensures that the visual details necessary for accurate damage detection are retained while enabling consistent processing across all samples. By maintaining this uniformity, the model operates efficiently and effectively regardless of the original size or scale of the cropped region.

Normalization: After resizing, the image undergoes normalization to conform to the input expectations of the pretrained ResNet-50 model. This involves two primary steps: first, scaling the pixel values to the range [0, 1], and second, standardizing these values using the mean (μ) and standard deviation (σ) of the ImageNet RGB channels. The normalized image is computed as $I_{norm} = \frac{I_{resized} - \mu}{\sigma}$. This normalization aligns the input data with the statistical properties of the data used during ResNet-50's pretraining, optimizing the model's ability to process the images effectively and ensuring consistency across all inputs.

In some cases, multiple damages like a scratch and a dent might appear close to each other, causing their bounding boxes to overlap. Instead of combining these areas, each bounding box is processed independently. This ensures that the model can analyze and classify each type of damage separately, maintaining a clear distinction between different damage categories.

The process of isolating damage regions follows a structured pipeline to ensure consistency and focus. First, the bounding box is used to identify and localize the exact damaged area within the image. Next, the corresponding region is extracted using the bounding box coordinates, effectively isolating the relevant portion of the image. Finally, the extracted region undergoes preprocessing, which includes resizing to a standard dimension and normalization to ensure compatibility with the model's input requirements. This systematic workflow ensures that the model concentrates exclusively on the damage region, minimizing distractions from irrelevant background elements and optimizing the accuracy of the analysis.

4). DAMAGE REGION CATEGORIZATION. Bounding box areas are classified into categories based on the COCO dataset standard,

facilitating performance evaluation across various object scales. This categorization recognizes that damages differ significantly in size, ranging from fine scratches to large dents or cracks. To address this variability, damage regions are divided into three classes: *Small Objects* (area $<128^2$), which include minimal damages like fine scratches; *Medium Objects* ($128^2 \leq$ area$< 256^2$), covering moderate damages such as small dents; and *Large Objects* (area $\geq 256^2$), which represent extensive damages like shattered windshields. The area of a bounding box is calculated as $Area = w \times h$. This classification ensures that the model balances its focus across damages of different scales, preventing smaller damages from being overlooked during analysis.

Localizing damage regions offers distinct advantages that significantly enhance the model's performance and efficiency. By isolating specific areas of interest, the model prioritizes relevant features, improving its sensitivity to subtle patterns and fine-grained details. This focused approach reduces background noise by excluding unrelated regions, such as vehicle surroundings, resulting in cleaner and more meaningful inputs. Moreover, analyzing smaller, localized regions reduces computational complexity, enabling efficient processing, especially when the damage occupies only a minor portion of the image. These combined benefits strengthen the model's capability to deliver accurate and reliable damage detection.

Localized damage regions are the foundation of the CropResMoE-50 model, serving as the primary input to its ResNet-50 feature extractor. By processing only these preselected regions, the model achieves a higher precision in detecting and classifying a variety of damage types. This specialized focus distinguishes CropResMoE-50 from conventional models, allowing it to excel in fine-grained vehicle damage detection. The localization process is more than a preprocessing step—it forms the core of the CropResMoE-50 pipeline. Through systematic isolation, standardization, and processing of the damage regions, this approach ensures consistent and accurate outputs, positioning the model as an effective tool for real-world vehicle damage assessment. Empirical evaluation of these models, detailed in Section VI, demonstrates that CropResMoE-50 outperforms others on fine-grained categories while maintaining efficiency.

D. ARCHITECTURE SUMMARY

For baseline comparison, the ResNet-50 backbone comprises four residual stages ($3 + 4 + 6 + 3$ bottleneck blocks) totaling 23.5 million parameters. The EfficientNet-B0 backbone consists of nine sequential stages built with MBConv (Mobile Inverted Bottleneck) blocks using variable expansion ratios, totaling 4.0 million parameters. The Swin-Transformer Tiny (Swin-T) backbone features four hierarchical stages containing shifted-window attention blocks arranged as $2 + 2 + 6 + 2$ across stages, amounting to 27.5 million parameters.

In contrast, the proposed CropResMoE-50 integrates a ResNet-50 feature extractor ($3 + 4 + 6 + 3$ configuration) with an MoE classification head consisting of a router network and five expert subnetworks, each containing two fully connected layers ($2048 \rightarrow 128 \rightarrow 6$) with ReLU activation. The complete architecture comprises approximately 24.8 million parameters, balancing accuracy, efficiency, and interpretability within a unified design.

While deeper variants such as ResNet-101 or ResNeXt could, in theory, improve representational depth, they introduce nearly double the parameters and computational cost of ResNet-50 without proportionate accuracy gain ($\approx$ 44–48 M vs 23.5 M in ResNet-50), making ResNet-50 the most balanced and interpretable choice for the proposed MoE integration.

V. DATASET AND PREPROCESSING STRATEGY

The development of robust damage detection systems relies heavily on the availability of domain-specific datasets, as well as rigorous preprocessing strategies that enhance model generalization in real-world contexts. This study integrates both standardized and proprietary data sources to form a cohesive training pipeline that mirrors actual insurance inspection scenarios. The combined dataset comprises three main components: the publicly available CarDD dataset, a curated collection of manually cropped sub-images, and a batch of unlabeled images derived from real-world insurance claims. Each component plays a distinct role in supporting both the supervised learning and semi-automated labeling objectives of this research.

A. BENCHMARK DATASET: CarDD

CarDD, introduced by Wang et al., remains the most comprehensive benchmark available for vehicle damage classification. Comprising over 4,000 high-resolution images with more than 9,000 bounding box annotations, it spans six distinct damage categories: dent, scratch, crack, glass shatter, paint off, and tire flat. All annotations adhere to the COCO format, facilitating compatibility with object detection frameworks and enabling detailed spatial analysis.

To ensure a robust experimental setup, the CarDD dataset was partitioned into training (70.4%), validation (20.25%), and testing (9.35%) subsets. Stratified sampling preserved the distribution of damage categories across these splits. Additionally, image deduplication was conducted using perceptual hashing to remove near-identical samples, thus mitigating data leakage and overfitting risks.

To evaluate performance across varying object scales, the study adopted COCO-style average precision (AP) metrics for small, medium, and large damage regions. This allowed for granular insights into the model's detection capabilities, particularly in challenging scenarios involving minor or fine-grained damages.

B. REGION-CENTRIC CROPPING FROM ANNOTATED BOUNDING BOXES

To improve the model's sensitivity to localized damage patterns, a region-centric cropping strategy was employed using the existing COCO-format bounding box annotations from the CarDD dataset. Rather than relying on an external object detection algorithm, annotated regions were extracted directly based on the dataset's ground truth. Each bounding box was used to crop the damage-specific region, and the resulting patches were manually reviewed to ensure quality and relevance.

These cropped instances, typically resized to 224×224 pixels, served as a supplementary dataset alongside the original full-resolution images. This dual representation allowed the proposed CropResMoE-50 model to learn both global context and localized visual semantics, particularly important for detecting subtle damage types like cracks or scratches that may be visually insignificant in the broader image frame.

By focusing on ground-truth-aligned regions, this approach ensures that the expert subnetworks within the MoE architecture are trained on semantically meaningful content, reducing background noise and improving convergence stability.

C. REAL-WORLD INSURANCE CLAIMS FOR AUTO-LABELING EVALUATION

To assess the model's applicability beyond the benchmark dataset, a batch of unlabeled images was sourced from internal insurance claims. These images simulate operational scenarios in which photos are submitted without annotations, representing a typical cold-start problem in data pipelines.

For these cases, the trained CropResMoE-50 model served as the primary classifier, with fallback support via ChromaDB-based retrieval when confidence scores fell below a defined threshold ($\tau = 0.6$). The retrieval mechanism, built using HNSW-based nearest neighbor search, leverages the previously cropped CarDD instances as a support database. Retrieved labels are aggregated via distance-weighted voting and blended with the MoE outputs to produce final predictions. This semi-automated approach enables reliable soft-label generation for underrepresented or ambiguous images without manual intervention.

By integrating real claims data in this way, the system demonstrates practical readiness for deployment while also supporting incremental dataset expansion through low-touch, human-in-the-loop verification workflows.

D. SUMMARY OF STRATEGY ALIGNMENT

The combination of high-resolution annotations from CarDD, damage-focused region crops, and unlabeled field data creates a multi-faceted dataset that mirrors operational constraints and class imbalances. More importantly, it enables comprehensive model evaluation under both supervised and semi-supervised conditions. This preprocessing pipeline directly supports the design objectives of the CropResMoE-50 model, ensuring that the expert subnetworks are trained not only on diverse features but also on deployment-aligned data regimes.

VI. EXPERIMENT SETUP

A. IMPLEMENTATION ENVIRONMENT

All experiments were conducted using PyTorch 2.1 and Torchvision 0.17 on an NVIDIA Tesla T4 GPU (16 GB VRAM) via Google Colab. The AdamW optimizer (learning rate $= 2 \times 10^{-1}$, weight decay $= 1 \times 10^{-4}$) was employed with a batch size of 64 for 20 epochs. Inference latency was measured as the mean forward-pass time per image (averaged over 10 batches) under mixed-precision mode. This consistent hardware and software environment ensures reproducibility across all baselines and model variants.

B. BASELINE CONFIGURATIONS

To benchmark the proposed CropResMoE-50 model, we implemented three state-of-the-art architectures as baselines: ResNet-50, EfficientNet-B0, and Swin-T. For each baseline, we froze the ImageNet-pretrained backbone and trained only the classification head on the CarDD dataset. This frozen-backbone approach ensures a fair comparison by evaluating each model's learned representations rather than its fine-tuning capacity, essentially testing only how well each model understands features, not how well it can be trained. To maintain experimental consistency, we applied identical hyperparameters across all models: optimizer, learning rate, batch size, and training epochs.

C. TRAINING STRATEGY (FROZEN VS UNFROZEN)

Two training configurations were explored: (a) frozen-backbone mode—the feature extraction layers were fixed while only the classification head was optimized; (b) unfrozen fine-tuning mode with all layers jointly trained to enable domain adaptation and deeper feature learning. The proposed CropResMoE-50 was evaluated under both settings to analyze the trade-off between accuracy, generalization, and computational efficiency. This dual-mode design also establishes a direct basis for comparison with conventional CNN and transformer baselines.

D. COMPUTATIONAL-COST METRICS

Model efficiency was quantified by three computational-cost indicators: parameter count (M), floating-point operations (GFLOPs) per 224×224 input, and mean inference latency (seconds per image). These values were measured using ptflops and runtime profiling under identical hardware conditions. The quantitative results of both frozen and unfrozen configurations are presented and analyzed in Section VII (Results and Discussion).

VII. RESULTS AND DISCUSSION

Following the experimental setup described in Section VI, this section reports the results for the proposed RawMoE, ResMoE-50, and CropResMoE-50 models, together with external baselines and focal loss sensitivity analysis.

A. ABLATION STUDY: PROGRESSIVE MODEL ENHANCEMENTS

To quantify the contribution of each architectural component, an ablation study is performed across three progressively refined variants: RawMoE, ResMoE-50, and CropResMoE-50. RawMoE serves as the baseline MoE classifier operating directly on flattened pixel inputs without deep feature extraction. ResMoE-50 introduces a ResNet-50 backbone as a 2048-dimensional feature extractor before the MoE head, isolating the effect of pretrained hierarchical features. CropResMoE-50 further incorporates region-aware cropping based on ground-truth bounding boxes, enabling localized and scale-aware analysis of damage regions. The stepwise comparison across these three models explicitly measures how deep feature extraction and region-focused inputs contribute to performance gains.

B. EXECUTION TIME ANALYSIS

This ablation confirms that each architectural refinement from RawMoE to ResMoE-50 to CropResMoE-50 has produced consistent improvements in execution efficiency and predictive performance (Table II and Fig. 3), highlighting the necessity of both deep feature extraction and region-aware cropping in the final design. The comparative analysis evaluates these models across multiple performance metrics, underscoring the trade-offs and gains achieved at each stage of development.

Table II. Execution time analysis for the proposed models

Step	RawMoE (seconds)	ResMoE (seconds)	CropResMoE (seconds)
Mount google drive	2.59	4.55	2.32–68.12
Load ResNet-50 model	-	2.69	0.61–2.47
Load datasets	1.44	1.09	0.62–2.64
Feature extraction	105.83	978.35	1068.04–1933.46
Train model	1236.85	30.72	7.66–11.54
Validation evaluation	0.23	0.04	0.37–0.69
Test evaluation	0.11	0.02	0.36–0.58
Display images	-	75.45	78.3 −79.91
Total exec. time	**~1422**	**~1094**	**~1158–2027**

The feature extraction stage represents a critical juncture in model complexity. RawMoE's relatively low feature extraction time (105.83 s) stems from its shallow representation pipeline, offering faster computation at the expense of semantic depth. From a theoretical lens, this simplicity limits its ability to abstract complex visual patterns, particularly problematic in fine-grained tasks such as vehicle damage classification. ResMoE's significant increase (978.35 s) corresponds to its integration of ResNet-50, which reflects the principle of transfer learning as feature regularization and improving representation robustness at the cost of inference time. CropResMoE-50, extending this further, applies region-based masking and analysis (1068.04–1933.46 s), aligning with the divide-and-specialize paradigm. The computational burden here is justified, as the model mimics human-like attention, prioritizing localized cues which is an essential strategy in fine-grained classification literature.

The training time trajectory reveals important convergence patterns. RawMoE's excessive training duration (1236.85 s) highlights the inefficiency of working with raw input features lacking representational structure. ResMoE dramatically cuts this to 30.72 s by leveraging feature reuse and reducing parameter optimization load, based on key principles from pretrained initialization theory. CropResMoE-50 further slashes this to just 7.66–11.54 s, suggesting that region-based decomposition not only enhances generalization but also speeds up convergence by minimizing gradient noise, aligning with recent findings on modular architectures and attention pruning techniques.

In terms of total execution time, the results highlight the common trade-off between computational cost and model accuracy. RawMoE takes the longest time (1422.32 s) because it relies on basic processing and lacks optimization. ResMoE performs better, completing all tasks in 1094.43 s, thanks to its use of a pretrained network (ResNet-50) and more efficient architecture. CropResMoE-50 has the widest time range (1158.00–2027.02 s) due to its region-based processing, which increases time but improves accuracy. From a research perspective, this model reflects an "adaptive computation" approach—where the system uses more resources only when needed, depending on the complexity of each image. This idea aligns with MoE and recent studies on dynamic inference, where models are designed to adjust their effort based on task difficulty, improving both performance and efficiency over time.

The shift from RawMoE to CropResMoE-50 isn't just about improving speed or accuracy, but it shows how the model architecture evolves to better match the actual task. Each version becomes more focused and smarter in how it uses computation, processing only what's important instead of treating all input equally. This approach reflects current research in efficient deep learning, where models are designed to be more selective and specialized. By focusing on relevant regions, especially for fine-grained damage detection, CropResMoE-50 brings us closer to real-world solutions that balance accuracy, speed, and resource use in a practical way. To further contextualize these runtime patterns, the following section compares the proposed model with several external baselines to assess accuracy–efficiency trade-offs.

C. ACCURACY, PRECISION, AND RECALL COMPARISON

These results extend the ablation study beyond runtime, showing how each added component, for instance, ResNet-50 features and region-aware cropping, translates into measurable gains in accuracy, precision, recall, and F1-score. The comparison of validation and test results shows a clear and consistent improvement as the model evolves from RawMoE to ResMoE-50 and then to CropResMoE-50. As shown in Fig. 3, test set metrics demonstrate a stepwise performance gain from RawMoE to ResMoE-50, with CropResMoE-50 achieving the highest overall scores across accuracy, precision, recall, and F1-score. On the validation set, CropResMoE-50 achieves 88.15% accuracy, which is a + 41.24% improvement over RawMoE. Precision, recall, and F1-score also increase significantly which are +59.52%, +59.00%, and +60.58%, respectively. The test set tells a similar story. CropResMoE-50 reaches 89.30% accuracy, outperforming RawMoE by

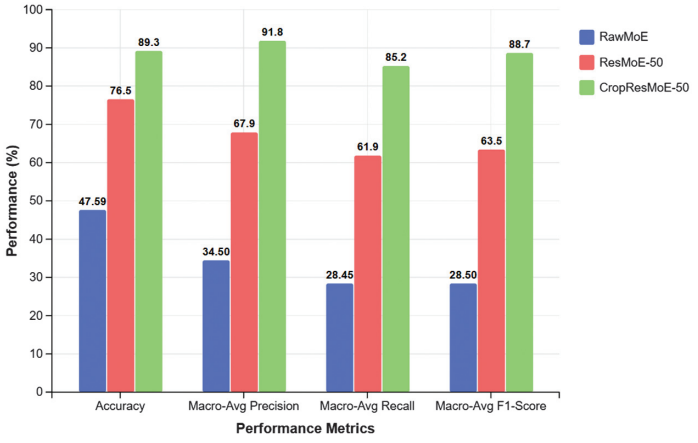

Fig. 3. Test set evaluation metrics of the proposed models RawMoE, ResMoE-50, and CropResMoE-50.

+41.71% and again showing strong gains in macro-average precision (+57.33%), recall (+56.75%), and F1-score (+60.15%).

These results show that each model upgrade not only improves performance during training but also leads to better generalization on unseen data. CropResMoE-50's strong metrics across both validation and test sets suggest that its region-focused architecture helps the model detect subtle damage patterns more effectively. This aligns with current research on task-specific feature specialization and supports the use of region-aware processing in real-world insurance and automotive damage detection systems. Collectively, the ablation across RawMoE, ResMoE-50, and CropResMoE-50 demonstrates that both deep feature extraction and localized region modeling are indispensable to achieving the final performance of CropResMoE-50.

A deeper look at each damage category shows just how much CropResMoE-50 has improved in handling specific, real-world damage types. For high-impact categories like glass shatter and tire flat, CropResMoE-50 reaches near-perfect performance with AP of 1.000 and 0.997, and consistently high precision and recall scores, highlighting its strong reliability in detecting critical safety issues. This category-wise trend is visualized in Fig. 4, where object detection metrics per damage class reveal stark contrasts between models, particularly for crack and lamp-broken categories.

In contrast, RawMoE struggles in several categories, especially for rare or complex damage types like cracks and broken lamps, where it records zero AP and precision, indicating a complete failure in detection. CropResMoE-50 overcomes this with +27.30% AP for crack detection and a major gain of +72.30% AP for broken lamp detection, along with 100% precision in both cases. These improvements are not just statistical as they reflect real advancements in the model's ability to recognize low-frequency, high-complexity patterns that often challenge conventional approaches. The percentage improvement from RawMoE to ResMoE-50 and CropResMoE-50 is further summarized in Fig. 5, highlighting model evolution in a cumulative manner.

From a research perspective, this shows the strength of combining region-focused processing with expert specialization. CropResMoE-50 effectively allocates more attention to local damage features, enabling it to outperform both RawMoE and ResMoE, especially in categories where subtle visual cues matter most. This aligns with current research in fine-grained classification and reinforces the model's suitability for real-world damage assessment systems, where achieving high accuracy across all categories, including rare and subtle cases, is essential for reliable and effective application. While these internal comparisons confirm progressive improvements, additional evaluation with external architectures is necessary to assess model competitiveness and scalability.

D. CROPRESMOE-50 ACCURACY AND PERFORMANCE ANALYSIS

The CropResMoE-50 model combines region cropping, ResNet-50 feature extraction, and a hybrid MoE structure. To handle class imbalance and emphasize harder examples, the model uses the focal loss function with tunable parameters α (alpha) and γ (gamma). This section evaluates how different α–γ configurations affect the model's AP, scale-specific performance (small, medium, large), and detection consistency.

Among all tested settings, the configuration $\alpha = 0.5$ and $\gamma = 2.0$ achieved the highest overall AP of 0.9352, along with strong APs across small (0.8725), medium (0.9567), and large (0.9763) objects. The detailed impact of focal loss configurations on object-size-based AP metrics is presented in Table III, covering small, medium, and large categories. This configuration strikes an effective balance: $\alpha = 0.5$ keeps a moderate focus on misclassified samples, while $\gamma = 2.0$ adjusts the gradient contribution from easier predictions without overwhelming the loss with outliers. The result is a model that generalizes well across object scales and maintains stable learning across diverse damage patterns.

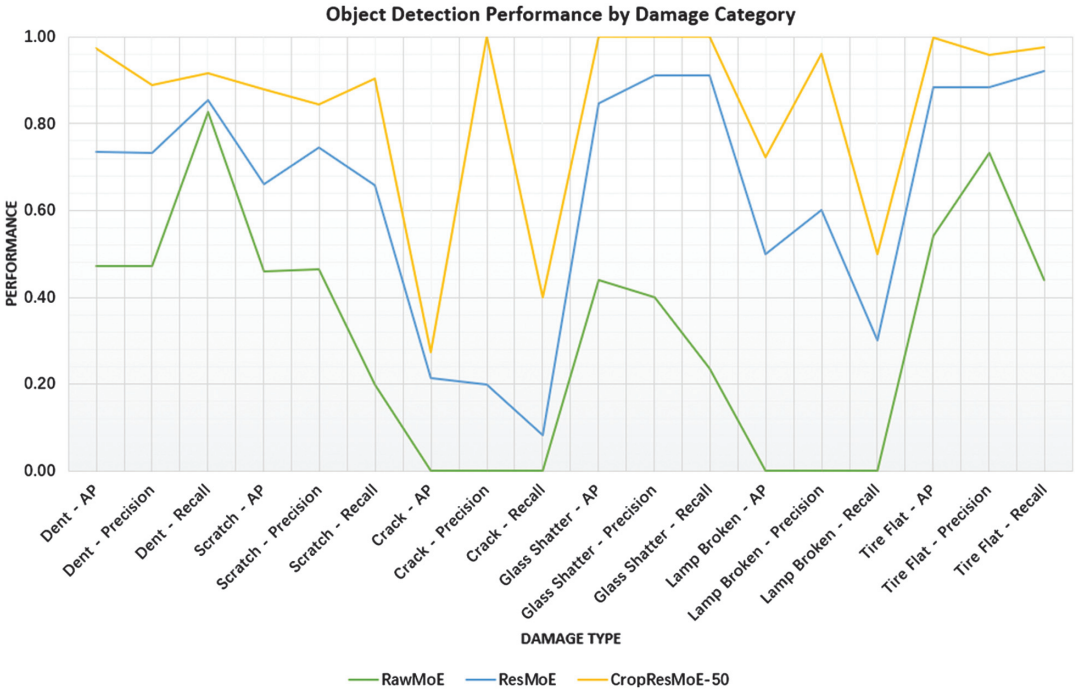

Fig. 4. Object detection performance by category.

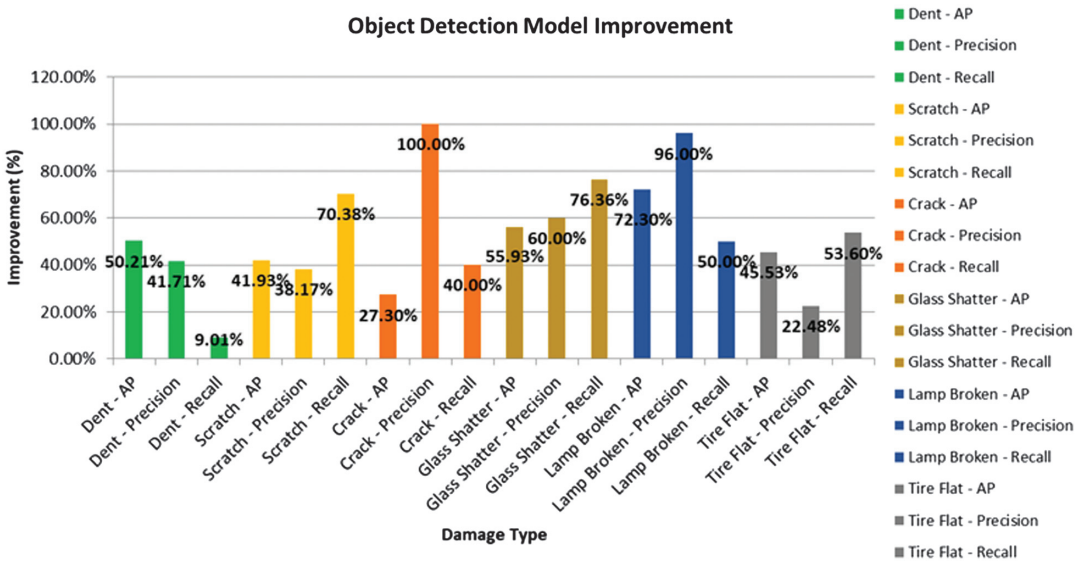

Fig. 5. Improvement in AP, precision, recall transition from RawMoE, ResMoE-50 to CropResMoE-50.

Table III. An accuracy comparison of average precision for different object size measurement (small, medium, and large) using the focal loss function parameterized by **α** and **γ**

α	γ	AP	AP_s	AP_m	AP_l
0.75	2	0.9225	0.8825	0.9148	0.9704
0.75	1	0.9308	0.8729	0.9377	0.9818
0.75	0.5	0.9329	0.8840	0.9384	0.9764
0.5	3	0.8948	0.8235	0.9069	0.9540
0.5	2	0.9352	0.8725	0.9567	0.9763
0.5	1	0.9255	0.8754	0.9277	0.9733
0.25	5	0.9204	0.8501	0.9288	0.9824
0.25	3	0.9201	0.8753	0.9213	0.9637
0.25	2	0.9279	0.8694	0.9383	0.9762
0.25	1	0.9272	0.8810	0.9253	0.9752

For small-scale objects, the best APs of 0.8840 are observed at $\alpha = 0.75$, $\gamma = 0.5$, showing that lower γ values can also benefit smaller object detection when α is higher. However, $\alpha = 0.5$, $\gamma = 2.0$ remains competitive, making it a more balanced choice across all sizes. In the medium-scale and large-scale categories, this same configuration also performs best with $AP_m = 0.9567$ and $AP_l = 0.9763$, confirming its reliability across varying spatial footprints.

These findings validate that focal loss tuning can be an effective lever to control model sensitivity across easy and difficult examples. The observed consistency in performance across different scales supports the idea that focal loss not only addresses class imbalance but also contributes to multi-scale robustness, which is critical in practical damage detection systems where object size varies widely.

The impact of focal loss parameters α and γ on CropResMoE-50's performance is evident across multiple metrics—AP, precision, recall, execution time, and category-specific accuracy. Overall, the findings show that while higher γ values (such as 3.0 or 5.0) can improve recall—especially for harder-to-detect or small-scale damages as they also lead to longer execution times and reduced precision, which disrupts the model's balance.

For instance, $\alpha = 0.25$, $\gamma = 5.0$ achieves macro-average recall of 0.86 and high dent recall (0.91) and lamp broken recall (1.00), but its macro-average precision drops and total execution time rises to 1448.51 seconds. Similarly, $\alpha = 0.50$, $\gamma = 3.0$, although strong in recall (0.85) and small-object AP (0.91), incurs the highest total execution time at 2027.02 seconds, making it less practical for time-sensitive applications.

In contrast, $\alpha = 0.25$, $\gamma = 1.0$ stands out as the most balanced configuration. It consistently achieves strong performance across validation and test datasets—macro-average precision and recall both at 0.87 and a strong overall AP of 0.93, while maintaining a low total runtime of 1167.86 seconds. It also performs reliably across all object sizes (APs = 0.88, APm = 0.93, APl = 0.98) and across multiple damage types, with precision and recall scores above 0.85 for nearly all categories.

From the research perspective, this confirms that $\alpha = 0.25$, $\gamma = 1.0$ offers the best trade-off between detection performance and computational efficiency. The lower α reduces the overemphasis on rare misclassifications, while $\gamma = 1.0$ maintains proportional gradient scaling, preserving learning stability. This balance avoids overfitting to extreme cases and allows the model to perform robustly across both frequent and rare damage categories.

Furthermore, execution time profiling (Table IV) confirms that feature extraction remains the most time-consuming step, varying widely from 1035.98 s to 2850.53 s, depending on α and γ. Training time, by contrast, is relatively stable, with the shortest training loop at 7.03 s under $\alpha = 0.75$, $\gamma = 2.0$, and only slightly longer (8.28s) for the best configuration $\alpha = 0.25$, $\gamma = 1.0$.

In summary, $\alpha = 0.25$, $\gamma = 1.0$ delivers the most efficient and consistent performance for real-world vehicle damage detection, with balanced accuracy, minimal latency, and strong generalization across scales and categories, making it the recommended default for practical deployment.

The integration of focal loss within CropResMoE-50 plays a central role in addressing class imbalance and improving generalization across damage categories. Focal loss, defined by its parameters α (class weighting) and γ (focusing factor), modulates the contribution of each sample to the loss gradient, effectively down-weighting well-classified examples while amplifying the influence

Table IV. Execution profiling with six damages category categorized by focal loss α and γ

Execution steps and time (in seconds)	Mount google drive	Load ResNet-50 model	Load datasets	Feature extraction	Remap labels and define output size	Define mixture of experts model	Training loop	Validation evaluation	Test evaluation	Calculate AP and AR for test set	Display test images	Total execution time
$\alpha = 0.75$ & $\gamma = 2.0$ (in seconds)	1.47	1.25	0.58	1035.98	0.00	0.02	7.03	0.29	0.27	0.02	53.51	1098.95
$\alpha = 0.75$ & $\gamma = 1.0$ (in seconds)	11.14	2.82	3.40	2237.59	0.02	0.02	7.85	0.53	0.36	0.02	53.25	2305.87
$\alpha = 0.75$ & $\gamma = 0.5$ (in seconds)	10.13	2.66	3.58	2352.69	0.02	0.01	7.68	0.38	0.24	0.01	50.84	2418.12
$\alpha = 0.50$ & $\gamma = 3.0$ (in seconds)	1.42	1.30	0.59	1080.60	0.01	0.03	7.09	0.85	0.29	0.02	54.79	1145.57
$\alpha = 0.50$ & $\gamma = 2.0$ (in seconds)	22.16	3.13	3.03	2850.53	0.02	0.03	11.41	0.55	1.18	0.02	64.62	2934.53
$\alpha = 0.50$ & $\gamma = 1.0$ (in seconds)	2.91	4.23	2.78	1348.94	0.00	0.02	11.31	0.31	0.79	0.02	64.84	1433.23
$\alpha = 0.25$ & $\gamma = 5.0$ (in seconds)	2.89	1.61	1.06	1351.75	0.01	0.02	11.28	0.45	0.32	0.03	81.97	1448.51
$\alpha = 0.25$ & $\gamma = 3.0$ (in seconds)	1.81	3.20	2.83	1349.04	0.00	0.02	10.38	0.60	0.43	0.02	66.27	1432.83
$\alpha = 0.25$ & $\gamma = 2.0$ (in seconds)	21.68	1.71	2.97	2754.35	0.03	0.04	8.82	0.61	1.10	0.02	51.62	2821.28
$\alpha = 0.25$ & $\gamma = 1.0$ (in seconds)	2.92	1.79	0.61	1103.66	0	0.02	8.28	0.41	0.4	0.02	52.67	1167.86

Table V. Performance average precision (AP) with six damages category categorized by focal loss α and γ

α	γ	Dent AP	Scratch AP	Crack AP	Glass shatter AP	Lamp broken AP	Tire-flat AP	Overall AP	AP_s	AP_m	AP_l
0.75	2.00	0.9684	0.8556	0.2157	1.0000	0.7226	0.9819	0.9225	0.8825	0.9148	0.9704
0.75	1.00	0.9776	0.8764	0.2687	1.0000	0.7663	1.0000	0.9308	0.8729	0.9377	0.9818
0.75	0.50	0.9724	0.8966	0.1946	1.0000	0.7278	0.9174	0.9329	0.8840	0.9384	0.9764
0.50	3.00	0.9405	0.8475	0.2854	1.0000	0.7052	1.0000	0.8948	0.8235	0.9069	0.9540
0.50	2.00	0.9784	0.8992	0.2217	1.0000	0.6944	0.9793	0.9352	0.8725	0.9567	0.9763
0.50	1.00	0.9722	0.8635	0.2296	1.0000	0.6259	0.9567	0.9255	0.8754	0.9277	0.9733
0.25	5.00	0.9666	0.8826	0.3889	1.0000	0.7595	1.0000	0.9204	0.8501	0.9288	0.9824
0.25	3.00	0.9664	0.8520	0.2126	1.0000	0.7329	0.9878	0.9201	0.8753	0.9213	0.9637
0.25	2.00	0.9617	0.8967	0.2483	1.0000	0.8413	1.0000	0.9279	0.8694	0.9383	0.9762
0.25	1.00	0.9719	0.8720	0.3167	1.0000	0.7440	0.9765	0.9272	0.8810	0.9253	0.9752

of harder, misclassified instances. This gradient modulation is especially relevant in fine-grained object detection, where inter-class imbalance and intra-class variance are prevalent.

A full comparison of category-wise AP under each loss configuration is shown in Table V, highlighting consistent gains in damage categories like dent and tire flat. Across tested configurations, $\alpha = 0.50$, $\gamma = 2.0$ emerges as the most effective in optimizing both detection performance and computational cost. It achieves the highest overall AP (0.9352), with strong category-specific detection, particularly for dent (AP = 0.9784), scratch (AP = 0.8992), and tire flat (AP = 0.9793). This setup demonstrates a well-balanced emphasis on hard and easy examples, allowing the model to maintain a smooth learning curve and avoid convergence issues often observed with more extreme γ values.

In contrast, configurations with $\gamma \geq 3.0$ demonstrate increased recall, particularly in underrepresented classes such as crack and lamp broken, but often at the cost of reduced precision and inflated computation time. This is consistent with theoretical expectations: larger γ values overly penalize easy examples, potentially leading to instability or diminished attention to dominant class patterns. The configuration $\alpha = 0.25$, $\gamma = 5.0$ performs well in terms of recall and category coverage but suffers from longer execution time (1448.51s), underscoring the trade-off between model sensitivity and real-time applicability.

Conversely, $\alpha = 0.25$, $\gamma = 1.0$ demonstrates a desirable equilibrium, achieving high AP across small ($AP_s = 0.88$), medium ($AP_m = 0.93$), and large-scale objects ($AP_l = 0.98$), while maintaining low total execution time (1167.86s). This validates theoretical findings in focal loss literature that moderate focusing strengths ($\gamma \approx$ 1.0–2.0) allow sufficient emphasis on hard samples without excessively reducing the gradient signal of easier ones and maintaining stable optimization and class balance.

The trend across core classification metrics under focal loss tuning is visualized in Fig. 6, showcasing how precision-recall trade-offs vary with α–γ values. Complementing this, Table VI presents the per-damage-type classification performance, indicating how focal loss tuning affects both frequent and rare categories. Weighted precision, recall, and F1-scores remain stable across configurations, further supporting the model's capacity to handle long-tail distributions effectively. Macro-averaged metrics show more variance, especially in rare categories such as crack, revealing the sensitivity of these metrics to both parameter tuning and sample frequency. This suggests that future work may benefit from adaptive focal loss formulations, where α and γ are dynamically

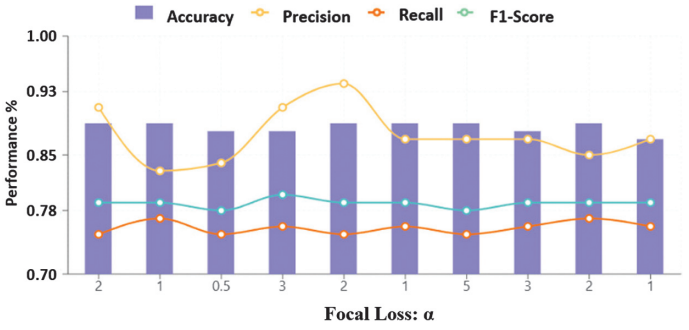

Fig. 6. Overall test set performance trends (accuracy, precision, recall, and F1-score) across focal loss α under fixed γ settings.

tuned based on real-time feedback such as class frequency, convergence rates, or uncertainty estimates.

In conclusion, $\alpha = 0.50$, $\gamma = 2.0$ is recommended as the default configuration when aiming for a high-performance and generalizable model. A holistic view of test set outcomes across focal configurations is summarized in Fig. 7, reinforcing $\alpha = 0.25$, $\gamma = 1.0$ as the most balanced setup. Meanwhile, $\alpha = 0.25$, $\gamma = 1.0$ provides a highly efficient alternative, particularly suited for embedded systems or latency-sensitive environments. This parameter tuning strategy confirms that focal loss should not be treated as a fixed or rigid function. Instead, it should be viewed as a flexible design mechanism that, when applied thoughtfully, can effectively adjust the balance between predictive accuracy, computational efficiency, and suitability for real-world applications.

Together, these analyses reaffirm the value of modular specialization, focal loss tuning, and region-centric attention for fine-grained vehicle damage detection in operationally constrained environments.

E. EXTERNAL BASELINES AND COMPUTATIONAL-COST ANALYSIS

To strengthen the evaluation scope, the proposed CropResMoE-50 was benchmarked against three widely adopted state-of-the-art architectures, namely ResNet-50, EfficientNet-B0, and Swin-Transformer Tiny (Swin-T), under identical conditions described in Section VI. Each backbone was implemented in frozen and

Table VI. Performance accuracy, precision, recall, and F1-score with six damages category categorized by focal loss α and γ

α	γ	Dent precision	Dent recall	Scratch precision	Scratch recall	Crack precision	Crack recall	Glass shatter precision	Glass shatter recall	Lamp broken precision	Lamp broken recall	Tire-flat precision	Tire-flat recall
0.75	2.00	0.8848	0.8949	0.7978	0.8486	0.7500	0.2000	1.0000	0.9909	0.9615	1.0000	0.9318	0.9762
0.75	1.00	0.8889	0.8864	0.8210	0.8406	0.6667	0.4000	1.0000	1.0000	0.8621	1.0000	0.9111	0.9762
0.75	0.50	0.8568	0.9006	0.8226	0.8127	0.7500	0.3000	1.0000	0.9909	0.9259	1.0000	0.9318	0.9762
0.50	3.00	0.8641	0.9034	0.8354	0.8088	0.8125	0.4333	1.0000	1.0000	0.8929	1.0000	0.9111	0.9762
0.50	2.00	0.8733	0.9006	0.8147	0.8406	0.7273	0.2667	1.0000	0.9909	1.0000	0.9600	0.9318	0.9762
0.50	1.00	0.8615	0.8835	0.8118	0.8247	0.7333	0.3667	1.0000	1.0000	0.9200	0.9200	0.9318	0.9762
0.25	5.00	0.8889	0.8864	0.8182	0.8606	0.8750	0.4667	1.0000	0.9909	0.9600	0.9600	0.9111	0.9762
0.25	3.00	0.8646	0.8892	0.8164	0.8327	0.7143	0.3333	1.0000	0.9818	0.9231	0.9600	0.9318	0.9762
0.25	2.00	0.8739	0.8864	0.8140	0.8367	0.7333	0.3667	1.0000	0.9909	0.9615	1.0000	0.9111	0.9762
0.25	1.00	0.8579	0.8920	0.8200	0.8167	0.7143	0.3333	1.0000	0.9909	0.9231	0.9600	0.9111	0.9762

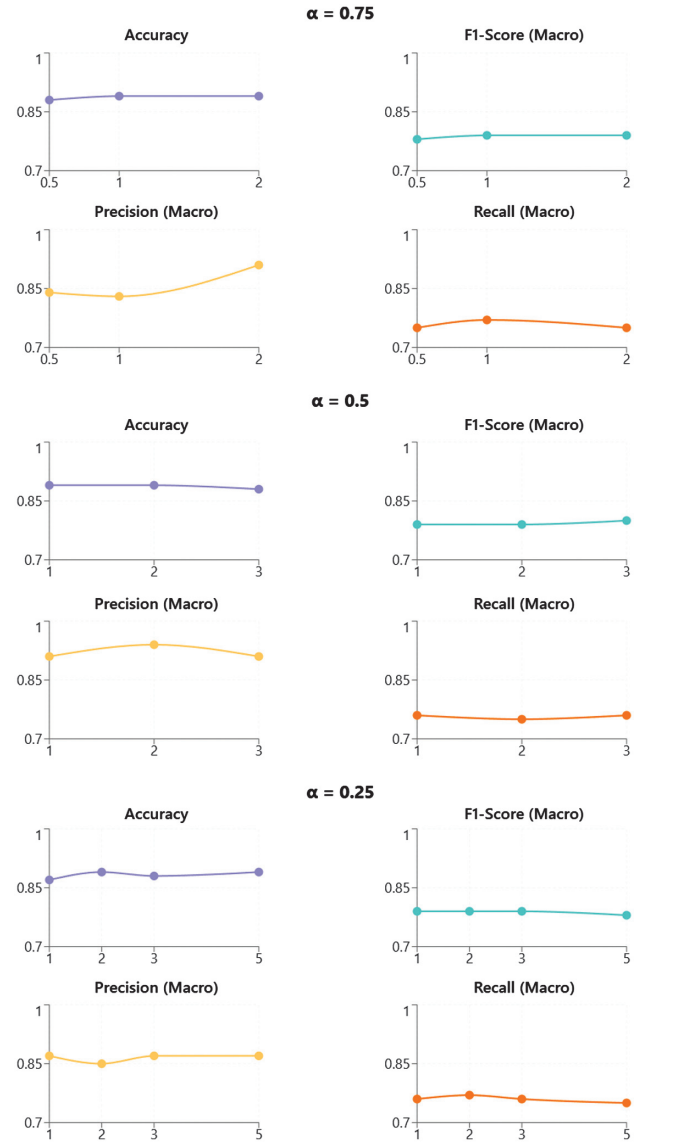

Fig. 7. Test set metric curves (precision, recall, accuracy, and F1-score) under varying focal loss α–γ settings.

unfrozen modes to quantify computational complexity, scalability, and efficiency, as shown in Table VII.

In the frozen setting, CropResMoE-50 achieves 87.4 % test accuracy and 86.2 % macro F1 at 4.13 GFLOPs, requiring roughly half the FLOPs of ResNet-50 and less than two-thirds that of Swin-T. This positions it on a favorable accuracy-efficiency frontier, demonstrating that its region-aware design yields strong performance with moderate complexity. It should be noted that this result was obtained under a standardized 20-epoch training protocol used for all baselines to ensure fair comparison. Under its original training configuration (24 epochs with tuned learning-rate scheduling), the same frozen CropResMoE-50 achieved 89.3 % test accuracy, confirming that the slight drop here stems from constrained hyperparameter settings rather than architectural differences.

When fully fine-tuned, all models improved; CropResMoE-50 (93.5 % test accuracy) remained competitive with EfficientNet-B0 while maintaining a more interpretable MoE structure. Swin-T showed limited gains, indicating less stable adaptation.

Table VII. Baseline models and computational-cost metrics

Model	Mode	Params (M)	FLOPs (G)	Latency (s/img)	Test Acc (%)	Macro F1 (%)
CropResMoE-50	Frozen	24.83	4.13	0.006	87.4	86.2
ResNet-50	Frozen	23.52	8.26	0.001	85.1	84.9
EfficientNet-B0	Frozen	4.02	0.82	0.001	83.2	83.1
Swin-T	Frozen	27.52	6.25	0.002	90.3	90.3
CropResMoE-50	Unfrozen	24.83	8.26	0.001	93.5	93.5
ResNet-50	Unfrozen	23.52	4.13	0.002	91.4	91.4
EfficientNet-B0	Unfrozen	4.02	0.41	0.011	94.9	94.9
Swin-T	Unfrozen	18.86	2.98	0.003	83.9	83.7

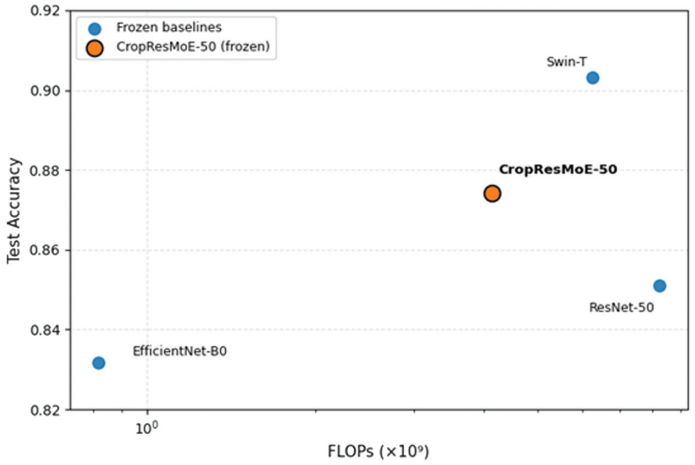

Fig. 8. Accuracy vs FLOPs trade-off (frozen models).

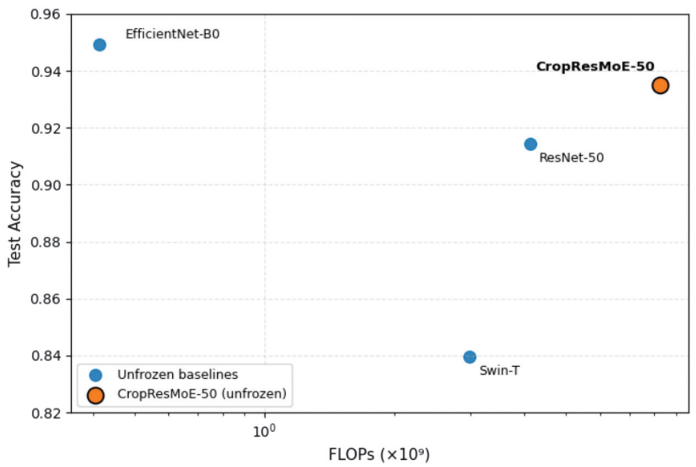

Fig. 9. Accuracy vs FLOPs trade-off (unfrozen fine-tuned models).

A scatter plot illustrating test accuracy versus FLOPs (log scale) reveals that CropResMoE-50 occupies the Pareto-optimal frontier, achieving high accuracy at moderate computational complexity. This positioning confirms its suitability for deployment on resource-constrained GPU systems. These results demonstrate that CropResMoE-50 achieves a superior accuracy–efficiency trade-off compared to recent CNN and transformer baselines, as shown in Fig. 8 and Fig. 9.

F. MODEL COMPLEXITY ANALYSIS

Table VII presents the computational requirements and parameter efficiency of all models in both frozen and unfrozen configurations. CropResMoE-50 demonstrates strong computational efficiency with 24.8M parameters and 4.13 GFLOPs in frozen mode, using approximately half the computational resources of ResNet-50 and two-thirds of Swin-T, while maintaining competitive accuracy.

Among all tested backbones, ResNet-50 provided the optimal balance between representational power, parameter efficiency, and interpretability, which justified its selection as the default feature extractor for the proposed CropResMoE-50 architecture. Although EfficientNet-B0 achieves the lowest computational cost through its mobile-optimized depthwise-separable convolutions, it prioritizes raw efficiency over interpretability. In contrast, CropResMoE-50 exposes its decision process via an explicit routing distribution over experts, providing transparent and interpretable predictions. Its specialized experts capture subtle damage patterns while maintaining a scalable design suitable for deployment. This architecture enables the model to effectively identify fine-grained regional cues essential for accurate vehicle damage assessment.

Full fine-tuning doubles the computational cost from 4.13 G to 8.26 G FLOPs but yields a 5–6 % accuracy improvement, demonstrating that the MoE architecture scales efficiently without excessive parameter inflation. This balanced profile of competitive inference speed, model interpretability, and high accuracy makes CropResMoE-50 particularly suitable for real-world insurance assessment systems, where explainable decisions are as important as computational efficiency.

To further validate interpretability, expert-routing statistics and gating visualizations were analyzed across all damage classes. The expert-class specialization matrix on the test set is generated (Fig. 10), while Fig. 11 and Fig. 12 present representative gating distributions for dent and tire-flat samples. The gating network exhibited nonuniform utilization, where specific experts dominated distinct categories. For example, dent samples were routed almost exclusively to Expert E2 ($\approx 89\%$), indicating learned specialization for structural deformation cues, whereas tire-flat images activated multiple experts (E0-E2, $\approx 28\%$ each), suggesting collaborative inference for texture-rich patterns.

This adaptive division of labor confirms that the MoE mechanism enables transparent and interpretable predictions, where each decision can be traced to a subset of experts with distinct functional roles. Unlike EfficientNet-B0, which achieves competitive accuracy but offers no visibility into its internal reasoning, CropResMoE-50 provides explainable expert routing that balances

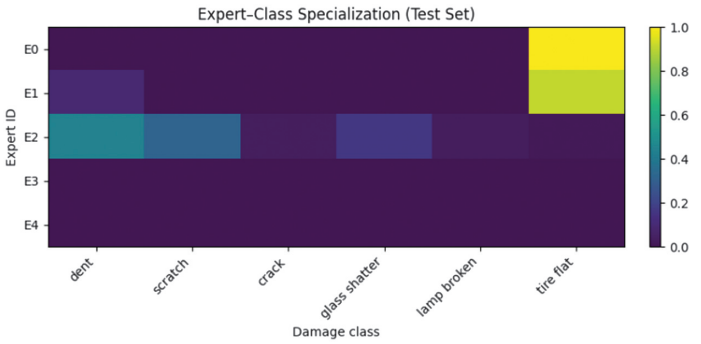

Fig. 10. Expert-class specialization (test set).

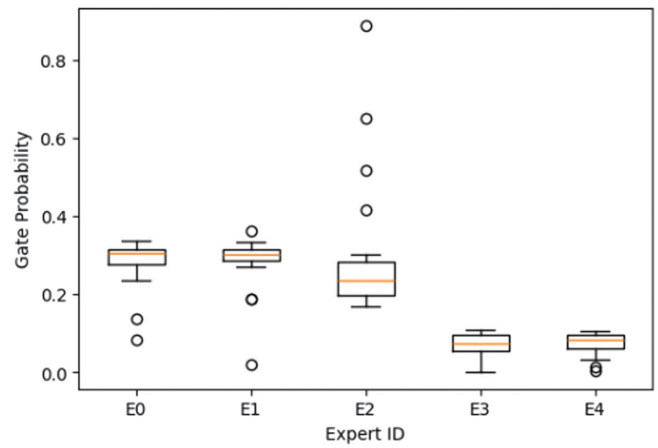

Fig. 11. Gating distribution across all "tire-flat" samples.

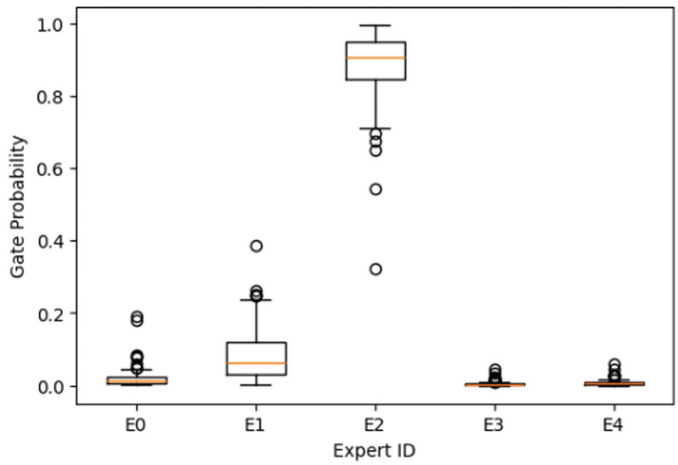

Fig. 12. Gating distribution across all "dent" samples.

accuracy, efficiency, and interpretability. Such traceable specialization is valuable for real-world insurance applications, where trust and explainability are equally critical as raw performance. Fig. 10 shows normalized routing probabilities of each expert (E0-E4) for every damage class on the CarDD test set. Expert E2 is strongly specialized for dent, while E0–E2 jointly cover tire flat and other classes, confirming nonuniform, class-specific expert utilization.

Overall, CropResMoE-50 achieves a balanced trade-off between predictive accuracy, computational efficiency, and interpretability. The observed expert specialization provides tangible evidence of transparent model behavior, bridging the gap between high-performance computer vision and explainable AI which is an essential requirement for insurance assessment and regulatory compliance. To further examine generalization under real-world conditions, additional validation was conducted on an insurance-claim dataset, as discussed in Section VIII.

VIII. CROSS-DATASET GENERALIZATION ON REAL-WORLD CLAIMS DATA

To evaluate the model's robustness beyond the public CarDD benchmark, an additional validation was performed using real-world insurance-claim images from Tune Protect Malaysia. The dataset consists of 291 manually annotated samples across six damage categories (dent, scratch, crack, glass shatter, lamp broken, and tire flat), standardized into the COCO format with 80-10-10 train/validation/test splits for compatibility with the CropResMoE-50 pipeline.

Under the frozen configuration, CropResMoE-50 achieved 83.3% test accuracy and 92.7% mean average precision (mAP), maintaining strong predictive performance despite the limited dataset size. High per-category precision and recall were observed for glass shatter (100%), tire flat (100%), and dent (85%), whereas scratch (50%) and lamp broken (67%) showed slightly reduced recall due to sample scarcity.

The model retained stable performance across object scales ($AP_s = 83$ %, $AP_m = 100$ %, $AP_l = 95$ %) and demonstrated minimal accuracy degradation (≈ -3 pp) relative to the CarDD test set, confirming robust cross-domain generalization. These findings validate the model's readiness for deployment in practical insurance scenarios, where real-world data exhibit greater diversity in lighting, perspective, and damage severity. This validation further supports the effectiveness of the MoE design in generalizing to unseen, real-world insurance data without any architecture modification.

Future work will focus on expanding the annotated real-world dataset and employing semi-automated labeling via ChromaDB to improve weak-category coverage and further enhance model robustness.

IX. SEMI-AUTOMATED DAMAGE LABELING WITH CHROMADB

This section introduces a practical enhancement that extends our classification framework into the domain of semi-automated annotation. Building upon the high-capacity CropResMoE-50 backbone, we incorporate a retrieval-augmented memory module using ChromaDB to facilitate soft pseudo-labeling. This hybrid configuration improves the efficiency of labeling unseen vehicle damage images, offering a viable strategy to bootstrap new datasets with minimal human intervention. The enhanced pipeline integrating CropResMoE-50 with ChromaDB-assisted retrieval is illustrated in Fig. 13, highlighting the dual-phase model-routing and confidence-based triage process.

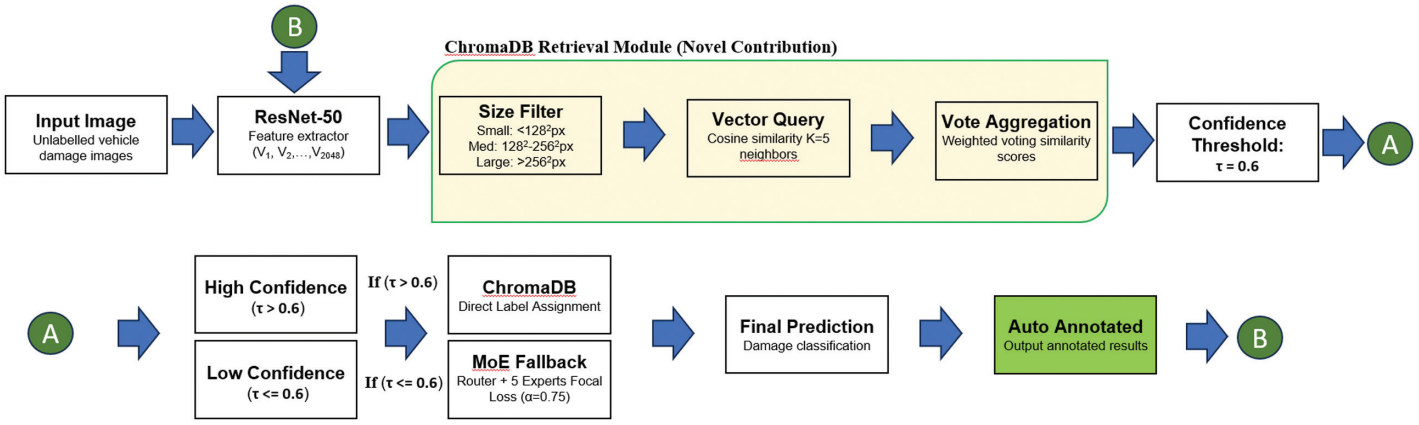

Fig. 13. Enhanced architecture of the CropResMoE-50 with ChromaDB retrieval.

A. SYSTEM OVERVIEW AND BLENDED PREDICTION LOGIC

Our pipeline is anchored by a pretrained ResNet-50 fused within the CropResMoE-50 architecture, trained on a curated blend of the CarDD dataset and manually cropped images. Once trained, the system accepts new vehicle damage inputs, typically derived from real-world insurance inspections. Each image is resized to 224×224, passed through ResNet-50, and converted into a 2048-dimensional feature vector.

This vector is then forwarded to the CropResMoE-50 model, which combines a routing layer and five expert subnetworks. The MoE generates a probability distribution across six known damage categories. If the confidence of the top-class prediction surpasses a predefined threshold ($\tau = 0.6$), the label is retained. Otherwise, the system invokes a fallback retrieval procedure via ChromaDB.

In the retrieval phase, the query embedding is matched against a pre-indexed feature database constructed from all training crops. These entries include both ground-truth labels and size metadata (categorized as "small," "medium," or "large"). The system retrieves the top-k nearest neighbors using HNSW-based ANN search. To enhance contextual alignment, additional filters such as size class or prior label type (e.g., "crack") may be applied. Instead of overriding the prediction, the retrieved neighbor votes are blended with MoE outputs using a convex formulation:

$$P_{final} = \lambda P_{MoE} + (1 - \lambda) P_{retrieval}$$

where $\lambda = 0.7$ governs the balance between the expert model and retrieved support. Here, P_{MoE} denotes the SoftMax output of the classifier, and $P_{retrieval}$ is the class distribution aggregated through distance-weighted voting. This mechanism allows exemplar memory to reinforce uncertain predictions without overpowering the model's learned semantics. The full hybrid pipeline is illustrated in Fig. 13.

This threshold $\tau = 0.6$ governs the switching logic between memory-based retrieval and model-driven prediction. Predictions above the threshold rely solely on the MoE output, while those below trigger retrieval augmentation.

B. EMPIRICAL TEST AND AUTO-LABELING OUTPUT

To evaluate real-world applicability, the semi-automated pipeline is tested on a batch of previously unseen, unlabeled vehicle damage images. This simulates a typical insurance workflow in which raw inspection images are submitted for initial triage without prior annotation.

The system was able to assign meaningful labels such as dent, glass shatter, and tire flat even in cases where the MoE model's top prediction confidence (ranging between 0.42 and 0.66). As visualized in Fig. 14, each output image is annotated with the predicted class and its associated confidence score. These examples demonstrate several key points: (1) Plausibility under uncertainty: Even when presented with challenging visual conditions (e.g., low lighting, reflective surfaces, or occlusion), the pipeline still infers semantically reasonable labels. (2) Interpretability via retrieval: The ChromaDB-based retrieval offers transparent support for low-confidence cases by surfacing similar historical exemplars, thereby improving explainability for human reviewers. (3) Practical value

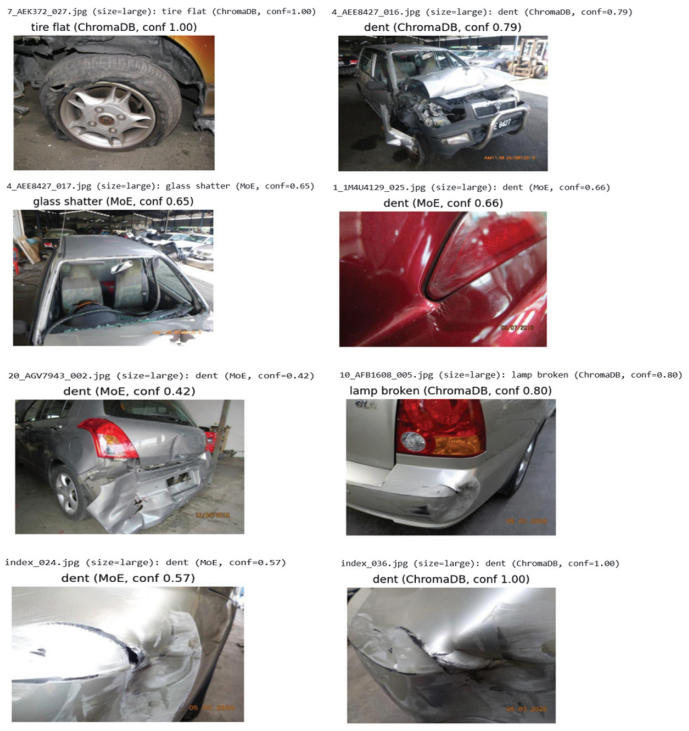

Fig. 14. Semi-automated annotated with the predicted class and its associated confidence score.

for dataset bootstrapping: For predictions exceeding the threshold $\tau = 0.6$, the pseudo-labels produced are sufficiently reliable to be used for expanding the training corpus without requiring manual verification or bounding box re-annotation.

This capacity to generate soft-labeled annotations in a hybrid, confidence-aware manner provides a scalable mechanism for continuous data enrichment. By incorporating retrieval support alongside probabilistic MoE outputs, the system bridges the gap between fully manual annotation and rigid, threshold-based automation.

C. LIMITATIONS IN SCOPE AND FULL AUTOMATION

While the proposed semi-automated labeling framework significantly reduces human effort, it is not a fully autonomous system. Its current design is confined to classification-only tasks, assuming pre-cropped inputs. Additionally, the retrieval component's reliability is contingent on the quality and diversity of the indexed embeddings. In scenarios with out-of-distribution (OOD) damage types or insufficient historical support, the retrieved exemplars may mislead the hybrid decision. Moreover, the MoE model, while robust, lacks temporal awareness or contextual reasoning, which are the factors that might be critical in multi-frame inspection scenarios (e.g., video-based claims). To advance toward a fully automated system, future iterations would require integration with damage detectors (e.g., Faster R-CNN or YOLO), bounding box regression heads, and confidence calibration modules. An adaptive self-training loop could also be employed to refine pseudo-label quality over time.

D. RESEARCH CONTRIBUTIONS AND FORWARD OUTLOOK

This study makes several technical and practical contributions. First, we introduce a hybrid semi-automated labeling framework that fuses an MoE classifier with a retrieval-augmented memory (ChromaDB), enabling confidence-aware pseudo-labeling of new vehicle damage samples. This novel fusion balances learned representations with example-driven interpretability which is an approach, especially useful in high-variance domains such as insurance and fleet inspection.

Second, we demonstrate that retrieval filtering by metadata (e.g., object size) can enhance semantic alignment without incurring significant computational cost. Third, the modular design allows for future extensibility, including integration into active learning workflows, inspection UIs, or digital claims systems.

Looking ahead, this methodology can generalize to other inspection-heavy domains such as construction, logistics, and agricultural damage assessment. By combining expert-based decision boundaries with memory-based similarity reasoning, the framework sets a precedent for more adaptive, transparent, and scalable annotation pipelines.

X. CONCLUSION

This study presented a comprehensive framework that integrates MoE architectures with retrieval-augmented learning to address both fine-grained classification and scalable annotation in vehicle damage detection. The proposed CropResMoE-50 model, enhanced with region-based preprocessing and scale-aware tuning, achieved significant improvements over prior architectures (RawMoE and ResMoE-50), reaching 89.30% test accuracy and high AP scores across small (0.88), medium (0.93), and large (0.98) object categories. These results underscore the value of spatially refined inputs and expert-routing mechanisms in handling intra-class variability and object-scale sensitivity.

When benchmarked against ResNet-50, EfficientNet-B0, and Swin-T, the proposed model demonstrated a superior accuracy–efficiency balance, maintaining high precision with moderate computational cost (24.8 M parameters, 4.13 GFLOPs). This confirms the architecture's suitability for real-time insurance workflows.

Furthermore, the strategic tuning of the focal loss function was instrumental in optimizing performance across imbalanced datasets. Configurations such as $\alpha = 0.25$ and $\gamma = 1.0$ offered robust generalization, while $\alpha = 0.50$ and $\gamma = 2.0$ yielded the highest overall AP (0.9352). These findings suggest that focal loss should be treated not merely as a loss function but as a sensitive hyperparameter framework tailored to dataset complexity.

To extend the utility of the model beyond static classification, we introduced a semi-automated labeling pipeline that combined MoE predictions with ChromaDB-based retrieval. This hybrid mechanism supported soft pseudo-labeling with confidence-aware blending, enabling scalable data enrichment without full manual annotation. Through empirical validation, we showed that even low-confidence inputs could be plausibly labeled with minimal supervision, thereby paving the way for practical deployment in real-world insurance workflows.

These results collectively validate CropResMoE-50 as a robust and interpretable framework that bridges fine-grained detection accuracy, computational efficiency, and semi-automated labeling scalability.

XI. FUTURE WORK

Looking forward, the next frontier lies in transitioning from semi-automated labeling to a fully automated, self-improving system. This would involve integrating the CropResMoE-50 backbone with object detection frameworks (e.g., YOLOv8, Faster R-CNN) to localize damage regions automatically, followed by classification and confidence-aware pseudo-label assignment. An active learning module could be incorporated to selectively retrain the model on high-uncertainty samples, enabling adaptive performance gains over time, building upon the strong accuracy–efficiency foundation established in this study.

The retrieval component (ChromaDB) will evolve into a dynamic, growing memory that assimilates newly labeled instances into its index, allowing real-time feedback and continual support for rare or edge-case predictions. Furthermore, the system could be deployed in a closed-loop architecture for auto-triaging insurance claims, providing label explanations via nearest-neighbor support —thereby reducing both annotation overhead and reviewer fatigue.

Expanding the dataset to encompass diverse vehicle types, weather conditions, camera angles, and damage modalities will also be critical for building generalizable models. Further evaluation on lightweight architectures and mobile inference platforms will also be explored to enhance deployment readiness. Ultimately, this research lays the groundwork for a scalable, interpretable, and autonomous pipeline that bridges high-performance damage detection with practical labeling efficiency.

The design philosophy underlying this study, the modular routing, retrieval support, and loss-based adaptability, has

presented a versatile blueprint for other fine-grained classification tasks beyond the automotive domain.

ACKNOWLEDGMENT

This study utilized the CarDD dataset, made publicly available by its creators. We acknowledge and appreciate the efforts of the dataset's authors in curating a comprehensive, high-resolution dataset tailored for vehicle damage detection. The CarDD dataset, with its COCO-format annotations and diverse damage categories, has been instrumental in enabling the training, validation, and evaluation of the proposed models. Further details about the dataset and its licensing terms can be found at https://cardd-ustc.github.io/.

CONFLICT OF INTEREST STATEMENT

The author(s) declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] S. Mishra, D. Kamal, and S. K. Kumar, "Vehicle damage identification using deep learning techniques," 2024, DOI. https://doi.org/10.1109/sceecs61402.2024.10481925.

[2] Y. Xiao *et al.*, "Real-time object detection for substation security early-warning with deep neural network based on YOLO-V5," in *2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, pp. 45–50, May 2022, DOI. https://doi.org/10.1109/GlobConET53749.2022.9872338.

[3] X. Li *et al.*,"YOLO V5-MAX: A multi-object detection algorithm in complex scenes," *2023 IEEE 6th Int. Conf. Ind. Cyber-Physical Syst. (ICPS)*, pp. 1–6, May 2023, DOI. https://doi.org/10.1109/ICPS58381.2023.10128009.

[4] R. E. van Ruitenbeek and S. Bhulai, "Convolutional neural networks for vehicle damage detection," *Mach. Learn. Appl.*, vol. 9, pp. 100332–100332, 2022, DOI. https://doi.org/10.1016/j.mlwa.2022.100332.

[5] C. Chen *et al.*, "Vehicle type recognition based on multi-branch and multi-layer features," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017, pp. 2245–2248. DOI. https://doi.org/10.1109/IAEAC.2017.8054374

[6] Z. Dong *et al.*, "Vehicle type classification using a semisupervised convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, pp. 2247–2256, 2015, DOI. https://doi.org/10.1109/tits.2015.2402438.

[7] W. Cai *et al.*, "A survey on mixture of experts," *arXiv preprint arXiv:2407.06204*, Jul. 2024. DOI. https://doi.org/10.48550/arXiv.2407.06204.

[8] Y. Kwon and S.-W. Chung, "MoLE : Mixture of language experts for multi-lingual automatic speech recognition," *ICASSP 2023-2023 IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 1–5, Jun. 2023, DOI. https://doi.org/10.1109/ICASSP49357.2023.10096227.

[9] M. M. Adnan *et al.*, "Automated image annotation with novel features based on deep ResNet50-SLT," *IEEE Access*, vol. 11, pp. 40258–40277, 2023, DOI. https://doi.org/10.1109/access.2023.3266296.

[10] A. Chaoub *et al.*, "Towards interpreting deep learning models for industry 4.0 with gated mixture of experts," *2022 30th Eur. Signal Process. Conf. (EUSIPCO)*, 2022, DOI. https://doi.org/10.23919/eusipco55093.2022.9909884.

[11] J. Thumm *et al.*, "Mixture of experts of neural networks and kalman filters for optical belt sorting," *IEEE Trans. Ind. Inf.*, vol. 18, 2021, DOI. https://doi.org/10.1109/tii.2021.3114282.

[12] W. A. R. Harshani and K. Vidanage, "Image processing based severity and cost prediction of damages in the vehicle body: A computational intelligence approach," *2017 Natl. Inf. Technol. Conf. (NITC)*, pp. 18–21, Sep. 2017, DOI. https://doi.org/10.1109/NITC.2017.8285649.

[13] P. Li, B. Shen and W. Dong, "An anti-fraud system for car insurance claim based on visual evidence," *arXiv(Cornell University)*, 2018, DOI. https://doi.org/10.48550/arXiv.1804.11207.

[14] X. Wang, W. Li, and W. Zhang, "CarDD: A new dataset for vision-based car damage detection," *arXiv (Cornell University)*, 2022, DOI. https://doi.org/10.1109/tits.2023.3258480.

[15] J. Zhou *et al.*, "Multi-task model fusion with mixture of experts structure," 2023, DOI. https://doi.org/10.1109/bigdia60676.2023.10429752.

[16] T. Das and S. Guchhait, "A hybrid GRU and LSTM-based deep learning approach for multiclass structural damage identification using dynamic acceleration data," *Eng. Fail. Anal.*, vol. 170, p. 109259, Mar. 2025, DOI. https://doi.org/10.1016/j.engfailanal.2024.109259.

[17] Y. A. A. Jarouf and M.-B. Kurdy, "A hybrid method to detect and verify vehicle crash with haar-like features and SVM over the web," 2018, DOI. https://doi.org/10.1109/comapp.2018.8460417.

[18] K. Peng *et al.*, "Multimodal fusion hybrid neural network approach for multi-class damage classification in high-speed rail track-bridge systems with multi-parameter," *Eng. Struct.*, vol. 328, p. 119710, Apr. 2025, DOI. https://doi.org/10.1016/j.engstruct.2025.119710.

[19] S. B. Sadkhan and S. F. Jawad, "Handwritten recognition based on hybrid ANN and wavelet transformation," 2018, DOI. https://doi.org/10.1109/ntccit.2018.8681190.

[20] N. Dhieb *et al.*, "A very deep transfer learning model for vehicle damage detection and localization," 2019, DOI. https://doi.org/10.1109/icm48031.2019.9021687.

[21] J. de Deijn, "Automatic car damage recognition using convolutional neural networks," *MSc Internship Report, Business Analytics, VU University Amsterdam*, Amsterdam, The Netherlands, 2018.

[22] P. M. Kyu and K. Woraratpanya, "Car damage detection and classification," 2020, DOI. https://doi.org/10.1145/3406601.3406651.

[23] N. T. Huynh *et al.*, "VehiDE Dataset: New dataset for automatic vehicle damage detection in car insurance," 2023, DOI. https://doi.org/10.1109/kse59128.2023.10299490.

[24] D. Widjojo, E. Setyati, and Y. Kristian, "Integrated deep learning system for car damage detection and classification using deep transfer learning," 2022, DOI. https://doi.org/10.1109/itis57155.2022.10010292.

[25] W. Zhang *et al.*, "Automatic car damage assessment system: Reading and understanding videos as professional insurance inspectors," in *Proc. . . . AAAI Conf. Artif. Intell.*, vol. 34, no. 9, pp. 13646–13647, Apr. 2020, DOI. https://doi.org/10.1609/aaai.v34i09.7110.

[26] M. A. Berwo *et al.*, "VEBD-HEL: A noval approach to vehicle exterior body damage parts classification in intelligent transportation systems," *Alex. Eng. J.*, vol. 108, pp. 961–975, Dec. 2024, DOI. https://doi.org/10.1016/j.aej.2024.09.050.

[27] K. Patil *et al.*, "Deep learning based car damage classification," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico: IEEE, Dec. 2017, pp. 50–54. DOI. https://doi.org/10.1109/ICMLA.2017.0-179.

[28] M. R. S. Ramazhan, A. Bustamam, and R. Anwar, "Car body damage detection system using YOLOv7," 2023, DOI. https://doi.org/10.1109/ice3is59323.2023.10335254.

[29] Z. Zheng and X. Zhu, "Mixture of prompt experts for natural language inference," 2024, DOI. https://doi.org/10.1109/ccece59415.2024.10667143.

[30] M.-D. Zhou, F. Xia, and X. Zhao, "An acoustic model using mixture of experts for environmental sound classification," 2024, DOI. https://doi.org/10.1109/cisat62382.2024.10695437.

[31] M. A. Aghdam, H. Jin, and Y. Wu, "DA-MoE: Towards dynamic expert allocation for mixture-of-experts models." 2024. [Online]. Available: https://arxiv.org/abs/2409.06669

[32] A. Shaabana, Z. Gharaee, and P. Fieguth, "Video relationship detection using mixture of experts," *IEEE Access*, vol. 11, 2023, DOI. https://doi.org/10.1109/access.2023.3257280.

[33] A. Vats *et al.*, "The evolution of mixture of experts: A survey from basics to breakthroughs," Preprints, Aug. 2024. DOI. https://doi.org/10.20944/preprints202408.0583.v2.

[34] C.-H. Hsieh, "Real-time car detection and driving safety alarm system with google tensorflow object detection API," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 1–4, Jul. 2019. DOI. https://doi.org/10.1109/ICMLC48188.2019.8949265.

[35] J. Krajewski *et al.*, "Scaling laws for fine-grained mixture of experts," *arXiv preprint arXiv:2402.07871*, Feb. 2024. DOI. https://doi.org/10.48550/arXiv.2402.07871.

[36] S. Angée, G. Thiebes and T. Bucher, "Towards an improved ASUM-DM process methodology for cross-disciplinary multi-organization big data & analytics projects," in L. Uden, B. Hadzima, and I.-H. Ting Eds., *Knowledge Management in Organizations*, Cham: Springer International Publishing, 2018, pp. 613–624. DOI. https://doi.org/10.1007/978-3-319-95204-8_51.

[37] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and Beyond," 2023.

[38] Y. W. Gustian *et al.*, "Detects damage car body using YOLO deep learning algorithm," *Sinkron*, vol. 8, pp. 1153–1165, 2023, DOI. https://doi.org/10.33395/sinkron.v8i2.12394.

[39] H. A. Setyawan, A. Bustamam, and R. Anwar, "Detection and assessment of damaged objects on the car body based on YOLO-V5," 2023, DOI: https://doi.org/10.1109/ice3is59323.2023.10335327.

[40] V. Sanh *et al.*, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," in *Proc. 5th Int. Conf. Learn. Representations (ICLR)*, 2020. [Online]. Available: https://arxiv.org/abs/1910.01108

[41] I. Analytics, "Analytics services datasheet," 2016, [Online]. Available: https://public.dhe.ibm.com/software/data/sw-library/services/ASUM.pdf

[42] F. Martínez-Plumed *et al.*, "CRISP-DM twenty years later: From data mining processes to data science trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 8, pp. 3048–3061, Aug. 2021, DOI. https://doi.org/10.1109/TKDE.2019.2962680.

[43] M. T. Hayat Suhendar and Y. Widyani, "Machine learning application development guidelines using CRISP-DM and scrum concept," *2023 IEEE Int. Conf. Data Softw. Eng. (ICoDSE)*, pp. 168–173, Sep. 2023, DOI. https://doi.org/10.1109/ICoDSE59534.2023.10291438.

[44] S. Maataoui, G. Bencheikh, and G. Bencheikh, "Predictive maintenance in the industrial sector: A CRISP-DM approach for developing accurate machine failure prediction models," in *2023 Fifth Int. Conf. Adv. Comput. Tools for Eng. Appl. (ACTEA)*, pp. 223–227, Jul. 2023, DOI. https://doi.org/10.1109/ACTEA58025.2023.10193983.