Optimizing Energy Efficiency in Multimodal Learning for Automated Vehicle Damage Evaluation

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Ahmed Sabanovic, BSc.
Matrikelnummer 12330648

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Ivona Brandić
Mitwirkung: Sabtain Ahmad, MSc
Dipl.-Ing. Daniel Bernhard May, BSc
Dipl.-Ing. Paul Joe Maliakel, BSc

Wien, 2. Dezember 2025

Ahmed Sabanovic	Ivona Brandić

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Optimizing Energy Efficiency in Multimodal Learning for Automated Vehicle Damage Evaluation

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Ahmed Sabanovic, BSc.
Registration Number 12330648

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Ivona Brandić
Assistance: Sabtain Ahmad, MSc
Dipl.-Ing. Daniel Bernhard May, BSc
Dipl.-Ing. Paul Joe Maliakel, BSc

Vienna, December 2, 2025

Ahmed Sabanovic

Ivona Brandić

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Ahmed Sabanovic, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 2. Dezember 2025

Ahmed Sabanovic

Acknowledgements

First of all, I would like to thank Prof. Ivona Brandić for the opportunity to work on this thesis and for her valuable guidance and support throughout the process. I would also like to thank my co-advisors Sabtain Ahmad, Daniel Bernhard May, and Paul Joe Maliakel for their dedicated support, constructive feedback, and constant encouragement during this journey.

My deepest thanks go to my dear parents. Your love, encouragement and support have meant the world to me. You have always encouraged me to push myself beyond my limits in order to achieve my goals and even in my moments of doubt, you never stopped believing in me.

To my beloved grandparents, thank you for your love, wisdom and encouragement. Our lovely conversations, your kind words, and your quiet sacrifices brightened my childhood and shaped who I am today.

Finally, I would like to thank my friends for their constant support, thoughtful advice, and for always being there for me. I also thank everyone who supported me in various ways throughout the course of my thesis work.

Kurzfassung

Fahrzeugkollisionen verursachen jedes Jahr über eine Million Todesfälle und führen zu erheblichen wirtschaftlichen Verlusten. Dennoch verlassen sich Versicherungsgesellschaften weiterhin auf manuelle, arbeitsintensive Schadensbewertungsprozesse, die langsam, fehleranfällig und schwer skalierbar sind. Daher ist eine automatisierte und genaue Bewertung von Fahrzeugschäden entscheidend, um die Bearbeitungszeit von Schadensfällen zu verkürzen, die Konsistenz der Schätzungen zu verbessern und die Kosten in der Kfz-Versicherungsbranche zu kontrollieren.

Diese Arbeit geht diese Herausforderung an, indem sie die Bewertung von Fahrzeugschäden als Benchmark-Aufgabe für die Bewertung von Feinabstimmungsstrategien für modernste Vision-Language-Modelle (VLMs) unter strengen Daten- und Energiebeschränkungen neu definiert. Ein multimodaler Korpus aus Fahrzeugbildern und textuellen Schadensbeschreibungen wird zusammengestellt und sowohl in roher als auch in vollständig vorverarbeiteter Form bereitgestellt, wobei letztere Qualitätsfilter wie Unschärferkennung, Belichtungssteuerung, Kontrastschwellenwert und Entfernung von Nahe-Duplikaten enthält. Fünf hochmoderne kompakte VLM-Architekturen, die für den Einsatz mit geringen Ressourcen entwickelt wurden (LLaVA, Qwen-VL, Bunny, Phi und SmolVLM), werden anhand eines repräsentativen Testsatzes bewertet. Anschließend wird das stärkste Basismodell mithilfe mehrerer Strategien domänenspezifisch feinabgestimmt. Die daraus resultierenden Kompromisse zwischen Energieverbrauch und Leistung werden mathematisch modelliert, um die effizienteste Konfiguration zu ermitteln.

Experimente zeigen, dass eine gezielte Feinabstimmung auf einem sorgfältig zusammengestellten Datensatz zu erheblichen Verbesserungen der praktischen Anwendbarkeit führt. Es wurde eine deutliche Verringerung des Energieverbrauchs im Vergleich zu Basismodellen beobachtet, verbunden mit einer bemerkenswerten Steigerung der Gesamtgenauigkeit und einer geringeren Inferenzlatenz. Im Vergleich zu einem naiv zusammengestellten Datensatz führt der kuratierte Korpus zu deutlich höheren Bewertungsmetriken, während gleichzeitig das Training beschleunigt und die Energieeffizienz deutlich verbessert wird. Diese Ergebnisse zeigen, dass eine energieeffiziente Feinabstimmung kompakter VLMs eine automatisierte Fahrzeugschadensbewertung für ressourcenbeschränkte Versicherungsabläufe praktikabel machen kann.

Abstract

Vehicle collisions cause over a million deaths each year and generate substantial economic losses. However, insurance companies continue to rely on manual and labor-intensive damage appraisal workflows that are slow, error-prone, and difficult to scale. Thus, an automated and accurate assessment of vehicle damage is crucial to reducing claim processing time, improving the consistency of estimates, and controlling costs in the automotive insurance industry.

This thesis tackles this challenge by reframing vehicle damage evaluation as a benchmark task to evaluate fine-tuning strategies in state-of-the-art vision-language models (VLMs) under strict data and energy constraints. A multimodal corpus of vehicle images and textual damage descriptions is assembled and provided in both raw and fully preprocessed forms, the latter incorporating quality filters such as blur detection, exposure control, contrast thresholding, and near-duplicate removal. Five state-of-the-art compact VLM architectures, designed for low-resource deployment (LLaVA, Qwen-VL, Bunny, Phi and SmolVLM) are benchmarked on a representative test set, after which the strongest baseline model undergoes domain-specific fine-tuning using multiple strategies. The resulting energy-performance trade-offs were mathematically modeled to identify the most efficient configuration.

Experiments demonstrate that targeted fine-tuning on a carefully curated dataset yields substantial improvements in practical utility. Specifically, the optimized model configuration achieved a 58.7% reduction in inference energy consumption compared to the baseline, while simultaneously increasing the F1-score from 61.05% to 68.35%. Compared to a naively constructed dataset, the curated corpus resulted in superior predictive performance while reducing training energy by 19.5%. Furthermore, a break-even analysis reveals that the energy cost of fine-tuning is recovered after processing approximately 48,000 images. These results indicate that energy-efficient fine-tuning of compact VLMs can make automated vehicle damage assessment viable for resource-constrained insurance workflows.

Contents

CHAPTER 1

Introduction

This thesis investigates the use of vision-language models (VLMs) for automated vehicle damage evaluation in energy-constrained environments. The research focuses on energy-aware inference and fine-tuning strategies that trade off predictive accuracy, inference latency, and power consumption across visual and textual data modalities. The work develops a dataset and an evaluation pipeline to compare compact, low-resource VLMs and to identify resource-efficient model adaptation approaches suitable for production insurance workflows.

1.1 Motivation

Road traffic injuries are a major global health and economic concern. According to the World Health Organization (WHO), vehicle accidents result in approximately 1.19 million annual fatalities globally and cost nations several percent of their gross domestic product (GDP) [Org23]. Given the large scale of these accidents, the insurance industry is facing scaling challenges while attempting to mitigate the financial repercussions of these incidents. Since 2020, the severity of bodily injury has risen by 20%, while material damage costs have surged by 47% among Generation Z drivers [Sol24]. Automotive insurance claim processing remains heavily reliant on manual inspections, whether performed in person or via photos and written reports. This dependency results in prolonged turnaround times, inconsistent evaluations, and escalating operational costs [AMNA24]. These challenges highlight the need for the insurance industry to embrace new solutions that will more effectively process the increasing volume of claims.

Automating damage appraisal with image analysis has clear potential to reduce these inefficiencies. However, recent research has highlighted the drawbacks of current methods, which typically overlook textual claim data and rely solely on image analysis [WLW23]. Textual data, typically found in incident reports, witness statements, or damage descriptions, provide essential context and granularity that images alone may miss. This single-modality limitation reduces the robustness and granularity of automated damage assessments [ZYHD20]. Another significant drawback of current large single-modality models is their computational inefficiency and high resource demands. These factors not only drive up operational costs but also limit effective scalability [TGLM20].

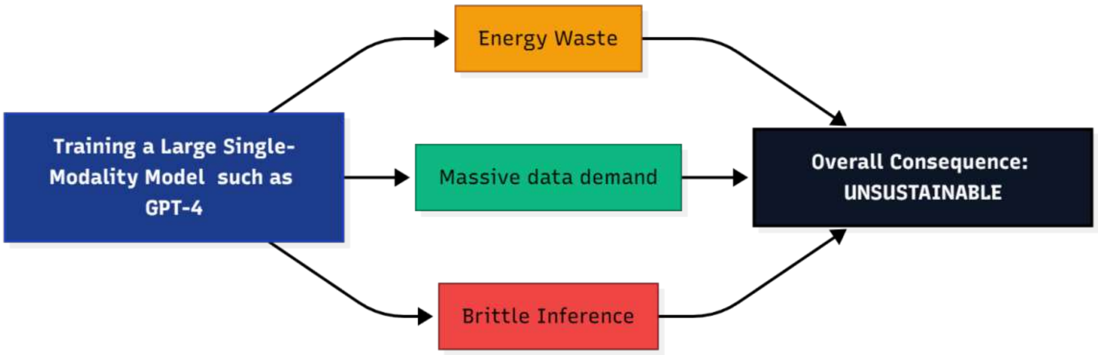

Training a Large Single-Modality Model such as GPT-4
Energy Waste
Massive data demand
Brittle Inference
Overall Consequence: UNSUSTAINABLE

Figure 1.1: An example illustrating the limitations during the training of a large single-modality model like GPT-4 [Ope23]. Three major challenges are highlighted: (1) Energy Waste; (2) Data Hunger; and (3) Brittle Inference.

As highlighted in Figure 1.1, the three main limitations when training large single-modality models such as GPT-4 are energy waste, data hunger, and brittle inference. While the substantial energy costs are detailed in the following section, the data requirements are equally staggering; training such models often necessitates datasets exceeding trillions of tokens, a scale that is increasingly difficult to sustain with high-quality data. Furthermore, these models suffer from brittle inference, where slight perturbations in input or shifts in domain can lead to disproportionately large errors, undermining their reliability in safety-critical applications like insurance assessment [Ope23].

Recent advances in multimodal deep learning, particularly in integrating vision and language models, hold significant promise in automating complex analysis tasks and improving overall efficiency in this field [GWW19]. By combining textual data and images, multimodal models are able to provide a more holistic approach to vehicle damage assessment. However, state-of-the-art VLMs such as Large Language and Vision Assistant (LLaVA) [LLLL24] and Qwen [BBC+23] often require massive datasets and high computational resources to reach peak performance. This raises important challenges around scalability, environmental impact, and deployment feasibility, especially for industries like insurance that demand real-time processing at scale. Training large multimodal models consumes significant amounts of energy: for example, training GPT-3 [BMR+20] required approximately 1,287 MWh of electricity and emitted an estimated 552 t of CO_2, raising not only financial but also serious sustainability concerns [PGL+21]. A recent study by Google provides insights into the inference-time energy costs of large-scale artificial intelligence (AI) systems. Google Gemini [ABW+23] production deployments show that the median text prompt consumes just 0.24 Wh of energy, which is equivalent to less than nine seconds of television viewing, while using about 0.26 mL of water for cooling, factoring in accelerator, host, idle, and data center overheads [EHP+25]. The creation of BLOOM [SFA+22], a 176B parameter open-access language model, provides further evidence of the environmental cost of extensive training. Its authors estimated total emissions of approximately 50.5 t CO_2eq when accounting for all processes, including power and equipment [LVL23].

These findings present a critical challenge: balancing high-performance inference with the urgent need to mitigate the environmental impact of training and deploying these models. A significant gap remains in designing architectures that simultaneously optimize accuracy, inference latency, and energy efficiency within constrained environments. Rather than replacing human inspectors, this thesis explores how AI can serve as a companion tool to assist insurance professionals by speeding up the appraisal process while maintaining reliability and scalability.

1.2 Research Questions and Objectives

This thesis seeks to answer the following research questions:

1. **RQ1:** How does the performance of a compact multimodal model fine-tuned on a dataset of car damage images and text descriptions compare to general-purpose VLMs in detection accuracy, processing time, and energy usage?
2. **RQ2:** What impact does a structured data curation pipeline have on model generalization and data efficiency versus naive aggregation?
3. **RQ3:** To what extent can energy-efficient model adaptation techniques minimize energy consumption while maintaining or improving model accuracy?

The primary objective of this thesis is to develop and fine-tune a sustainable, efficient, and scalable VLM specifically optimized for automated vehicle damage assessment. To address these research questions, the following steps will be undertaken:

1. **Curate a Multimodal Benchmark:** Assemble and preprocess a representative corpus of paired vehicle-damage images and structured textual descriptions to enable joint vision-language learning and evaluation (addresses RQ2).
2. **Evaluate State-of-the-Art Compact VLMs:** Benchmark five leading open-source VLM architectures (LLaVA [LLLL24], Qwen [BBC+23], Bunny [HLW+24], Phi-3 [AJA+24], and SmolVLM [MZF+25]) on practical tasks such as damage classification and type localization. The best performing model in terms of accuracy and energy usage will be selected for fine-tuning (addresses RQ1).
3. **Design an Energy-Efficient Fine-Tuning Pipeline:** Develop and integrate techniques to reduce compute and energy usage while maintaining performance. This includes:

- Clustering-based deduplication and automated quality filtering to minimize redundant or noisy data.
- Employ Unsloth [HHt23] to implement fast, resource-efficient training methods and lightweight model adaptation strategies.
- Compare image-only learning with joint image and text learning to determine which approach is more effective under limited data and energy constraints.
- Apply iterative pruning to reduce model size and inference latency.
- Adjust input prompts to enhance accuracy while minimizing energy usage.
- Apply quantization techniques to reduce memory and computational requirements with minimal performance loss (addresses RQ2, RQ3).

4. **Measure Sustainability Metrics:** Use NVIDIA Management Library (NVML) [NVI25] to log graphics processing unit (GPU) energy consumption for each training run, enabling transparent reporting of the computational footprint of every optimization (addresses RQ3).

5. **Compare Fine-Tuned VLM to General VLMs:** Benchmark the fine-tuned model against off-the-shelf VLMs, using metrics such as accuracy, precision, recall, F1-score, inference latency, and energy consumption per prediction (addresses RQ1).

The expected outcome is a fine-tuned VLM capable of accurate vehicle damage detection while minimizing energy consumption. By prioritizing efficiency and low energy consumption, the model aims to support insurance inspectors in accelerating claims processing while contributing to sustainable AI practices.

1.3 Outline

This thesis is organized as follows. Chapter 2 provides the theoretical background for the methods used in this research. Key concepts related to damage assessment in the insurance industry are introduced, as well as the basics of image and text data integration. The growing importance of sustainability in the context of large-scale automation and computational models is also discussed.

Chapter 3 presents a review of related work, highlighting current challenges and approaches to automating damage evaluations, with a focus on how visual and textual data can be integrated for more efficient claim processing. The chapter also addresses the environmental concerns associated with large models and the need for sustainable solutions.

Chapter 4 introduces the proposed framework for automating damage assessments, focusing on the integration of image processing and textual claim data to improve accuracy, efficiency, and sustainability.

Chapter 5 explains the experimental setup, including the datasets used for training and testing, evaluation metrics, and the model architecture. The energy consumption and resource demands of the model are also examined, discussing strategies to minimize energy usage, and in turn minimize the environmental impact while ensuring scalability.

Chapter 6 discusses the results of the experimental evaluations, providing information on the performance, efficiency, and sustainability of the model. This chapter also compares the proposed solution with existing methods and evaluates its operational and environmental effectiveness.

Finally, Chapter 7 summarizes the findings of this research and reflects on the possible implications of this research on both the insurance industry and the environment. Potential further improvements that can be made to make the model even more efficient, scalable, and sustainable are discussed.

CHAPTER 2

Background

This chapter provides the foundational knowledge necessary to understand the methods and contributions of this thesis. It begins with a brief overview of neural networks and the transformer architecture, which form the basis of many modern models.

The field of multimodal learning is then explored, with a focus on current state-of-the-art VLMs such as LLaVA [LLLL24], Qwen [BBC+23], Bunny [HLW+24], Phi-3 [AJA+24], and SmolVLM [MZF+25].

Subsequently, the task of automated vehicle damage assessment, its role in the modern insurance industry, and the key models used in this field are introduced.

Following this, the significant computational and environmental costs associated with training and deploying large-scale multimodal models are discussed. A major component of this thesis is the need for more efficient training techniques that are driven by sustainability concerns. As a result, core techniques used to reduce power consumption in deep learning are introduced.

Finally, the importance of data quality and diversity in multimodal tasks is highlighted, explaining how careful dataset curation can improve generalization while reducing energy consumption.

2.1 Neural Networks and Transformers

Neural networks [LBH15] are computational models inspired by the structure of the human brain. They consist of layers of interconnected units that process information and learn representations from data. These advances have enabled breakthroughs in fields such as image recognition, speech processing, and natural language understanding [LBH15].

The Transformer architecture [VSP+17] marked a significant milestone by using self-attention mechanisms instead of recurrence or convolution. This design enables models to process extensive dependencies across sequences while maintaining parallel training capabilities for billions of parameters [VSP+17].

2.2 Multimodal Learning and Vision-Language Models

Multimodal AI refers to machine learning (ML) models capable of processing and integrating information from multiple modalities or types of data. These modalities can include text, images, audio, video, and other forms of sensory input [WGC+23]. Consequently, multimodal models can produce more accurate, context-sensitive results.

For example, a model that analyzes both an image and its accompanying text can generate responses that reflect both the visual and textual context, improving performance on tasks such as Visual Question Answering (VQA), image captioning, and multimodal summarization.

Figure 2.1 illustrates the concept of a multimodal model, visually representing how different data types are fed into a unified system. This allows the model to process and reason across these different data types, analogous to how humans integrate multiple senses.

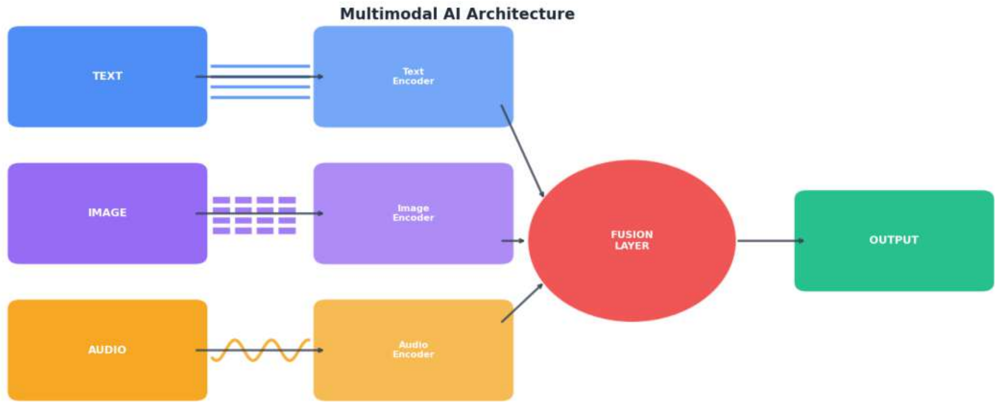


Figure 2.1: Architecture of a Multimodal Model

Multimodal models have advanced rapidly since the 2010s. Recent studies have shown a 32.4% improvement in accuracy for cross-modal tasks, a 41.3% increase in task completion rates, and a 37.8% reduction in error rates. These improvements have been achieved while maintaining fast processing speeds, averaging around 45 milliseconds [Nay25].

2.2.1 State-of-the-Art VLMs

Following the discussion on the basics of multimodal AI, several leading VLMs currently in use are examined. These models are among the most influential and widely adopted in the field.

LLaVA

LLaVA represents a significant contribution to the field of multimodal AI, particularly through its approach to visual instruction tuning [LLLL24]. Visual instruction tuning is a training paradigm for multimodal models in which a Large Language Model (LLM) and a vision encoder are fine-tuned on a dataset of image-text instruction-response pairs. The instructions ask the model to perform tasks that depend on visual content (e.g., describing, reasoning, answering questions), and much of the dataset is generated by a strong language model (e.g., GPT-4) to scale diversity. Vision embeddings are projected into a space compatible with the language model, and then the model learns to follow instructions that combine textual and visual input [LLWL23].

LLaVA's architecture connects visual and language components using a projection layer. Training typically happens in stages: first, only the projection layer is trained while keeping the vision and language models fixed. Later, the language model is fine-tuned as well, while the vision encoder often stays unchanged. This method helps ensure that the visual and linguistic components are well aligned and can work together effectively to produce coherent, context-sensitive outputs [LLWL23, LZG+25].

Qwen

Qwen-VL is an open-source large VLM known for its powerful combination of advanced capabilities and accessibility. It features numerous improvements in resolution handling, multimodal integration, and language coverage compared to other VLMs. One of its key innovations is the dynamic resolution system, which allows the model to adapt to different image sizes on the fly, ensuring that it captures more details and delivers more accurate results across a range of inputs [BBY+24, BBC+23, WBT+24].

Another central feature is Multimodal Rotary Position Embedding (M-RoPE), which fuses spatial and temporal information across text, images, and videos. M-RoPE allows the model to better capture real-world 3D and time-based dynamics, making it highly capable in video understanding and streaming content. This model has achieved state-of-the-art performance on several industry benchmarks, such as DocVQA, InfoVQA, and MathVista [WBT+24].

A further key feature of Qwen-VL is its exceptional multilingual support. While most models are limited to English or Chinese, Qwen-VL supports most European languages, Japanese, Korean, Arabic, and Vietnamese. The Qwen2-VL series includes models of varying sizes, from the efficient Qwen2-VL-2B (designed for on-device use) to the highly capable Qwen2-VL-72B (for complex tasks), all leveraging a 675M vision encoder and different LLM sizes [WBT+24].

The recently introduced Qwen2.5-VL [BCL+25] builds on these strengths with sharper reasoning and efficiency. It achieves a more precise alignment between the text and visual modalities and demonstrates significant gains in grounding and multimodal mathematical reasoning. These improvements make it especially strong for document understanding and complex visual question answering [BCL+25].

Bunny

Bunny is a family of lightweight yet powerful multimodal models featuring plug-and-play language and vision backbones, aligned through a multimodal projector. A major challenge with multimodal LLMs (MLLMs) is their high compute and memory demands. Although reducing model size improves efficiency, it typically impacts performance [CWZZ17]. Bunny demonstrates that this is not necessarily the case. By focusing on high-quality training data, it delivers small, yet competitive multimodal models. This approach highlights that performance does not scale solely with model size, but can also be significantly influenced by the quality of the training data [HLW+24].

Its primary innovation lies in data-centric training. Instead of relying on raw scale, Bunny uses "dataset condensation" to build a cleaner, more informative dataset from LAION-2B. LAION-2B is an openly available large-scale dataset consisting of approximately 2.17 billion image-text pairs collected from the web [LHLW24]. By employing semantic deduplication, alignment filtering, and diversity sampling, it reduces billions of samples to just 2M high-quality pairs for pre-training, plus a 695K dataset for instruction tuning [HLW+24].

Phi-3

The Phi-3 model family represents a major advancement in the development of compact yet high-performing language models. Its flagship variant, Phi-3-mini, contains approximately 3.8 billion parameters and was trained on roughly 3.3 trillion tokens. Notably, despite its relatively modest size, it achieves benchmark performance, around 69% on Massive Multitask Language Understanding (MMLU) and 8.38 on MT-bench, comparable to substantially larger models such as GPT-3.5 and Mixtral 8×7B [AJA+24].

The core innovation of Phi-3 lies not in parameter count, but in the quality and scale of its training corpus: a carefully curated and filtered web dataset enriched with synthetic, chat-aligned, and safety-optimized data. The technical report further introduces scaling variants, including Phi-3-small ($\approx$ 7B parameters) and Phi-3-medium ($\approx$ 14B parameters), each trained on approximately 4.8 trillion tokens, achieving $\approx$ 75% and $\approx$ 78% on MMLU, respectively [AJA+24].

SmolVLM

The SmolVLM family is an open-source collection of compact VLMs developed by Hugging Face to investigate the trade-off between model size, computational efficiency, and multimodal performance [MZF+25]. In contrast to conventional large-scale VLMs, SmolVLM prioritizes lightweight architectures that operate effectively on consumer-grade hardware while retaining competitive accuracy. The models span 256 million to 2.2 billion parameters, achieving substantial reductions in memory and computational demands with minimal degradation in downstream performance.

A key innovation of SmolVLM lies in its balanced capacity distribution between the vision encoder and the language model, which optimizes token compression and visual feature representation for efficient cross-modal reasoning. The architecture employs aggressive image token reduction strategies such as sub-image splitting and pixel shuffling, to preserve spatial fidelity while minimizing token count. Consequently, even the smallest variant, SmolVLM-256M, supports multimodal reasoning using less than 1 GB of GPU memory, outperforming substantially larger systems such as Idefics-80B on certain benchmarks [MZF+25].

2.3 Automated Vehicle Damage Assessment

Automated vehicle damage assessment refers to the use of AI and computer vision techniques to detect, classify, locate, and estimate the severity and repair costs of vehicle damage from images or videos, with minimal human supervision. In practical deployments, this automation can accelerate claims processing, reduce human error and subjectivity, improve consistency across appraisals, and enable large-scale fleet monitoring and fraud detection [PZCGPM+24].

This process is typically integrated into the First Notice of Loss (FNOL) workflow, the initial step in the insurance claims process where the policyholder reports an incident. Traditionally, FNOL involves a phone call or a mobile app submission where the user provides a textual description of the accident and uploads photos. An automated system must then reconcile these two modalities, verifying, for instance, that a "dent on the left bumper" described in the text matches the visual evidence provided [Cha22].

Historically, vehicle damage assessment involved in-person inspection by automotive appraisers and manual review of numerous images, which was a time-consuming process. This process frequently led to subjectivity, inconsistency, and delays in claims management, contributing to financial losses for companies. The financial incentive for insurers is a primary driver for AI adoption, compelling the industry to pursue higher precision and lower false positive rates to minimize financial losses. High false positive rates, in particular, can significantly contribute to financial losses, making it a key issue that AI systems aim to address [PZCGPM+24, VAA+25].

Automated vehicle damage assessment uses powerful computer vision and deep learning models, which enable machines to "see" and interpret images or videos. These models can detect damage, pinpoint its location, and assess how severe it is [PZCGPM+24]. Some of the most popular and effective models used in this space include the following:

2.3.1 Convolutional Neural Networks

A broad class of deep learning architectures, convolutional neural networks (CNNs) form the backbone of most vehicle damage detection systems. CNNs learn hierarchical feature representations via convolutional filters, pooling, and nonlinearities; earlier layers capture low-level patterns (edges, textures) while deeper layers learn higher-level semantics (parts, objects). They are trained on extensive datasets of labeled images to accurately classify damage types, even under varying lighting and angle conditions. CNNs have demonstrated robust performance in object detection and image classification tasks, making them well-suited for identifying and localizing vehicle damage [Zhu25].

2.3.2 You Only Look Once

You Only Look Once (YOLO) [RDGF16] is a family of models that builds on the CNNs paradigm and specializes in real-time object detection. Unlike generic CNNs classifiers, YOLO reframes object detection as a single regression problem, directly predicting bounding box coordinates and class probabilities from full images in one evaluation. YOLO models, including YOLOv5, YOLOv7, and YOLOv8, represent state-of-the-art deep learning architectures widely adopted for real-time object detection. These models enable rapid and accurate detection and classification of vehicle damage, with some ensemble approaches significantly improving inference speed to process thousands of instances per minute. YOLOv8, in particular, has been recognized for its strength and stability in damage detection applications [PZCGPM⁺24, VAA⁺25]. Recently, YOLOv13 has been introduced as a state-of-the-art model, demonstrating superior performance in real-time object detection tasks [LLW⁺25].

2.3.3 Mask Region-Based Convolutional Neural Network

Mask Region-Based Convolutional Neural Network (Mask R-CNN) takes object recognition a step further by adding the ability to perform instance segmentation. This means that Mask R-CNN can not only recognize objects, but also separate them into distinct regions, even within complex scenes. For vehicle damage detection, this model can be used to accurately crop vehicles from images and precisely identify areas of damage. Additionally, to speed up training and improve accuracy, Mask R-CNN models are often initialized with pretrained weights from large, diverse datasets, such as Microsoft COCO [CJ25].

2.3.4 Residual Network

Residual Network (ResNet) models [HZRS16] are deep learning models that are particularly effective at detecting complex features and improving classification accuracy. As demonstrated by He et al. [HZRS16], ResNets achieved a top-5 error rate of just 3.57% on the ImageNet test set, winning the 1st place in the ILSVRC 2015 classification task, which highlights their robustness and generalization ability across domains. In the context of vehicle damage assessment, ResNets like ResNet50 are commonly used for identifying damaged vehicles in images. Pretrained ResNet models, such as ResNet50, can be fine-tuned on specific vehicle damage datasets through transfer learning, leveraging knowledge gained from broader image recognition tasks [VSBS25].

2.3.5 Generative Adversarial Networks

Generative adversarial networks (GANs) [GPM+20] have a unique role in automated vehicle damage assessment. As introduced by Goodfellow et al. [GPM+20], GANs are trained by pitting two models against each other in a minimax game, where a generator creates synthetic data and a discriminator attempts to distinguish it from real data, leading to the generation of highly realistic samples. GANs are capable of generating realistic images of damaged vehicles. This is particularly valuable because damaged vehicle images are sparse and often hard to obtain. Generated vehicle damage images can then augment training datasets and improve the robustness and generalization of damage estimators [VSBS25].

2.3.6 Limitations of Unimodal Approaches

While the aforementioned computer vision models have advanced the state of damage detection, they suffer from a critical limitation: they are unimodal. Models like YOLO or ResNet process visual data in isolation, ignoring the rich contextual information contained in the claimant's textual report. This "semantic gap" can lead to several issues:

- **Ambiguity:** A visual scratch might be old wear-and-tear or fresh accident damage. Without the textual context (e.g., "I scraped the side against a pillar yesterday"), a vision-only model cannot distinguish relevance.
- **Inconsistency:** If a user reports "front bumper damage" but uploads a photo of the rear, a unimodal model will correctly detect the rear damage but fail to flag the inconsistency, potentially allowing fraudulent or erroneous claims to proceed.
- **Lack of Reasoning:** Unimodal models excel at pattern recognition but lack the reasoning capabilities to infer causality or severity from a combination of visual cues and descriptive narratives.

VLMs address these shortcomings by fusing visual features with linguistic semantics, enabling a more holistic and accurate assessment process.

Despite technical progress, several operational challenges remain before automated damage assessment can be widely adopted. Key issues include limited data, ensuring models perform well in real-world conditions, managing high computational demands, and integrating with legacy systems. In addition, concerns about bias, transparency, data privacy, and accountability need to be addressed to build trust and meet regulatory standards [PZCGPM+24, VAA+25, Zhu25, VSBS25, CJ25].

2.4 Energy-Efficient Learning Techniques

Large model training and deployment carry substantial environmental and operational costs. For instance, a training run for a 175B-parameter model such as GPT-3 has been reported to consume on the order of 1,287 MWh of electricity and emit roughly 552 metric tons of CO_2. Furthermore, this energy consumption continues after the model is trained. For instance, a single ChatGPT query uses roughly five times the energy of a typical web search [Zew25].

These energy demands are not just limited to the training phase. On the infrastructure side, data-center electricity demand has surged: North American data-center capacity roughly doubled (from ≈2,688 MW to 5,341 MW between 2022 and 2023), and global data-center power use reached ≈460 TWh in 2022, surpassing the total energy consumption of many nations [Zew25].

Concurrently, Strubell et al. [SGM19] estimate that developing modern NLP models, once hyperparameter tuning and repeated experimentation are included, can emit hundreds of thousands of kilograms of CO_2, comparable to the lifetime emissions of several cars. More recent large-scale efforts report similarly sobering figures: the training of BLOOM-176B alone produced ≈25-50 tonnes CO_2eq, depending on whether only electricity use or broader operational factors are considered [LVL23].

These numbers clearly illustrate the enormous compute and energy resources required by large multimodal models, both during their training and deployment. The resulting environmental costs (electricity use, carbon emissions, water usage, electronic waste, etc.) are now comparable to those of small countries. This reality has driven the rise of "Green AI" research, which seeks methods to preserve model performance while cutting resource consumption.

2.4.1 Core Efficiency Techniques

To address the energy challenges posed by large models, researchers have developed a variety of techniques to improve efficiency and reduce computational costs. One of the most common approaches is model compression, which includes strategies such as pruning, quantization, and knowledge distillation [PSUL25]. These techniques aim to make models smaller and faster without sacrificing too much accuracy. Recent surveys highlight the growing importance of these methods: Zhu et al. [ZLL+24] provide an overview of compression strategies for LLMs, while a more recent study focuses on transformer-specific compression techniques, including architectural modifications alongside pruning, quantization, and distillation [TWG+24].

Pruning

Pruning is a technique that involves removing unnecessary weights or neurons from a neural network. By eliminating these redundant elements, pruning reduces the overall size of the model and the number of computations (known as floating point operations, or FLOPs) required during inference. The result is a more efficient model that can perform predictions faster and with less resource consumption [PSUL25].

Figure 2.2, which was inspired by [PSUL25], illustrates the pruning process. It shows how a dense, fully-connected network is gradually transformed into a sparser version by removing less important connections.

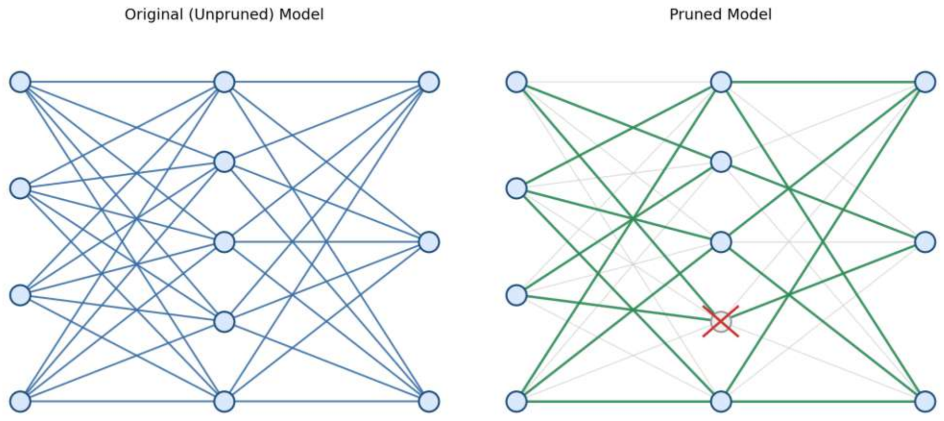


Figure 2.2: Comparison Between Unpruned and Pruned Models. The figure shows a dense, unpruned model (left) with all connections intact versus a pruned model (right) with less significant connections removed.

Quantization

Quantization is another key technique for improving the energy efficiency of deep learning models. It involves reducing the numerical precision of the model's weights and activations. For example, instead of using 32-bit floating point numbers, quantization maps these values to smaller representations, such as 8-bit integers or even fewer bits. This reduction in precision helps reduce both the memory requirements and computational costs, as smaller numbers take less space to store and require less processing power to handle [PSUL25].

Quantization can be applied in two main ways: post-training quantization (PTQ) and quatization-aware training (QAT). PTQ converts a pre-trained model to lower precision without retraining, making it fast and easy to apply but potentially leading to accuracy loss. QAT, on the other hand, simulates quantization effects during the training process, allowing the model to adapt to the lower precision and typically yielding better performance at the cost of increased training time [PSUL25]. The PTQ process is shown in Figure 2.3.

Knowledge distillation

Knowledge distillation is a technique that involves training a smaller model, called the "student," to mimic the behavior of a larger, more complex model, called the "teacher." The goal here is to transfer the performance capabilities of the larger model to a more

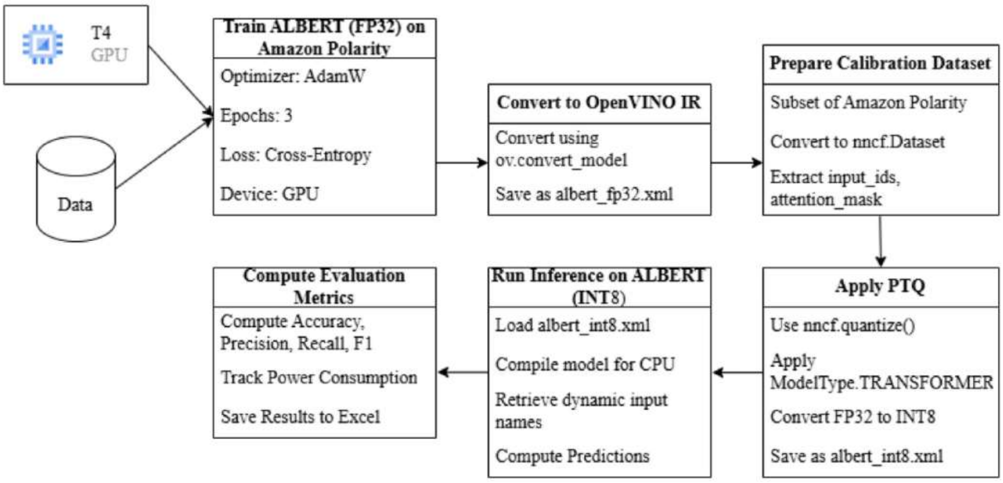


Figure 2.3: PTQ process. This figure outlines the process of applying PTQ to the ALBERT model, showing data calibration, OpenVINO IR conversion, and quantization into INT8 format for efficient inference ([PSUL25], Fig. 11).

compact and efficient one, maintaining similar accuracy while significantly reducing the size and complexity of the network [PSUL25].

As illustrated in Figure 2.4, the student model learns by minimizing a combined loss function. This loss typically consists of two components: a distillation loss, which aligns the student's “soft” class probabilities with those of the teacher, and a standard student loss that compares the student's predictions against the ground truth labels. This dual objective ensures the student not only learns the correct answers but also mimics the teacher's generalization patterns [PSUL25].

Research has shown that applying techniques such as pruning, quantization, and knowledge distillation can lead to substantial environmental benefits. For example, a study found that combining pruning and distillation reduced the energy consumption of a Transformer model (BERT) by about 32% while preserving nearly 96% of the model's accuracy [PSUL25]. This shows that it is possible to achieve significant energy savings without a dramatic loss in model performance.

Parameter-Efficient Fine-Tuning

Beyond model compression, another critical research direction is parameter-efficient fine-tuning (PEFT) [WCJ$^+$25]. While compression strategies primarily aim to improve inference-time efficiency, PEFT is designed to enable cost-effective adaptation and training. Instead of updating all parameters, PEFT modifies only a small, carefully chosen subset of the pretrained model's weights. This approach significantly reduces memory, computation, and energy demands, while still achieving performance that is often close to, or even matches, that of full fine-tuning [WCJ$^+$25].

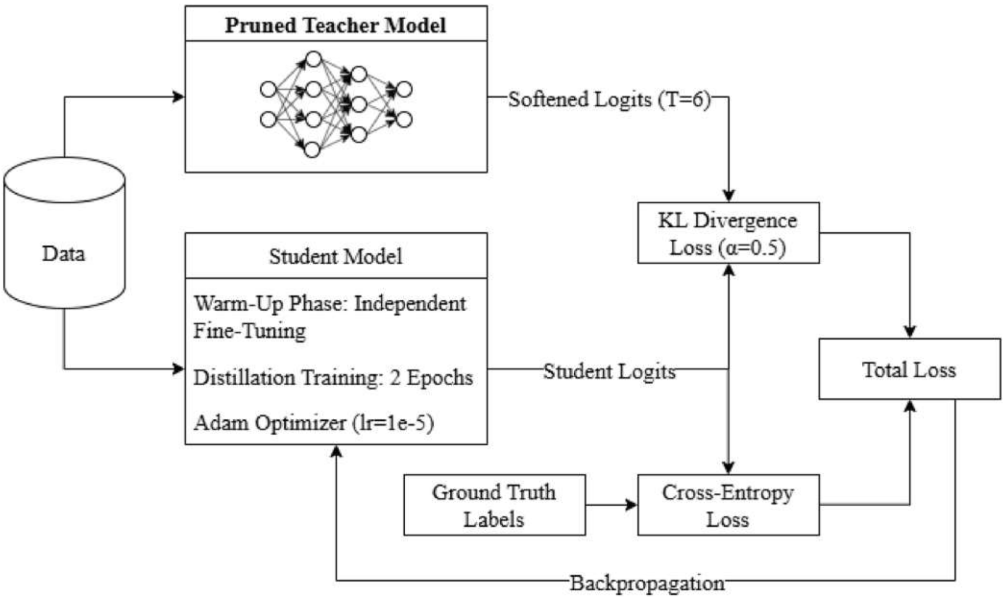


Figure 2.4: Knowledge distillation process. This figure shows how the student model is trained by minimizing a combined loss that equally balances KL divergence with the teachers' softened outputs and cross-entropy with ground truth labels ([PSUL25],Fig. 11).

A widely adopted variant of PEFT is Low-Rank Adaptation (LoRA) [HSW+22]. LoRA operates on the premise that the change in weights during model adaptation has a low "intrinsic rank". Instead of updating the full weight matrix W, LoRA freezes the pre-trained weights and injects trainable rank decomposition matrices A and B into each layer, such that the weight update is represented as $\Delta W = BA$. Since the dimensions of A and B are much smaller than W, the number of trainable parameters can be reduced by up to 10,000 times, drastically lowering GPU memory requirements.

Building on this, Quantized LoRA (QLoRA) [DPHZ23] further improves efficiency by backpropagating gradients through a frozen, 4-bit quantized pre-trained model into Low-Rank Adapters. This allows for the fine-tuning of massive models on a single consumer GPU, making high-performance multimodal adaptation accessible in resource-constrained environments.

2.4.2 Measuring Energy Efficiency

To quantify the sustainability of AI models, precise metrics are required beyond standard accuracy scores. "Green AI" research emphasizes the measurement of the computational cost required to achieve a result. Key metrics include:

- **Energy Consumption (Joules/kWh):** The total electrical energy drawn by the hardware (GPU, CPU, memory) during a training run or inference batch. This is often measured using hardware-specific interfaces like the NVML.

- **Inference Latency:** The time taken for the model to process a single input and generate a prediction. Lower latency is critical for real-time insurance workflows.
- **Throughput:** The number of samples processed per second. High throughput is essential for batch processing large volumes of claims.
- **Carbon Footprint (CO_2eq):** The estimated carbon emissions resulting from the energy consumption, calculated based on the carbon intensity of the local power grid.

By optimizing these metrics alongside predictive performance (F1-score, Accuracy), researchers can develop models that are not only intelligent but also environmentally and economically sustainable.

2.5 Data-Centric Optimization in Multimodal Systems

Multimodal fusion refers to the process of combining data from different sources, each contributing its own unique information to help create a richer, more comprehensive understanding of a problem. There are two main approaches to this: early fusion, where raw features from different modalities (like text, images, and audio) are combined before any modeling is performed, and late fusion, where separate models for each modality are built first and then their outputs are merged at the final stage. The central point is that each modality can add unique, task-relevant information, but the value of that information depends critically on its quality, alignment with other modalities, and how it is integrated [LT24].

The performance of a multimodal system is therefore limited by the data it receives. High-quality, well-annotated, and diverse datasets are essential for reliable performance: if one modality is noisy, biased, or poorly aligned with the others, the entire model can inherit those flaws. This exemplifies the “garbage in, garbage out” principle within the context of multimodal learning. In practical terms, noisy textual descriptions, poorly framed photographs, or systematic omissions in annotation (for example, missing minor scratches or inconsistent severity labels) can bias the model, reduce generalization to new conditions, and increase costly operational failures. Consequently, careful attention to dataset curation and annotation protocols is as important as architectural choices when building robust multimodal systems.

2.5.1 Dataset Limitations and Generalization

One of the biggest challenges in fields like vehicle damage assessment is the scarcity of large, balanced, and well-annotated datasets. Public and proprietary collections frequently overrepresent common, visually salient damage types (scratches, small dents) and underrepresent rare but operationally important cases such as frame deformation, subframe fractures, or engine-bay damage. This class imbalance leads models to prioritize majority classes during training and to underperform on examples that, although rare, can have disproportionate cost implications for insurers.

For example, in the domain of vehicle damage detection, most datasets may only contain a handful of examples of more serious damage like frame cracks or engine component failures, leading the model to underperform when tasked with identifying those types of issues in the real world.

Generalization challenges are not unique to vehicle damage models. Wang et al. [WMMX17] demonstrate that limited or biased multimodal training data can cause models to overfit to spurious attributes (e.g., specific speaker traits) and propose a method to boost generalization across domains.

Crucially, in multimodal systems, diverse and well-rounded datasets are needed to ensure that models can generalize well to new, unseen data. Otherwise, they will become overly proficient at memorizing the training set and fail to perform when faced with real-world complexity, a classic case of overfitting.

2.5.2 Why Quality Beats Quantity

The conventional belief that "more data is always better" is being revised in light of recent evidence favoring data quality. Carefully curated datasets, even when smaller, can produce superior downstream performance and markedly greater data efficiency than massive, noisy corpora.

Xu et al. [XSW$^+$25] argue that simply having a vast amount of data from web crawls is not enough for optimal multimodal model performance. More important is adopting a data-centric approach that centers on curation, cleaning, and focusing on data that are truly useful for the model.

2.5.3 Cleaning and Consistent Annotation

Data cleaning and consistent annotation are the foundation of any data-centric effort. Practical cleaning steps include deduplication, outlier detection, label normalization, and explicit alignment of multimodal pairs. Multimodal systems benefit disproportionately from explicit alignment checks. Ensuring that the textual description refers to the visual content improves signal quality for joint learning [GLX$^+$23].

Investing in cleaning and annotation not only leads to better model performance, but also makes the training process more efficient. Well-annotated, high-quality data can significantly reduce the time and computational resources required for model training and optimization, which in turn lowers the overall costs. Thus, improving data quality pays off at every stage of the entire AI development pipeline [GLX$^+$23].

This chapter has presented the essential background on multimodal learning, recent developments in VLM, techniques used in automated vehicle damage assessment, and strategies for improving efficiency and data quality. These elements are critical to understanding the methods and innovations proposed in the remainder of this thesis.

CHAPTER 3

Related Work

This chapter provides an overview of related work in the field of energy efficiency optimization within multimodal learning systems, specifically in the context of automated vehicle damage evaluation. The chapter begins by discussing strategies aimed at reducing the computational cost of ML models, such as model compression, PEFT, and energy measurement techniques.

Next, classical computer vision approaches and multimodal learning approaches in the context of automated vehicle damage evaluation are explored, highlighting key studies that incorporate image recognition to assess vehicle condition. Finally, work that intersects energy optimization and multimodal systems is reviewed, identifying common strategies and challenges.

3.1 Energy-Efficient Machine Learning

The rapid growth of deep learning has delivered remarkable breakthroughs, from conquering the game of Go to driving top-tier results in image recognition, speech processing, and language translation. However, this leap forward has come at a significant increase in the consumption of computing power. This trend is becoming economically, technically, and environmentally unsustainable [TGLM20], motivating a broad research effort to retain or recover performance while lowering compute cost. Work in this area spans algorithmic compression, compact adaptation, dynamic compute, and better measurement methodologies, each addressing different parts of the training and inference lifecycle.

Beyond model compression, recent work has focused on optimizing the training process itself. You et al. introduced Zeus [YCC23], an optimization framework that automatically tunes training hyperparameters to minimize energy consumption rather than just training time. This work demonstrates that energy-aware training schedules can significantly reduce the carbon footprint of deep learning without requiring changes to the model architecture.

3.1.1 Model Compression Techniques

Recent research has extended classical compression methods to the multimodal and VLMs setting, where the joint representation of text and images introduces new interaction patterns and hardware considerations. For instance, CASP, introduced by Gholami et al. [GACZ25], exploits attention sparsity in large multimodal architectures by combining low-rank decompositions of the query and key weight matrices with optimal per-layer bit-allocation quantization. This approach achieves state-of-the-art performance in ultra-low-bit regimes across both image-language and video-language benchmarks.

Paula et al. [PSUL25] provide a comparative study across pruning, quantization, and distillation, showing that the combination of these methods can reduce training and inference costs by up to an order of magnitude while preserving competitive performance.

3.1.2 Parameter-Efficient Fine-Tuning with Low-Rank Adaptation and Quantized Low-Rank Adaptation

A widely adopted variant of PEFT is LoRA [HSW+22], introduced by Hu et al. LoRA takes advantage of the insight that weight updates in neural networks often lie in a low-dimensional subspace. Instead of learning full-rank updates for weight matrices, LoRA approximates these updates by decomposing them into the product of two low-rank matrices. During fine-tuning, only these small matrices are trained while the primary weights remain unchanged, resulting in substantial reductions in trainable parameters without sacrificing downstream performance.

Based on the principles of LoRA, QLoRA [DPHZ23] presents a further improvement in PEFT by integrating model quantization into the LoRA framework. QLoRA keeps the model frozen while quantizing it to 4-bit precision. This hybrid quantization strategy dramatically reduces both memory and computational demands, allowing fine-tuning of massive models (up to 65 billion parameters) on a single GPU with only 48GB of Video Random Access Memory (VRAM). By merging the low-rank adaptation of LoRA with aggressive yet effective quantization techniques, QLoRA makes the fine-tuning of ultra-large LLMs more accessible and efficient [DPHZ23].

3.1.3 Energy Measurement and Benchmarking

At the hardware level, energy consumption can be precisely monitored using NVML [NVI25]. NVML is a C-based API that provides direct access to low-level management and monitoring functionalities of NVIDIA GPUs. It enables developers to retrieve real-time metrics such as power draw, temperature, memory usage, and utilization. By integrating NVML into training and inference pipelines, researchers can collect accurate measurements of GPU energy consumption across different stages of model execution. These data can then be aggregated to compute total power usage and benchmark the energy efficiency of various model architectures or optimization techniques. Unlike estimation-based tools, NVML reports actual power data from on-board GPU sensors, ensuring high-fidelity energy profiling for AI workloads.

3.2 Classical Computer Vision Approaches

Early and recent studies on vehicle damage often start with single-image approaches such as classification, detection, or segmentation. One of the most widely used resources is the CarDD dataset [WLW23], which contains 4,000 high-resolution images and more than 9,000 annotated damage instances of six types, such as scratches, dents, and cracks. CarDD offers annotations for segmentation, detection, and classification, making it a versatile resource for comparison and evaluation of methods. However, the dataset also exposes the core limitations of single-image analysis. For example, small, subtle, or partially obscured damage types often elude even high-performing models, especially when lighting conditions, occlusions, or surface reflections degrade visual clarity [WLW23].

A study by Perez et al. demonstrates that classical single-image damage detection methods can be significantly enhanced by ensemble learning [PZCGPM+24]. They introduce an ensemble of ten YOLOv5-based detectors, each trained with varying configurations, and apply confidence threshold optimization and box fusion through a voting system. This approach markedly improves detection precision and recall while dramatically reducing false positives.

However, even the best classical models are ultimately constrained by their lack of contextual understanding and temporal coherence. More recent multimodal or sequential approaches aim to address these limitations.

3.3 Multimodal Learning for Vehicle Damage Evaluation

Hasan et al. propose GroundingCarDD [HNO+24], a text-guided multimodal phrase-grounding framework specifically for car damage detection and localization. Their approach focuses on the fusion of visual features with textual phrases (e.g., “front bumper scratch”, “left rear dent”) using a context-aware attention mechanism that strengthens correspondences between language tokens and localized image regions. This targeted fusion enables the model to separate subtle defects from confounding visual artifacts such as reflections, shadows, or background clutter. Evaluation on a mix of curated and public datasets demonstrates meaningful gains over competitive baselines [HNO+24].

Asgarian et al. present AutoFraudNet [ASJP23], a multimodal reasoning framework for detecting fraudulent auto-insurance claims. The model integrates multiple input sources, through a cascaded slow-fusion strategy employing BLOCK-Tucker fusion blocks [BCTC19], which balance model capacity and overfitting control. Using pre-extracted image and text features, AutoFraudNet demonstrates that multimodal fusion produces superior fraud detection performance, achieving over a 3% absolute improvement in PR-AUC compared to unimodal baselines. Although optimized for fraud detection, the framework also illustrates how outputs from image-based damage localization and classification models can be effectively incorporated into downstream multimodal claim validation tasks [ASJP23].

Yang et al. propose the AIML [YCD+23] framework for auto-insurance fraud detection, which explicitly integrates computer vision, natural language processing, and knowledge-based features. The approach draws on visual evidence, such as detected damaged parts and Optical Character Recognition (OCR)-extracted information from invoices or license plates, transforming these inputs into structured representations. These are then fused with textual claim descriptions and tabular claim fields to capture inconsistencies indicative of fraudulent activity. Evaluation in real-world insurance claim collections demonstrates that incorporating multimodal inputs yields consistent improvements over models trained exclusively on structured or tabular data [YCD+23].

In the broader context of industrial inspection, Gu et al. proposed AnomalyGPT [GZZ+23], a method that adapts Large VLMs for industrial anomaly detection. By utilizing a lightweight visual-textual feature matching mechanism, their approach allows the model to detect and describe surface defects (such as scratches and dents) with very few training examples. Although applied to general industrial manufacturing, the core problem of identifying surface irregularities is directly analogous to vehicle damage assessment, suggesting that similar few-shot VLM techniques could be effective in the insurance domain.

3.4 Energy Optimization and Multimodal Systems

Recent work has begun to address modality-sensitive efficiency considerations in VLMs. Li et al. introduce Modality-Balanced Quantization (MBQ) [LHN+25], a PTQ technique designed for large VLMs. The authors conclude that visual and textual tokens exhibit differing sensitivities to quantization, which can degrade performance when treated uniformly. To address this, MBQ applies modality-specific calibration strategies to balance quantization error between modalities and reduce reconstruction loss. The experimental results show that MBQ consistently outperforms previous PTQ methods, improving task precision by approximately 4.4% to 11.6% in various quantization settings [LHN+25].

Patnaik et al. provide a systematic survey of small VLMs (sVLMs), that is, VLM architectures optimized for deployment under computational constraints. The review categorizes approaches into transformer-based, hybrid, and lightweight designs, and examines efficiency techniques such as knowledge distillation, modality pre-fusion, and optimized attention mechanisms. The survey highlights the trade-offs inherent in compact architectures, particularly between accuracy, computational cost, memory footprint, and latency. In addition, it highlights key challenges for sVLMs, including limited generalization, increased sensitivity to bias, reduced robustness, and increased vulnerability to domain shifts [PNA+25].

3.5 Summary

Research that unifies multimodal design with energy optimization is still early but expanding. Shared ideas include modality-aware quantization, module-level compression, staged adaptation that uses parameter-efficient tuning to find task-relevant capacity before distilling models for deployment, and data curation that removes redundant or noisy samples. Work that blends these techniques often reports the strongest real-world gains, showing that small, targeted architectural or algorithmic adjustments can sharply improve energy per prediction with little loss in accuracy.

Overall, recent progress in multimodal grounding and fusion shows the value of combining text and image signals for tasks such as damage localization, fraud detection, and claim validation. At the same time, efficiency research on quantization, parameter-efficient tuning, distillation, and compact vision-language model design offers practical ways to run these systems under strict compute and energy limits. Key gaps persist, including the lack of unified energy-performance models for multimodal compression, inconsistent evaluation practices, and limited curated datasets for vehicle damage analysis. Although AnomalyGPT advances defect detection, no prior work combines QLoRA-based fine-tuning with energy-aware data curation and compression for the insurance domain. This thesis fills that gap by creating a curated multimodal benchmark, applying staged PEFT and modality-aware compression, standardizing energy measurements, and modeling energy vs. accuracy trade-offs.

CHAPTER 4

Methodology

This chapter provides an outline of the experimental methodology used to investigate the potential of VLMs for automated vehicle damage assessment under data and energy constraints. First, the data collection process is described, which involves gathering data from multiple public sources and a subsequent pre-processing pipeline designed to ensure quality and consistency.

Next, the models selected for benchmarking, namely LLaVA, Qwen-VL, Bunny, Phi-3, and SmolVLM, are detailed, and the justification for their inclusion is explained. Detailed descriptions of these models, including their architectural differences and unique features, are provided in Chapter 2. In this chapter, the focus is on how these models were employed in the benchmarking and fine-tuning pipeline.

The benchmarking strategy used to evaluate the model performance is described, including the metrics chosen and the integration of NVIDIA NVML for energy tracking. The fine-tuning procedure applied to the top-performing model is then outlined, focusing on domain adaptation techniques and training optimizations.

Finally, the post-fine-tuning evaluation strategy is presented, in which the comparison of the fine-tuned model against its baseline counterpart is outlined.

4.1 Data

Datasets were identified and acquired from two widely recognized platforms that serve as central repositories for ML and computer vision research:

Kaggle: A platform well known for hosting ML competitions, but also providing a substantial dataset repository contributed by researchers and practitioners. Kaggle datasets are frequently used in applied research, including tasks such as object detection, classification, and anomaly detection. In the context of this study, datasets focused on vehicle damage detection and severity classification were particularly relevant, as they provide annotated images that support model benchmarking and evaluation[Kag25].

Roboflow: A platform specializing in the preparation, annotation, and hosting of datasets for computer vision applications. Roboflow allows researchers to preprocess, augment, and share image datasets through an intuitive interface. The curated automotive datasets available on this platform are specifically designed to support detection and classification tasks, making them highly applicable to this research[Rob25].

By integrating datasets from these two sources, diversity and reproducibility are ensured.

4.1.1 Datasets

A set of publicly available datasets was selected that together provide broad coverage of vehicle imagery and a variety of damage types. For each dataset below, its origin, content, typical annotation format, and how it was used in this study are summarized.

Car Connection Picture Dataset

The Car Connection Picture [Pro17] dataset consists of over 60,000 images of cars sourced from Kaggle. The dataset includes a diverse collection of car images from various models and types. The images feature cars in good condition and in various environments, such as urban, highway, and parking lots. Images are provided in JPEG format at a range of resolutions. In this project, images from this dataset were used primarily to populate the category "No damage": their visual variety assists models in learning non-damage appearance modes and reduces false positives during detection and classification.

Car Damage Assessment

The Car Damage Assessment [Man21] dataset, also from Kaggle, comprises 1512 images depicting vehicles with different degrees of damage. It covers a range of car models and types, categorizing damage into specific classes such as:

- Unknown
- Door dent
- Bumper scratch
- Door scratch
- Glass shatter

The images are supplied in JPEG format and are available in a range of resolutions. The corresponding category information is detailed in the accompanying CSV file. This dataset was used as a primary source of small-to-moderate severity damage examples.

Car Damage Images Computer Vision Dataset

Obtained from Roboflow, the Car Damage Images Computer Vision [Dam23] dataset comprises 300 images illustrating various damage scenarios. It encompasses diverse vehicle models and damage types, such as:

- Broken glass
- Dent
- Misaligned part
- Missing part
- Paint damage
- Scratch
- Front-end damage
- Rear-end damage
- Side-impact damage

Provided as JPEG files with varying resolutions, this dataset includes category labels in a CSV file. This dataset provided fine-grained examples for several damage modes that are underrepresented in larger repositories.

Car Damage Detection Computer Vision Model

Another Roboflow-sourced collection, the Car Damage Detection Computer Vision Model [CAP23] dataset, contributes 3226 images showing cars with varying damage levels. It spans multiple car models and damage categories, notably:

- Bonnet
- Bumper
- Dickey
- Door
- Fender
- Light
- Windshield

The images are in JPEG format across different resolutions, with metadata and labels provided in a CSV file.

CAR DAMAGES Computer Vision Dataset

The CAR DAMAGES Computer Vision [rod22] dataset, also from Roboflow, offers 839 images highlighting different damage severities. It includes a variety of car types and specific damage classifications like:

- Crack and hole

- Medium deformation
- Severe deformation
- Severe scratch
- Slight deformation
- Slight scratch
- Windshield damage

These JPEG images come in mixed resolutions, accompanied by a CSV file containing the necessary category information.

Car-Dent-Detection2 Computer Vision Dataset

The Car-Dent-Detection2 Computer Vision [Saw24] dataset consists of 3746 images of cars with varying levels of damage sourced from Roboflow. The images feature cars from various models, types, and damage categories, including:

- Front Windscreen Damage
- Headlight Damage
- Major Rear Bumper Dent
- Rear Windscreen Damage
- Running Board Dent
- Side Mirror Damage
- Signal Light Damage
- Taillight Damage
- Bonnet Dent
- Door Outer Dent
- Fender Dent
- Front Bumper Dent
- Medium Bodypanel Dent
- Pillar Dent
- Quarter Panel Dent
- Rear Bumper Dent
- Roof Dent

The images are supplied in JPEG format and are available in a range of resolutions. The corresponding category information is detailed in the accompanying CSV file.

Car-Defect-2 Computer Vision Dataset

With 3356 images, the Car-Defect-2 Computer Vision [R22] dataset from Roboflow provides a substantial collection of damaged vehicle imagery. It covers diverse car models and specific defect types such as:

- Front Windscreen Damage

- Headlight Damage
- Rear Windscreen Damage
- Running Board Dent
- Side Mirror Damage
- Signal Light Damage
- Taillight Damage
- Bonnet Dent
- Door Outer Dent
- Fender Dent
- Dent
- Medium Bodypanel Dent
- Pillar Dent
- Quarter Panel Dent
- Rear Bumper Dent
- Roof Dent

Images are provided in JPEG format with mixed resolutions, and the corresponding annotations are detailed in a CSV file.

Car-Dects-New Computer Vision Dataset

The Car-Dects-New Computer Vision [ag21] dataset, another Roboflow contribution, comprises 3756 images depicting various vehicle damages. It features a wide range of car types and damage categories, including:

- Front Windscreen Damage
- Headlight Damage
- Major Rear Bumper Dent
- Rear Windscreen Damage
- Running Board Dent
- Side Mirror Damage
- Signal Light Damage
- Taillight Damage
- Bonnet Dent
- Door Outer Dent
- Fender Dent
- Front Bumper Dent
- Medium Bodypanel Dent
- Pillar Dent
- Quarter Panel Dent
- Rear Bumper Dent
- Roof Dent

The images are supplied in JPEG format and are available in a range of resolutions. The corresponding category information is detailed in the accompanying CSV file.

CarDD

Sourced from Google Drive, the CarDD [WLW23] dataset contains 4000 images illustrating different levels of vehicle damage. It encompasses a variety of car models and damage types, such as:

- Tire flat
- Lamp broken
- Glass shatter
- Scratch
- Crack
- Dent

These images are provided as JPEGs in various resolutions, with detailed category information available in the accompanying CSV file.

4.2 Data Preprocessing and Quality Control

Vehicle damage imagery varies substantially in quality due to factors such as lighting conditions, image resolution, motion blur, and the capabilities of user devices. Careful preprocessing is therefore essential to produce a reliable training corpus and to avoid expending computational resources on low-utility samples. By filtering out low-quality images before training, the model is prevented from unnecessarily expending GPU cycles attempting to learn features from uninformative samples, effectively increasing the "information density" per Joule of energy consumed. The following subsections describe the automated filtering and quality-control steps applied to produce the curated dataset used in the experiments.

4.2.1 Filtering Procedures

The following preprocessing steps were implemented to create the cleaned corpus:

Blur Detection - Images with insufficient sharpness were discarded. A variance of the Laplacian operator [Hua25] was used as a sharpness score.

The Laplacian operator is a second-order derivative operator commonly used in image processing to highlight regions of rapid intensity change, such as edges. Mathematically, the Laplacian of a two-dimensional image function $f(x, y)$ is defined as the sum of the second partial derivatives with respect to spatial coordinates, as shown in Equation (4.1):

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{4.1}$$

In OpenCV, the discrete Laplacian operator is implemented using convolution with kernels approximating these second derivatives. When applied to a grayscale image I, the Laplacian highlights areas where the intensity changes sharply. This response is often used for blur detection by computing the variance of the Laplacian response, defined in Equation (4.2):

$$\text{Blur Score} = \text{Var}\left(\nabla^2 I\right) \tag{4.2}$$

A high variance indicates a sharp image with many edges, whereas a low variance suggests blurriness. Thus, the variance of the Laplacian serves as an effective quantitative measure of image sharpness and is commonly computed in OpenCV as in Equation (4.3):

$$\texttt{cv2.Laplacian}(I, \texttt{cv2.CV\_64F}).\texttt{var}() \tag{4.3}$$

where the function computes the Laplacian of the image I and then the variance of the result [Hua25]. This method helps to ensure that important structural details, such as dents, cracks, or scratches in vehicle damage images, remain visible.

Exposure Control - To ensure high-quality data, images with extreme exposure levels were excluded. These issues can hinder the performance of computer vision models by affecting the quality of feature extraction [EYEW22]. For exposure checks, each image was converted to grayscale and the mean intensity was computed. Images with a mean intensity below 30 were flagged as underexposed and those above 225 as overexposed. These numeric thresholds were validated on a curated subset to balance the removal of unusable images against retaining naturally dark or bright but informative examples (for instance, nighttime images where damage is still visible).

Near-Duplicate Removal - To avoid redundancy and minimize computational waste, perceptual hashing was applied to identify visually similar images. The Average Hash (aHash) [FJZL17] algorithm was used, which captures low-frequency image structure and is efficient for large corpora. The aHash computation proceeds as follows:

1. Resize the input image to 8×8 using bicubic interpolation, ignoring the original aspect ratio.
2. Convert the resized color image to grayscale and compute the mean pixel value M.
3. Binarize each pixel value A_i (where $i = 1, \ldots, 64$) using the following Equation (4.4):

$$h_i = \begin{cases} 0, & \text{if } A(i) \leq M, \\ 1, & \text{otherwise.} \end{cases} \tag{4.4}$$

4. Combine the binary values h_i to form a 64-bit hash code.

To measure the similarity between two images with hashes, the Hamming distance [Ham50] is computed. The Hamming distance between two strings of equal length is the number of positions in which the corresponding symbols are different [Ham50]. A small Hamming distance indicates high similarity between images. In the pipeline, pairs of images with Hamming distance less than two were considered near-duplicates; in such cases the earliest instance (sorted by acquisition or ingestion time) was retained and subsequent duplicates were removed.

Annotation Consistency Checks - Automated checks were used to validate annotation files and remove invalid or empty annotations. Annotation files labeled as `Other`, or those that were empty or contained only whitespace, were identified and removed together with their corresponding images. The cleaning pipeline also checked CSV manifest fields for missing bounding boxes, malformed coordinates, or inconsistent class labels and flagged these entries for manual review. These systematic checks ensured that only samples with valid, non-empty captions and correct annotation formats were retained, reducing downstream noise without requiring wholesale manual re-labeling.

After preprocessing, the size of the dataset was reduced from 24,800 to 18,200 high-quality samples. Examples of annotated images can be seen in Figure 4.1. The adopted filtering approach ensures a trade-off between dataset scale and reliability. Both the raw and preprocessed datasets are retained and versioned so that experiments can compare robustness and performance when models are trained on noisy versus curated data.

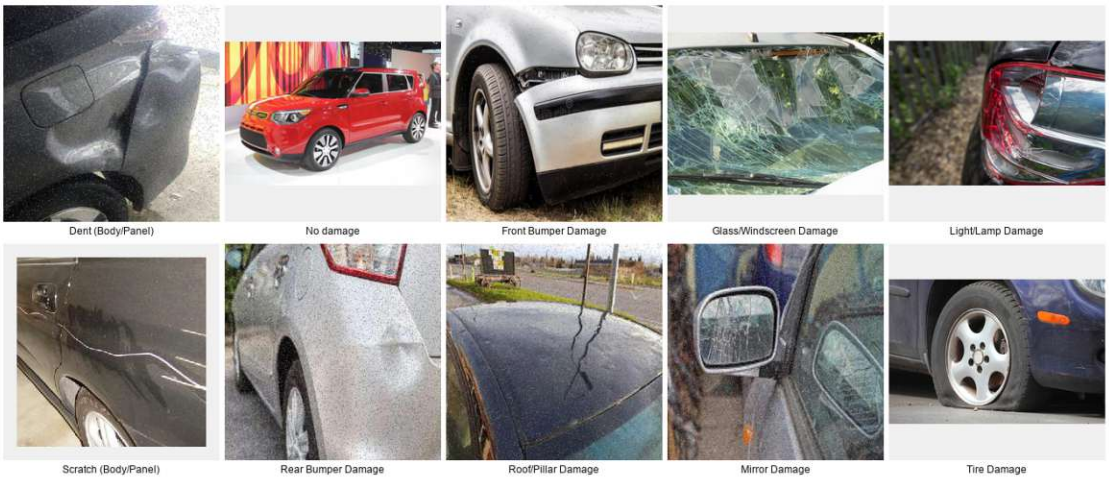


Figure 4.1: Examples of the annotated images retained after the data filtering process.

4.3 Model Selection

Given the focus of this thesis on benchmarking fine-tuning strategies under efficiency constraints, five state-of-the-art VLMs were selected that represent a broad range of trade-offs among parameter count, architectural design, and intended deployment profile.

The selection intentionally spans large instruction-tuned models with strong multimodal alignment, family releases that emphasize high-quality pretraining data and grounding, and compact architectures built for low-resource inference.

LLaVA-v1.5-7B: Known for its instruction-following capabilities and strong alignment between vision and language modalities. LLaVA has been shown to reach the GPT-4 level in multimodal tasks [LLLL24]. Table 4.1 provides a technical overview of the LLaVA-1.5-7B model.

Table 4.1: LLaVA-1.5-7B Model Specifications

Attribute	**Details**
Model Name	LLaVA-1.5-7B
Backbone	Vicuna-7B (fine-tuned from LLaMA)
Vision Encoder	Contrastive Language-Image Pre-training (CLIP)-based (e.g., Vision Transformer (ViT)-L-14)
Number of Parameters	Approximately 7 billion
Training Datasets	558K filtered image-text pairs from LAION/CC/SBU captioned by BLIP; 158K GPT-generated multimodal instruction-following data; 450K academic-task-oriented VQA mixture; 40K ShareGPT data.
Training Framework	Fine-tuned using multimodal instruction-following data

Qwen-VL-7B: A versatile model developed by Alibaba that incorporates a structured training pipeline across multiple multimodal datasets. It has achieved leading results in a range of visual understanding benchmarks [BBC+23]. Table 4.2 provides a technical overview of the Qwen2.5-VL-7B-Instruct model.

Bunny-Llama-8B: A lightweight VLM family designed for efficiency. Bunny employs smaller parameter counts with optimized encoders, making it a strong candidate for resource-constrained environments where energy and latency matter [HLW+24]. Table 4.3 provides a technical overview of the Bunny-v1.1-Llama-3-8B-V model.

Phi-3.5-Vision-Instruct: A lightweight but highly capable VLM from Microsoft that fuses a CLIP-ViT-L/14 vision encoder with the Phi-3 Mini language backbone. It supports very long context (128K tokens) for reasoning-heavy image understanding tasks [AJA+24]. Table 4.4 provides a technical overview of the Phi-3.5-Vision-Instruct model.

SmolVLM-Instruct: A highly efficient multimodal model from Hugging Face that combines a shape-optimized SigLIP vision encoder with the SmolLM 2 language backbone. It offers strong image understanding capabilities while maintaining a compact size [MZF+25]. Table 4.5 provides a technical overview of the SmolVLM-Instruct model.

All models were initialized from publicly available pre-trained checkpoints to ensure comparability and reproducibility. Implementations and training were performed in PyTorch, and when official or community checkpoints differed in tokenizer, image preprocessing, or input template conventions these elements were standardized as part

Table 4.2: Qwen2.5-VL-7B-Instruct Model Specifications

Attribute	**Details**
Model Name	Qwen2.5-VL-7B-Instruct
Backbone	Qwen2.5 (fine-tuned from a variant of Qwen model family)
Vision Encoder	ViT
Number of Parameters	Approximately 7 billion
Training Datasets	1.5T tokens: image captions, visual knowledge, OCR data, 2T tokens: interleaved image-text data, VQA, multimodal math, agent tasks, video understanding, pure text, 0.6T tokens: long videos, long agent trajectories, long documents, >10,000 object categories: grounding data (bounding boxes and points, public + synthetic), 1M charts: synthesized using matplotlib, seaborn, plotly, 6M tables: real-world tabular data processed with end-to-end table recognition, Multilingual OCR: French, German, Italian, Spanish, Portuguese, Arabic, Russian, Japanese, Korean, Vietnamese (synthetic + real-world images)
Training Framework	Supervised fine-tuning with over 1 million samples and multistage reinforcement learning

Table 4.3: Bunny-v1.1-Llama-3-8B-V Model Specifications

Attribute	**Details**
Model Name	Bunny-v1.1-Llama-3-8B-V
Backbone	LLaMA (3B parameters) fine-tuned for multimodal tasks
Vision Encoder	CLIP-based vision encoder (ViT-L/14)
Number of Parameters	Approximately 8 billion
Training Datasets	2M image-text pairs (LAION-2B), 1.2M fine-tuned instruction data, 150K image-question pairs
Training Framework	Fine-tuned using high-quality data and dataset condensation techniques

Table 4.4: Phi-3.5-Vision-Instruct Model Specifications

Attribute	**Details**
Model Name	Microsoft Phi-3.5-Vision-Instruct
Backbone	Phi-3 Mini and Llama-2
Vision Encoder	CLIP ViT-L/14
Number of Parameters	4.2 billion
Training Dataset	Approximately 500 B tokens of interleaved image-text documents, FLD-5B image-text pairs, synthetic OCR/PDF, chart/table comprehension datasets, text-only data
Training Framework	Two-phase training: supervised fine-tuning + direct preference optimization

Table 4.5: SmolVLM-Instruct Model Specifications

Attribute	**Details**
Model Name	SmolVLM-Instruct
Backbone	SmolLM 2
Vision Encoder	SigLIP, shape-optimized, with 384×384 patches
Number of Parameters	1.7 billion
Training Dataset	Approximately 839 million tokens and 1.9 million images from The Cauldron and Docmatix datasets
Training Framework	Supervised fine-tuning on multimodal instruction data

of the experimental controls. The principal rationale for using pre-trained models is pragmatic and methodological: pre-training captures large amounts of world knowledge and multimodal structure, which dramatically reduces the data and compute required for effective domain adaptation. This allows the thesis to focus on fine-tuning strategies, data curation, and energy-aware interventions rather than end-to-end pretraining.

4.4 Benchmarking Strategy

To evaluate models comprehensively, standard predictive performance metrics are combined with sustainability-oriented indicators of computational cost. This dual approach enables a rigorous assessment of both the utility of each model and its environmental footprint under realistic training and inference conditions.

4.4.1 Predictive Performance Metrics

Model predictions are evaluated using four primary metrics derived from the confusion matrix counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Accuracy

Accuracy reflects the overall proportion of correctly classified instances. It is defined in Equation 4.5:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

Because accuracy can be misleading for imbalanced class distributions, Precision, Recall, and the F1 score are reported to give a clearer picture of positive-class behavior [THKG20].

Precision

Precision measures the fraction of positive predictions that are actually correct. It is defined in Equation 4.6:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.6}$$

Recall

Recall measures the proportion of actual positive instances that are correctly identified. It is defined in Equation 4.7:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.7}$$

F1 score

The F1 score represents the harmonic mean of precision and recall. It is defined in Equation 4.8:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.8}$$

Both macro-averaged (unweighted mean across classes) and weighted-averaged (weighted by class support) variants of these metrics are reported, and per-class results are shown.

4.4.2 Resource-Oriented Metrics

To complement accuracy-based evaluation, sustainability metrics are incorporated to quantify computational efficiency.

Energy consumption of the training process was measured using the NVML [NVI25], which provides low-level access to GPU power telemetry. NVML was used to log per-GPU energy usage throughout each training run, enabling transparent reporting of the computational footprint of every optimization step.

The system contained three GPUs in total: two remained idle for the entire duration of the experiments, while one GPU was actively used for training. The analysis therefore focuses on the energy consumption of the active GPU, while still recording the negligible baseline energy usage of the idle devices for completeness.

The combined use of predictive and resource-oriented metrics provides a comprehensive picture of each model's accuracy, robustness, latency, and environmental footprint.

4.5 Fine-tuning

This section describes the methodology used to fine-tune a VLM for the vehicle damage task using Unsloth runtime patches and parameter-efficient adapters (LoRA). The Unsloth library was selected because it provides optimized Triton kernels that significantly reduce VRAM usage (by up to 30%) and accelerate training throughput (by up to 2x) compared to standard implementations [HHt23]. This choice is directly aligned with the thesis objective of maximizing energy efficiency during the training phase.

It documents the intended inputs and outputs, the high-level flow of the fine-tuning script, the conditional steps that are taken depending on flags, and the checks and artifacts that ensure reproducibility. Experimental specifics (exact hyperparameters, split sizes, hardware, and full evaluation runs) are reserved for Chapter 5.

4.5.1 Fine-tuning workflow

Upon invocation, the script applies the Unsloth runtime patches and establishes default environment variables needed for the training process. It then parses all command-line arguments governing dataset configuration, I/O options, LoRA hyperparameters, training settings, and optional energy-tracking behavior. The dataset reader performs systematic validation of the CSV files. It resolves absolute image paths using the provided `-image-root` or the directory containing the CSV file, verifies that required columns are present, optionally truncates the dataset if limits are specified, and constructs a HuggingFace `Dataset` object. Rows referencing missing images trigger an immediate `FileNotFoundError` so that resource-intensive training runs do not begin under invalid conditions.

The data is transformed into the chat-style message format expected by the tokenizer and the data collator. Each user message embeds the corresponding image and, where applicable, auxiliary prompt text, while the assistant message contains the target answer. The chat template used for formatting is installed directly on the tokenizer and supplied to the `UnslothVisionDataCollator`, ensuring that tokenization and image-token placement remain consistent between training and inference.

Model initialization proceeds via `FastVisionModel.from_pretrained()` using Unsloth's optimized loader. When the `load_in_4bit` option is enabled, the default model identifier points to a quantized Unsloth-optimized checkpoint. LoRA adapters are attached using `FastVisionModel.get_peft_model()` in accordance with the user-specified configuration, including rank, scaling, and dropout parameters. The resulting model is placed into training mode through`FastVisionModel.for_training()` so that all Unsloth-specific optimizations are activated.

An `UnslothVisionDataCollator` is instantiated to perform all image transformations and batching routines. It enforces template consistency and controls vision-token placement. Training is then executed using `trl.SFTTrainer`, which receives all hyperparameters defined at the command line, including batch size, gradient accumulation steps, learning rate schedule, weight decay, warm-up configuration, precision mode, and the cadence of logging and checkpointing. Evaluation can be triggered at fixed steps or, alternatively, derived from a user-supplied number of evaluations per epoch.

Once enabled, GPU-energy usage is measured by an `EnergyMonitor` wrapper around NVML. The monitor records the cumulative energy counter for each tracked GPU exactly twice, once before inference begins and once after it ends, and reports the aggregate energy expenditure (plus any NVML warnings) at the conclusion of the run. As illustrated in Figure 4.2, energy monitoring operates alongside the broader data-preparation and model-loading stages of the training pipeline.

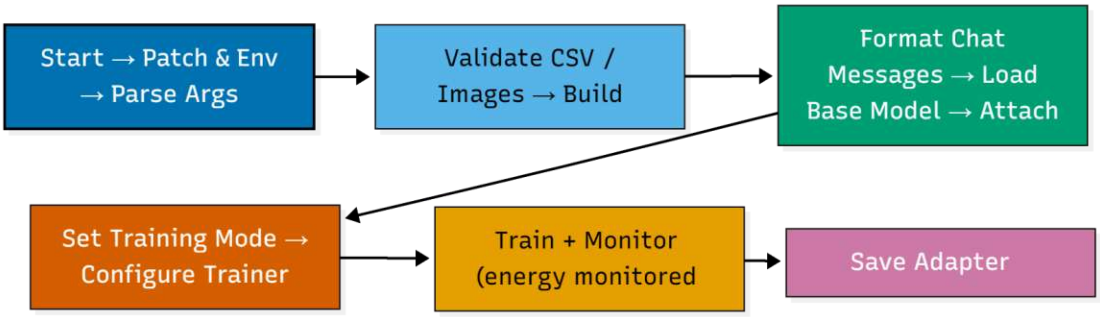

Start → Patch & Env → Parse Args
Validate CSV / Images → Build
Format Chat Messages → Load Base Model → Attach
Set Training Mode → Configure Trainer
Train + Monitor (energy monitored
Save Adapter

Figure 4.2: Training pipeline: data preparation, model loading, and training with continuous energy monitoring.

The training loop, executed through `trainer.train()`, is enclosed in a `try/finally` block to guarantee that energy monitoring is properly terminated and temporary resources are released, even in the event of user interruption or runtime errors. Once training concludes, the trainer state and metrics are saved. The LoRA adapter weights and tokenizer are written using `model.save_pretrained(output_dir)` and `tokenizer.save_pretrained(output_dir)`. When merged output is requested, the model is converted to inference mode using `FastVisionModel.for_inference(model)` and saved through `model.save_pretrained_merged()` to produce a fully merged 16-bit checkpoint. The post-training workflow, including optional merging and energy logging, is summarized in Figure 4.3.

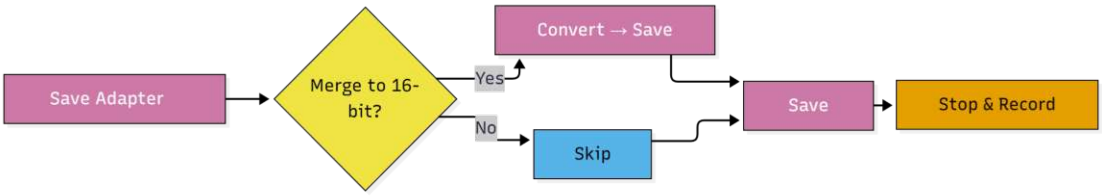


Figure 4.3: Post-training pipeline: adapter save point, optional 16-bit merge, metrics saving, and energy logging.

4.6 Post-fine-tuning evaluation

Following the fine-tuning process described in Section 4.5, the resulting models were subjected to the same rigorous evaluation framework established in Section 4.4. This evaluation phase was designed to assess the marginal performance gains and resource utilization differences compared to their original, pre-trained counterparts.

The evaluation primarily focused on two comparative analyses:

1. Performance Comparison: The fine-tuned models' Precision, Recall, and F1 scores were compared directly against the baseline models (e.g., zero-shot or few-shot inference) to quantify the improvement achieved by domain-specific training. This step confirmed the efficacy of the fine-tuning process.
2. Efficiency Analysis: The total energy consumption logged during the fine-tuning process (as measured by NVML) was reported alongside the training time and the resulting performance.

This allows for a critical assessment of the return on investment and whether the computational cost and energy expenditure of fine-tuning are justified by the observed uplift in predictive performance. By applying the dual metric system (F1 score and energy usage) consistently across both the baseline and fine-tuned models, any observed deviations in performance are directly attributable to the effects of the fine-tuning interventions.

CHAPTER 5

Experimental Setup

This chapter outlines the experimental setup used to evaluate the proposed methodology. The experiments systematically benchmark five state-of-the-art multimodal architectures with respect to predictive performance, computational efficiency, and energy consumption. The evaluation is conducted on both the raw and preprocessed variants of the dataset described in Chapter 4.

The chapter begins by describing the computational environment, detailing the specific hardware and software configurations that support the experiments. It then presents the benchmarking procedure, including a detailed overview of the dataset, its statistical properties, and the rationale for its selection. Following this, the training procedures are detailed, specifying model architectures, optimization strategies, hyperparameter settings, and training durations to ensure reproducibility. Finally, the evaluation protocols are explained, defining the performance metrics and validation strategies. Together, these components ensure that the experimental design is reproducible, transparent, and aligned with the overarching thesis objectives.

5.1 Experimental Environment

5.1.1 Hardware Configuration

All experiments were conducted on a high-performance GPU computing cluster with the following hardware specifications:

- **CPU:** AMD EPYC 9255 24-Core Processor (24 cores, 1 thread per core, max 3200 MHz)
- **Memory (RAM):** 755 GiB
- **Operating System:** Ubuntu 24.04.3 LTS

- **Storage:** High-speed NVMe storage (approx. 7 TB data volume + 1.7 TB system volume)
- **GPU:** 3 × NVIDIA RTX Workstation GPUs
  - 2 × NVIDIA RTX 4500 Blackwell (32 GiB VRAM each)
  - 1 × NVIDIA RTX 6000 Blackwell Max-Q Workstation Edition (96 GiB VRAM)
  - Driver: 580.95.05, CUDA: 13.0

This configuration provided sufficient computational capacity for VLM training and benchmarking.

5.1.2 Software Stack

The software environment was constructed to enable efficient, reproducible, and GPU-accelerated experimentation. The primary tools and libraries used included:

- **Language:** `Python 3.12.3`
- **Machine Learning Framework:** `Pytorch 2.9.0`
- **Fine-Tuning Utilities:** `PEFT`, `TRL`
- **Model Formats:** `safetensors`, `ONNX`
- **Vision / Image Handling:** `Pillow (PIL) 11.0.0`, `torchvision 0.24.0+cu128`
- **Data Handling:** `NumPy 2.1.2`

To ensure reproducibility, a fixed random seed ($seed = 2258$) was applied across Python, NumPy, and PyTorch random number generators. While complete determinism is challenging due to the asynchronous nature of CUDA kernels, these controls, combined with averaged runs, minimize experimental variance.

5.2 Benchmarking Dataset

The dataset was partitioned into training, validation, and test sets using a stratified splitting strategy with an 80:10:10 ratio. This approach ensures that the distribution of damage categories remains consistent across all splits, preventing bias where certain rare damage types might be underrepresented in the evaluation set. Figure 5.1 illustrates the distribution of labels across these splits, confirming that the stratification process successfully preserved the relative frequency of each category.

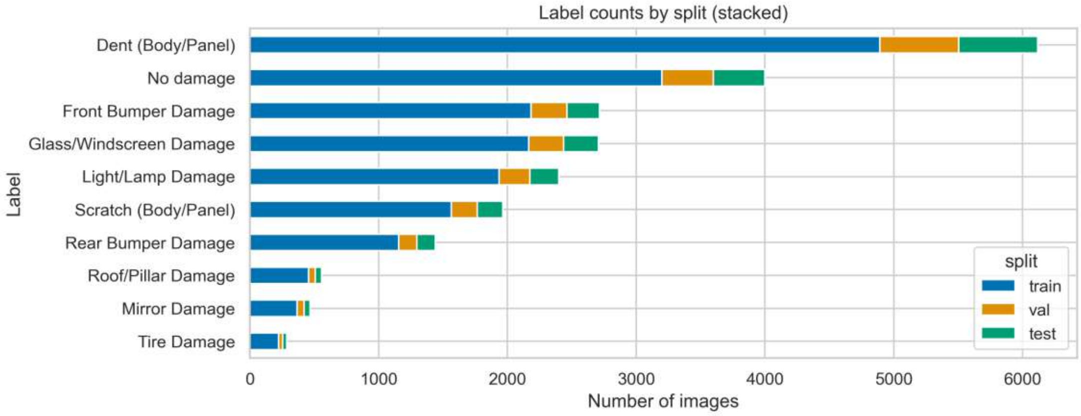


Figure 5.1: Horizontal stacked bar plot showing the distribution of labels across training, validation, and test splits in the dataset.

5.2.1 Rationale for Selection

This specific dataset was selected to challenge the multimodal capabilities of the target architectures beyond standard object recognition tasks. Unlike general-purpose benchmarks (e.g., COCO or ImageNet) which often feature distinct, centered subjects, vehicle damage assessment requires fine-grained visual understanding. The dataset presents three critical challenges for benchmarking:

1. **Multi-label Complexity:** A single image often contains multiple overlapping damage types (e.g., a “dent” and “scratch” on the same panel), requiring the model to perform multi-label classification rather than simple single-label prediction.
2. **Visual Ambiguity:** Distinguishing between visually similar categories (e.g., “glass shatter” vs. “headlamp broken”) tests the model’s visual discrimination and semantic grounding.
3. **Class Imbalance:** As is typical in real-world data, damage types are not equally distributed; common damages like scratches significantly outnumber rare events like fire, testing the models’ robustness to imbalanced training data.

5.2.2 Statistical Properties

To understand the data distribution, both the absolute volume of samples per class and their relative frequencies were analyzed. Figure 5.2 illustrates this distribution across the datasets. The left subplot displays the Absolute Count per class, highlighting the long-tail distribution inherent in the data. The right subplot shows the Relative Distribution (%), normalizing for dataset size to provide a clearer view of the density of each category relative to the whole.

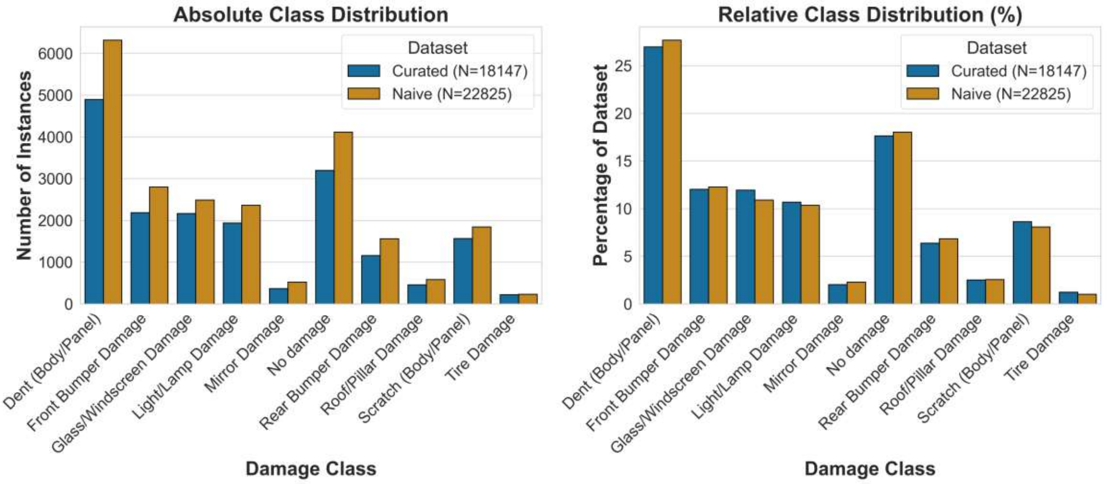
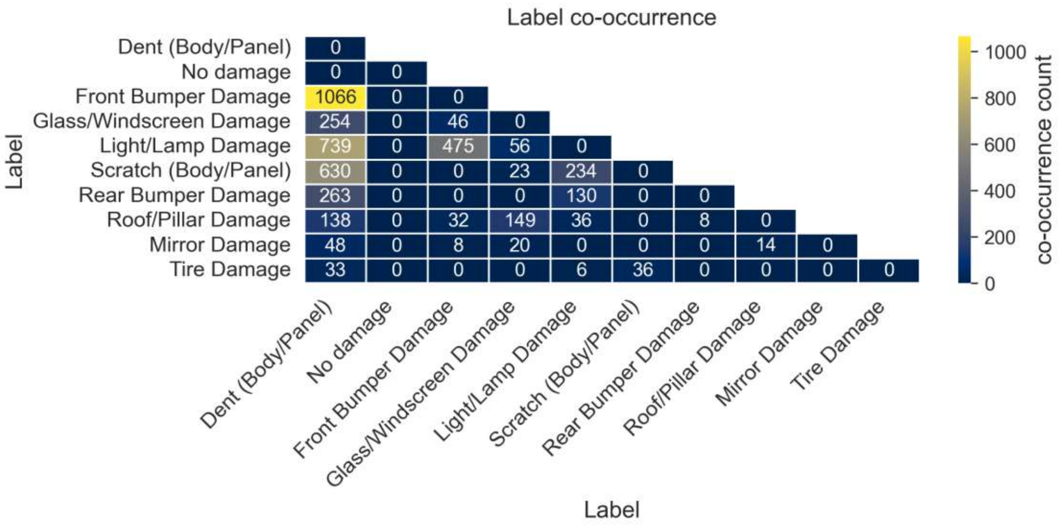


Figure 5.2: Class distribution across the datasets. The left subplot shows the Absolute Count per class, while the right subplot shows the Relative Distribution (%), normalizing for dataset size.

A key characteristic of this dataset is the co-occurrence of damage types. For instance, “dents” and “scratches” frequently appear together. Understanding these correlations is crucial for evaluating a model’s ability to capture complex, inter-related visual features. Figure 5.3 presents a heatmap of category co-occurrence, highlighting the frequency with which pairs of damage categories appear in the same image.



Label	Dent (Body/Panel)	No damage	Front Bumper Damage	Glass/Windscreen Damage	Light/Lamp Damage	Scratch (Body/Panel)	Rear Bumper Damage	Roof/Pillar Damage	Mirror Damage	Tire Damage
Dent (Body/Panel)	0									
No damage	0	0								
Front Bumper Damage	1066	0	0							
Glass/Windscreen Damage	254	0	46	0						
Light/Lamp Damage	739	0	475	56	0					
Scratch (Body/Panel)	630	0	0	23	234	0				
Rear Bumper Damage	263	0	0	0	130	0	0			
Roof/Pillar Damage	138	0	32	149	36	0	8	0		
Mirror Damage	48	0	8	20	0	0	0	14	0	
Tire Damage	33	0	0	0	6	36	0	0	0	0

Figure 5.3: Heatmap of category co-occurrence in the dataset, showing the frequency with which pairs of categories appear together.

5.3 Benchmark Preparation

Each model was initialized with its publicly available pre-trained weights. These models were not fine-tuned at this stage, enabling a direct comparison of their general-purpose performance on the vehicle damage dataset. All models were evaluated on the same test set under identical hardware and software conditions to ensure comparability. To reduce the effects of random variation (e.g., GPU scheduling, thermal fluctuations), each model's inference on the benchmark was repeated three times, and the arithmetic mean of each metric is reported.

5.3.1 Default Benchmark Run

The default benchmarking workflow ran batched inference for each evaluated vision-language model using a standardized procedure to ensure comparability. Each job was executed inside the project virtual environment with NVML/GPU telemetry initialized so that per-GPU power/energy traces were available. Inference consumed images referenced from a test CSV (resolved against the dataset root) and was executed on GPU 0 with a batch size of 1 to make per-sample latency and energy attribution straightforward. The inference output for every sample included both the raw model response and a normalized prediction token (the latter used for metric computation); these per-sample results were written to a CSV for downstream analysis. Energy telemetry was recorded for the active device and the monitored idle devices, so reported energy numbers explicitly include both active consumption and idle baseline unless otherwise stated.

5.3.2 Evaluation Step

A separate evaluation script then consumed the predictions CSV, aligned the normalized prediction column to the ground-truth column, and computed the classification metrics used throughout the thesis (accuracy, precision, recall, F1, along with per-class, macro, and weighted summaries). The evaluation preserved raw responses for qualitative error analysis and wrote a combined evaluation CSV containing predictions, ground truth, and computed metrics; energy and latency figures reported in the Results chapter correspond to the NVML logs collected during the matching inference run. This default run was the canonical, reproducible procedure used across all GPUs. The inference-evaluation workflow is illustrated in Figure 5.4.

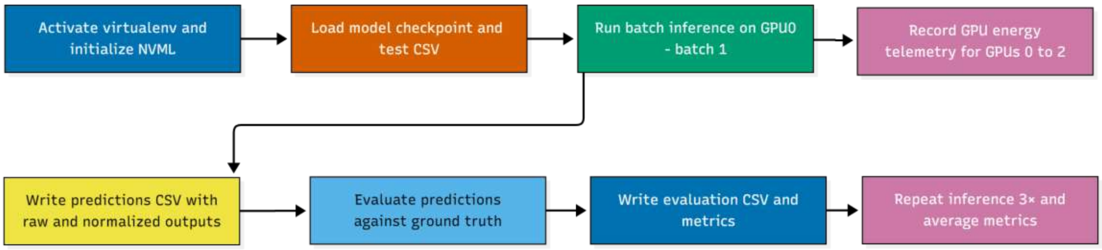

Activate virtualenv and initialize NVML
Load model checkpoint and test CSV
Run batch inference on GPU0 - batch 1
Record GPU energy telemetry for GPUs 0 to 2
Write predictions CSV with raw and normalized outputs
Evaluate predictions against ground truth
Write evaluation CSV and metrics
Repeat inference 3× and average metrics

Figure 5.4: Benchmarking inference and evaluation pipeline used for all vision-language models: environment and NVML initialization, batched inference with GPU telemetry, prediction persistence, and automated evaluation with repeated runs averaged for stability.

5.4 Fine-Tuning Setup

This section details the specific experimental runs executed using the fine-tuning methodology defined in Chapter 4. While the previous chapter outlined the mechanism of the training pipeline (e.g., Unsloth integration, LoRA adapter injection), this section specifies the data inputs and ablation configurations used to evaluate the impact of data quality and model adaptation strategies.

5.4.1 Dataset Variants: Naive vs. Curated

To quantify the value of the data preprocessing steps described in Chapter 4, parallel fine-tuning experiments were conducted on two distinct versions of the dataset:

- **Naive Dataset:** The raw, unmodified dataset containing all original images. This serves as a baseline to measure model performance in the presence of real-world data imperfections.
- **Curated Dataset:** The refined dataset resulting from the cleaning, re-stratification, and quality control processes.

Comparing the performance of the same model architecture across these two variants isolates the specific contribution of data quality to the final predictive accuracy.

5.4.2 Model Configuration and Hyperparameters

All fine-tuning experiments utilized the Qwen2.5-VL-7B-Instruct model as the base architecture. To ensure computational efficiency on the available hardware, the model was loaded with 4-bit quantization using the Unsloth framework. LoRA was utilized to update a small subset of parameters while keeping the pre-trained backbone frozen.

The global hyperparameters for all runs were standardized as follows to ensure fair comparability:

- **Epochs:** 1 (to prevent overfitting given the dataset size and pre-trained capabilities).
- **Evaluation Strategy:** 5 evaluations per epoch (to monitor convergence and loss dynamics granularly).
- **Batch Size:** 1 (with gradient accumulation to simulate larger effective batches).
- **Precision:** Mixed precision (bfloat16) for stability and speed.

5.4.3 Modality-Specific Ablation Studies

The locus of model adaptation was investigated by performing targeted ablation studies on the curated dataset. Figure 5.5 visualizes the three distinct training strategies executed using the Qwen2.5-VL-7B-Instruct architecture:

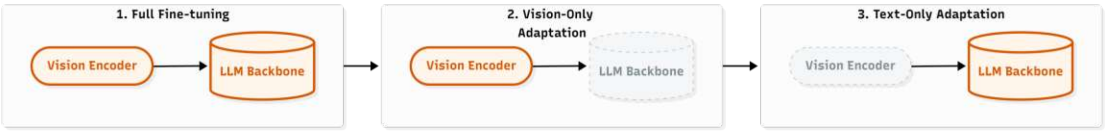


Figure 5.5: The diagram illustrates which components of the Qwen2.5-VL architecture (Vision Tower vs. LLM Backbone) are frozen (grey) and which are updated via LoRA adapters (colored) across the three experimental configurations: Full Fine-tuning, Vision-Only, and Text-Only.

1. **Full Fine-tuning (Standard):** LoRA adapters were applied to both the visual encoder and the language model backbone. This allows the system to simultaneously adapt its visual perception of damage and its textual alignment with the output format.
2. **Vision-Only Adaptation:** In this configuration, the language model was frozen, and adapters were trained only on the vision tower. This tests the hypothesis that domain adaptation is primarily a visual feature extraction task.
3. **Text-Only Adaptation:** Conversely, this run froze the vision tower and trained adapters only on the language model. This evaluates whether the pre-trained visual features are already sufficient and if the task is primarily one of semantic alignment.

5.4.4 Energy and Resource Monitoring

Consistent with the benchmarking protocols, energy consumption was tracked throughout the fine-tuning process. Although training was executed on a single active GPU (GPU 0), the energy monitoring system tracked power draw across all three available GPUs (0, 1, and 2) to account for system-wide overhead and idle power consumption during the training workload. This data provides the basis for the efficiency analysis presented in Chapter 6.

5.4.5 Model Merging and Export

Following the completion of the fine-tuning phase, the trained LoRA adapters were merged back into the base model weights to produce standalone artifacts. This step was essential to eliminate the runtime overhead of dynamic adapter loading during the extensive benchmarking phase and to ensure compatibility with standard Hugging Face inference pipelines.

The merging process involved:

1. Loading the base Qwen2.5-VL-7B-Instruct model in half-precision (float16) on the CPU to avoid VRAM fragmentation.
2. Loading the specific LoRA adapters (from the Naive, Curated, Vision-only, or Text-only runs).
3. Executing a `merge_and_unload()` operation to permanently fuse the adapter weights into the base model parameters.
4. Exporting the final fused model and processor as a standard Hugging Face checkpoint.

This resulted in fully independent float16 models for each experimental condition, which were then subjected to the standardized benchmarking procedure.

5.5 Post-Fine-Tuning Evaluation

To ensure a rigorous and direct comparison between the baseline pre-trained models and the fine-tuned variants, the post-fine-tuning evaluation strictly adhered to the standardized benchmarking workflow detailed in Section 5.3.

The merged models, produced as described in Section 5.4, were treated as direct replacements for the base architectures and were subjected to the identical inference and evaluation pipeline. Specifically:

- **Inference Consistency:** The fine-tuned models were evaluated on the same held-out test set, and executed on the identical hardware configuration. This ensured that the latency and energy profiles remained directly comparable to the pre-trained baselines.
- **Metric Calculation:** The same evaluation script was employed to compute all classification metrics (accuracy, precision, recall, F1) and to aggregate energy consumption statistics.

By holding the evaluation protocol constant, it is ensured that any observed deviations in performance are directly attributable to the effects of the fine-tuning interventions.

CHAPTER 6

Results and Discussion

This chapter presents the results of the experimental evaluations conducted to assess the proposed methodology. The experiments systematically analyze and compare multiple state-of-the-art VLMs with respect to their predictive performance, computational efficiency, and energy sustainability. The evaluation framework examines how each model performs under standardized experimental conditions, using both quantitative performance metrics and energy-based efficiency indicators.

Furthermore, the results provide a comparative analysis between the proposed solution and existing approaches, highlighting its operational effectiveness and environmental impact. The findings discussed in Chapter 6 thus offer insights into the balance between accuracy, efficiency, and sustainability in multimodal model design and deployment.

6.1 Comparison of VLMs on Damage Classification Benchmark

The initial phase of evaluation assessed the zero-shot capabilities of five distinct VLM architectures on the vehicle damage classification task. Table 6.1 summarizes the performance metrics and resource consumption for each model.

Model	**Time**	**GPU**	**Acc.**	**Prec.**	**Rec.**	**F1**
	(s)	**(kJ)**	**(%)**	**(M/W)**	**(M/W)**	**(M/W)**
Qwen-VL-7B	374.99	92.66	**57.99**	**63.28/70.22**	**53.53**/58.04	**55.68/61.05**
Qwen-VL-3B	368.07	**66.30**	32.56	49.66/61.09	44.98/35.04	39.12/36.77
LLaVA-1.5-7B	**282.96**	84.68	34.20	40.30/34.69	38.90/34.32	35.93/30.44
SmolVLM	432.52	102.14	22.74	52.37/55.12	34.31/25.02	28.02/21.75
Bunny-Llama-8B	551.16	165.29	55.88	61.84/67.82	51.75/**58.40**	53.63/60.15
Phi-3.5-Vision	1447.40	434.04	46.75	44.24/54.60	44.66/48.61	39.80/47.93

Table 6.1: Comparison of VLMs on Damage Classification Benchmark. Precision (Prec.), Recall (Rec.), and F1 Score (F1) are reported in the format Macro/Weighted (M/W). Best values in each column are shown in bold.

6.1.1 Predictive Performance Analysis

Qwen-VL-7B demonstrated superior performance among the evaluated architectures, achieving the highest scores across all predictive metrics. With an accuracy of 57.99% and a weighted F1-score of 61.05%, it demonstrated a robust ability to generalize to the complex, multi-label nature of car damage assessment. This suggests that the Qwen architecture's visual encoder preserves fine-grained spatial details more effectively than its counterparts.

Bunny-Llama-8B followed closely, securing the second-best position with an accuracy of 55.88% and a weighted F1-score of 60.15%. This indicates that the Llama-3 backbone provides strong reasoning capabilities, though it fell slightly short of Qwen in pure visual recognition tasks.

In contrast, LLaVA-1.5-7B and SmolVLM struggled significantly, with accuracies of 34.20% and 22.74%, respectively. The poor performance of LLaVA-1.5 is particularly notable given its popularity in general benchmarks; this discrepancy highlights the domain gap between general object recognition (e.g., "a car") and specific damage assessment (e.g., "rear bumper damage"). The model likely exhibits "hallucination" or an inability to distinguish between similar damage types without domain-specific fine-tuning.

6.1.2 Computational Efficiency and Latency

The inference time results reveal a non-linear relationship between model size and latency. Qwen-VL-3B and Qwen-VL-7B demonstrated highly efficient inference pipelines, completing the benchmark in 368.07s and 374.99s, respectively. Notably, doubling the parameter count from 3B to 7B resulted in a negligible latency increase (≈1.8%), suggesting that the Qwen architecture is highly optimized for parallel execution.

Conversely, Phi-3.5-Vision exhibited a prohibitive computational overhead, requiring 1447.40 seconds to complete the same task, which is nearly 4x slower than the Qwen models. This high latency, coupled with a substantial energy consumption of 434.04 kJ, renders the Phi-3.5 architecture unsuitable for real-time or edge-deployment scenarios in its current configuration.

6.2 Energy Efficiency Analysis

A core objective of this thesis is the evaluation of the sustainability of multimodal AI. To this end, "Green AI" metrics were analyzed, specifically focusing on the energy cost required to achieve a unit of performance. The efficiency ratio is defined as follows:

$$\text{Metric-per-kJ} = \frac{\text{Metric (\%)}}{\text{Total Energy Usage (kJ)}}$$

Table 6.2 presents these efficiency ratios.

Model	**Acc. per kJ (M/W)**	**Prec. per kJ (M/W)**	**Rec. per kJ (M/W)**	**F1 per kJ (M/W)**
Qwen-VL-7B	**0.626/0.626**	0.683/0.758	0.578/**0.627**	**0.601/0.659**
Qwen-VL-3B	0.491/0.491	**0.749/0.922**	**0.678**/0.529	0.590/0.555
LLaVA-1.5-7B	0.404/0.404	0.476/0.410	0.459/0.405	0.424/0.359
SmolVLM	0.223/0.223	0.513/0.540	0.336/0.245	0.274/0.213
Bunny-Llama-8B	0.338/0.338	0.374/0.410	0.313/0.353	0.325/0.364
Phi-3.5-Vision	0.108/0.108	0.102/0.126	0.103/0.112	0.092/0.110

Table 6.2: Energy efficiency comparison of VLMs on the Damage Classification Benchmark. Each cell shows the Metric-per-kJ value (% per kJ) for Macro/Weighted (M/W) variants. Best values in each column are shown in bold.

6.2.1 The Accuracy-Energy Trade-off

Figure 6.1 illustrates the trade-off between accuracy and energy consumption. Qwen-VL-3B exhibited superior efficiency metrics, achieving the highest Weighted Precision per kJ (0.922). While its raw accuracy was lower than the 7B variant, its energy footprint (66.30 kJ) was significantly smaller.

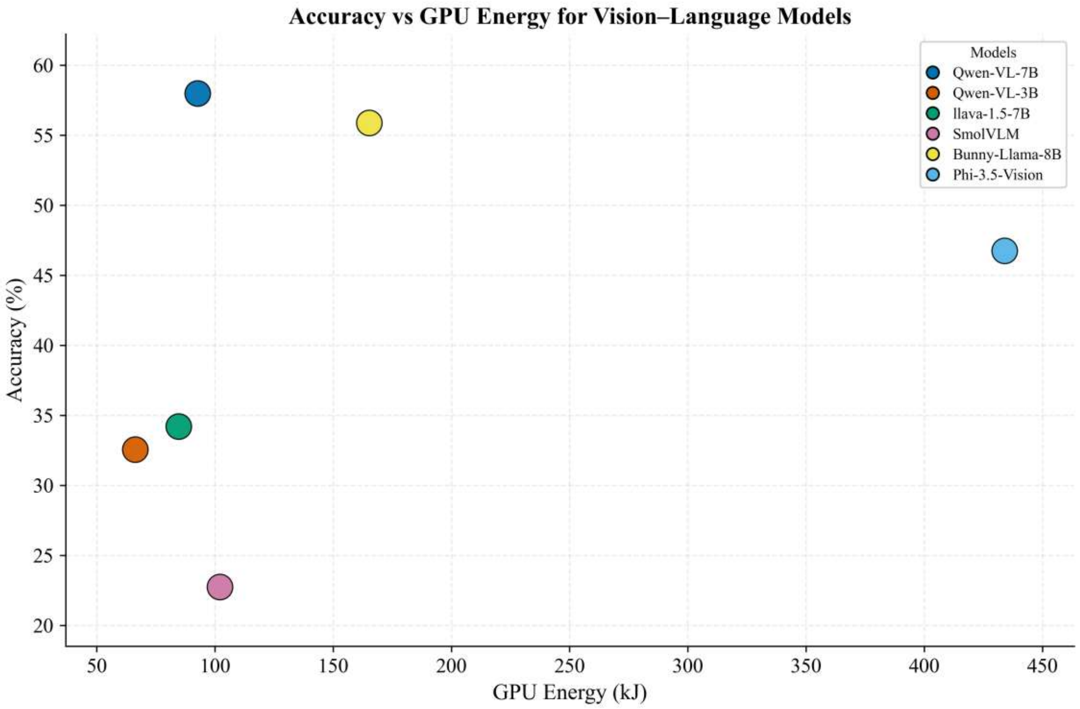


Figure 6.1: Relationship between model accuracy and total energy consumption (in kJ) across VLMs. Each point represents a model, illustrating the trade-off between predictive performance and computational energy efficiency.

Qwen-VL-7B represented a balanced choice, offering the highest raw performance while maintaining a respectable efficiency profile (0.626 Acc/kJ). In contrast, Phi-3.5-Vision demonstrated extremely poor efficiency (0.108 Acc/kJ), consuming disproportionate amounts of energy for marginal predictive gains. This result highlights the hidden environmental cost of unoptimized architectures; despite being a modern model, its operational carbon footprint was nearly 4x that of the best-performing model.

6.3 Comparison of Fine-Tuning Statistics

Based on the benchmarking results, the Qwen-VL-7B architecture was selected for fine-tuning. A series of experiments were conducted to isolate the impact of data curation and to determine whether the model learns primarily through visual adaptation or textual alignment. Table 6.3 details the training statistics.

6.3.1 Impact of Data Curation (Naive vs. Curated)

The comparison between “Naive” and “Curated” datasets provided strong evidence for the Data-Centric AI approach.

- **Convergence Speed:** The Curated Full run completed in roughly 58 minutes, whereas the Naive Full run required over 1 hour and 12 minutes. The curation process removed ambiguous and low-quality samples, allowing the model to converge faster.
- **Throughput:** The Curated dataset allowed for a slightly lower throughput (4.326 samples/s vs 4.416 samples/s), likely due to the overhead of processing more complex, valid samples, but the Total Energy consumed was significantly lower (963 kJ vs 1197 kJ) due to the reduced training duration.

6.3.2 Modality Ablation: Vision vs. Text

The ablation studies provided critical insight into the learning dynamics of VLMs:

1. **Text-Only Performance:** The "Text only" fine-tuning was notably the most efficient, completing in just 21 minutes with a substantially higher throughput of 11.8 samples/s. This is because backpropagation was restricted to the LLM layers, bypassing the computationally expensive vision tower. The low training loss (0.0622) suggests that the pre-trained visual features were already robust; the model primarily needed to learn the *vocabulary* of damage assessment rather than how to "see" the damage.
2. **Vision-Only Performance:** The "Vision only" fine-tuning consistently showed higher training loss (0.0833 for Curated, 0.0775 for Naive) compared to their Full or Text-only counterparts. This indicates that adapting the vision encoder alone is more difficult than adapting the language model. While the loss remained relatively stable, the higher values suggest that the visual features are harder to shift towards the domain specifics without the support of the language model's semantic flexibility.

Configuration	**Train Loss**	**Runtime (h:m:s)**	**Samples /s**	**Total Energy (kJ)**	**Duration (s)**
Curated Full	0.0595	0:58:24.19	4.326	963.66	3505.28
Curated Vision only	0.0833	0:57:07.50	4.422	934.99	3428.54
Curated Text only	0.0622	**0:21:22.68**	**11.817**	**355.39**	**1283.67**
Naive Full	**0.0536**	1:12:22.24	4.416	1197.59	4343.32
Naive Vision only	0.0775	1:10:35.95	4.526	1070.37	4237.05
Naive Text only	0.0567	0:27:19.65	11.694	449.43	1640.67

Table 6.3: Fine-Tuning Statistics for the Qwen-VL-7B Model. Best values (lowest Loss, Energy, and Duration; highest Samples/s) are shown in bold.

6.4 Impact of Fine-Tuning on Predictive Performance

Following the training phase, the fine-tuned models were evaluated on the classification benchmark. Table 6.4 presents a direct comparison between the baseline Qwen-VL-7B and the fine-tuned variants trained on Naive and Curated datasets.

Model Configuration	**Time (s)**	**GPU (kJ)**	**Acc. (%)**	**Prec. (M/W)**	**Rec. (M/W)**	**F1 (M/W)**
Baseline	374.99	92.66	57.99	63.28/70.22	53.53/58.04	55.68/61.05
Naive Fine-Tuned	370.05	93.07	62.74	66.90/75.49	56.38/62.76	59.44/66.77
Curated Fine-Tuned	371.33	93.39	**63.64**	**73.74/76.70**	**57.26/63.79**	**62.12/68.35**

Table 6.4: Performance comparison of Baseline vs. Fine-Tuned models. The Curated model demonstrates the impact of data quality on final predictive capability. Best values are shown in bold.

The results validate the data-centric AI hypothesis. The model trained on the curated dataset not only converged faster but also achieved superior predictive performance compared to the Naive baseline. This confirms that removing ambiguous and noisy samples helped the model form sharper decision boundaries.

6.5 Efficacy of Modality-Specific Adaptation

To understand the learning mechanism, the performance of the modality-specific ablation runs was compared. Table 6.5 details the F1 scores for full, vision-only, and text-only adaptation strategies.

Adaptation Strategy (Curated Dataset)	**Time (s)**	**GPU (kJ)**	**Acc. (%)**	**Prec. (M/W)**	**Rec. (M/W)**	**F1 (M/W)**
Full Fine-Tuning	371.33	93.39	63.64	73.74/76.70	57.26/63.79	62.12/68.35
Vision-Only	378.52	95.80	62.90	69.12/74.55	55.09/64.06	59.90/68.32
Text-Only	379.99	94.28	58.42	62.93/68.81	51.84/58.72	55.58/62.27

Table 6.5: Impact of modality-specific adaptation on model performance (Curated Dataset). Vision-Only adaptation drives the majority of the performance gain.

As shown in Table 6.5, the vision-only adaptation achieved an F1 score of 68.32%, which is statistically indistinguishable from the Full Fine-Tuning result (68.35%). In contrast, the text-only adaptation yielded a much more modest improvement (62.27%).

This indicates that the pre-trained visual encoder of Qwen-VL-7B, while powerful, required significant domain-specific adaptation to distinguish between the fine-grained damage categories (e.g., differentiating a “dent” from a “scratch”). The language model, conversely, required minimal adjustment to align with the output format. Thus, while text-only training is faster, vision-only training is the most effective strategy for maximizing predictive performance in this domain.

The disparity between vision-only and text-only performance highlights a fundamental characteristic of the damage assessment task: it is a perceptual challenge rather than a semantic one. The model does not struggle to understand the concept of a “dent” (semantic knowledge), but rather to identify the subtle visual cues, such as light distortion and surface irregularity, that constitute a dent in a 2D image (perceptual knowledge). Consequently, freezing the vision tower (as done in text-only adaptation) prevents the model from learning these necessary visual features, capping its performance regardless of how well the language model is tuned.

6.6 Operational Efficiency: Batching and Resolution Scaling

Beyond training efficiency, strategies to optimize inference energy consumption were investigated. The impact of two key variables was analyzed: batch size and input resolution.

6.6.1 Impact of Batch Sizes

A critical finding of this study was the increased resilience of the fine-tuned model to optimization techniques. As shown in Table 6.6, increasing the batch size from 1 to 4 significantly reduced the energy cost per sample.

Configuration	**Time (s)**	**GPU (kJ)**	**Acc.**	**Prec.** **(M/W)**	**Rec.** **(M/W)**	**F1** **(M/W)**
Baseline (BS=1)	374.99	92.66	57.99	63.28/70.22	53.53/58.04	55.68/61.05
Baseline (BS=4)	255.51	56.49	53.88	53.44/64.78	47.89/54.18	47.83/55.91
Fine-Tuned (BS=1)	371.33	93.39	63.64	73.74/76.70	57.26/63.79	62.12/68.35
Fine-Tuned (BS=4)	256.42	57.34	56.41	59.47/69.56	48.92/57.19	52.05/61.13

Table 6.6: Impact of batching on energy and performance. The fine-tuned model maintains high accuracy even at larger batch sizes, enabling massive energy savings.

While the baseline model suffered a degradation in accuracy when batched (likely due to padding effects or attention masking variance), the Fine-Tuned model remained robust. This enabled a dual benefit: aggressive batching could be utilized to lower energy consumption by ≈37% while simultaneously achieving higher accuracy than the unoptimized baseline.

6.6.2 Resolution Scaling

In addition to batching, the impact of input resolution on model efficiency was investigated. While modern VLMs typically operate at high resolutions (e.g., 1024 × 1024 or dynamic aspect ratios) to capture fine details, this incurs a quadratic cost in the visual encoder's attention mechanism. It was hypothesized that for damage assessment, a lower resolution might suffice if the model is properly adapted.

The fine-tuned model was evaluated at a reduced resolution of 400 × 400 pixels. Table 6.7 compares the performance of the standard resolution (dynamic) versus the fixed 400 × 400 input.

Configuration	**Time**	**GPU**	**Acc.**	**Prec.**	**Rec.**	**F1**
	(s)	**(kJ)**	**(%)**	**(M/W)**	**(M/W)**	**(M/W)**
Standard Res (BS=1)	371.33	93.39	63.64	73.74/76.70	57.26/63.79	62.12/68.35
Standard Res (BS=4)	256.42	57.34	56.41	59.47/69.56	48.92/57.19	52.05/61.13
Reduced Res (BS=1)	284.54	77.76	58.05	73.55/72.29	53.98/57.73	56.24/60.68
Reduced Res (BS=4)	155.22	38.52	56.62	62.28/69.34	52.76/56.56	53.42/59.36

Table 6.7: Impact of input resolution and batching on performance and energy. Reducing resolution to 400 × 400 combined with batching yields significant energy savings.

The results indicate a favorable trade-off. Reducing the resolution decreased the total energy consumption by 16.7% (from 93.39 kJ to 77.76 kJ) and reduced inference time by 23.4%. While there was a drop in the weighted F1-score (from 68.35% to 60.68%), the model's performance remained competitive with the full-resolution Baseline model (61.05%).

Significantly, when combined with batching (e.g., Batch Size 4 at 400px), the energy consumption dropped even further to 38.52 kJ, representing a 58.7% reduction compared to the standard single-image inference. This configuration (400×400, BS=4) represents the optimal efficiency configuration for edge deployment, where battery life and throughput are prioritized over maximal precision.

6.7 Per-Category Performance Analysis

While global metrics provide a high-level overview, they often mask the model's performance on specific, challenging sub-categories. To understand the precise impact of fine-tuning, the F1-scores for individual damage classes were analyzed. Table 6.8 and

Figure 6.2 present a breakdown of performance across key damage types for the Baseline (Zero-Shot) and Curated Fine-Tuned models.

Damage Category	**Support**	**Baseline F1**	**Fine-Tuned F1**	**Improvement (Δ)**
Glass/Windscreen	270	90.19%	91.05%	+0.86%
Tire Damage	31	83.02%	83.64%	+0.62%
Dent (Body/Panel)	612	51.28%	**64.41%**	**+13.13%**
Front Bumper	251	56.18%	57.97%	+1.79%
Rear Bumper	144	50.43%	55.59%	+5.16%
Light/Lamp	224	46.69%	58.43%	+11.74%
Scratch (Body/Panel)	197	31.40%	**48.00%**	**+16.60%**

Table 6.8: Per-category F1-score comparison between Baseline Qwen-VL-7B and Curated Fine-Tuned model. The largest gains are observed in subtle, texture-based categories like Scratches and Dents.

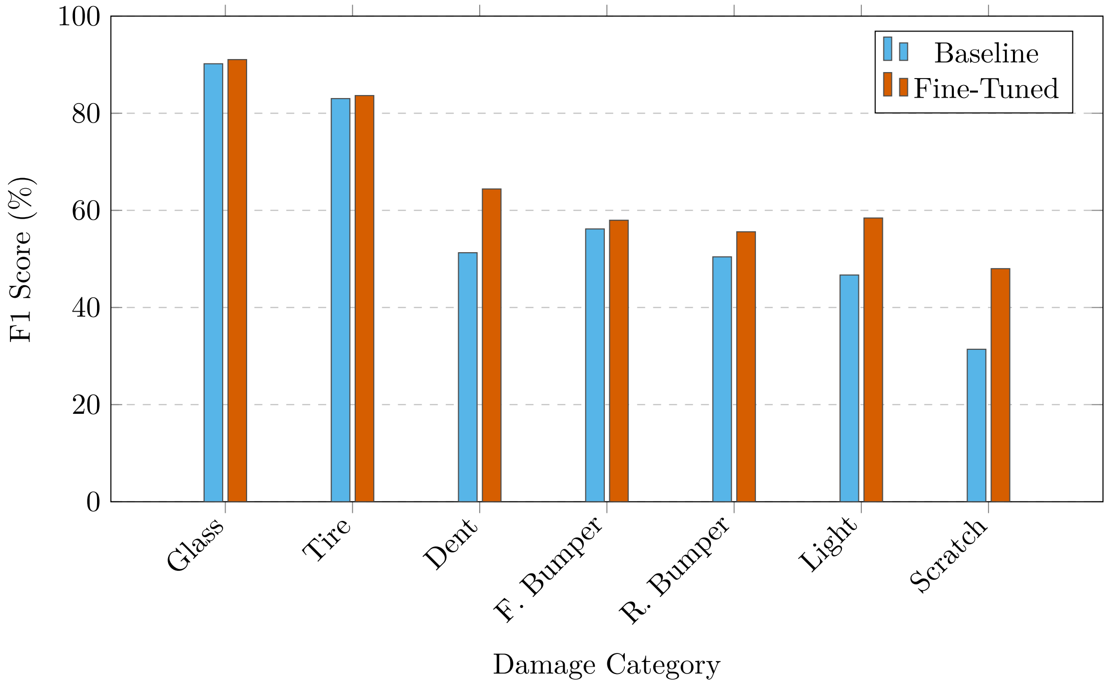


Figure 6.2: Visual comparison of per-category F1 scores. The fine-tuned model shows dramatic improvements in challenging categories like Dents and Scratches.

6.7.1 Visual Saliency and Feature Granularity

The results reveal a clear correlation between the visual prominence of the damage and the model's zero-shot capability.

- **High-Saliency Features (Glass, Tires):** Categories with distinct shapes and high contrast, such as shattered windshields or flat tires, achieved high performance ($> 80\%$) even without fine-tuning. The pre-trained visual encoder already possesses the semantic concepts for "glass" and "wheel," requiring minimal adaptation.
- **Low-Saliency Features (Scratches, Dents):** The most significant gains were observed in "Scratch" (+16.6%) and "Dent" (+13.1%) categories. These damages are often defined by subtle texture changes, reflections, or minor deformations rather than distinct object boundaries. The zero-shot model frequently missed these fine-grained details, likely interpreting them as lighting artifacts. Fine-tuning successfully adapted the visual encoder to attend to these specific high-frequency texture cues.

6.8 The "Green AI" Break-Even Analysis

A critical question in sustainable AI is whether the energy cost of training is justified by the subsequent efficiency gains during inference. A "Break-Even Analysis" is proposed to quantify the operational point at which the fine-tuning investment pays off.

6.8.1 Calculating the Energy Debt

As detailed in Section 6.3, the fine-tuning process (Curated Full) consumed a total of **963.66 kJ**. This represents the "energy debt" that must be amortized over the model's operational lifecycle.

6.8.2 Inference Savings and Payback Period

By enabling the use of a larger batch size (BS=4) without sacrificing accuracy (as shown in Section 6.6), the fine-tuned model reduces the energy cost per sample significantly compared to the robust baseline setting (BS=1).

- **Baseline Energy Cost (BS=1):** 53.10 J per sample.
- **Optimized Fine-Tuned Cost (BS=4):** 33.12 J per sample.
- **Energy Savings:** $\Delta E = 53.10 - 33.12 = 19.98$ J per sample.

The break-even point (N_{BE}), as illustrated in Figure 6.3, is calculated as:

$$N_{BE} = \frac{\text{Training Energy Debt}}{\text{Energy Savings per Sample}} = \frac{963,660 \text{ J}}{19.98 \text{ J/sample}} \approx 48,231 \text{ samples}$$

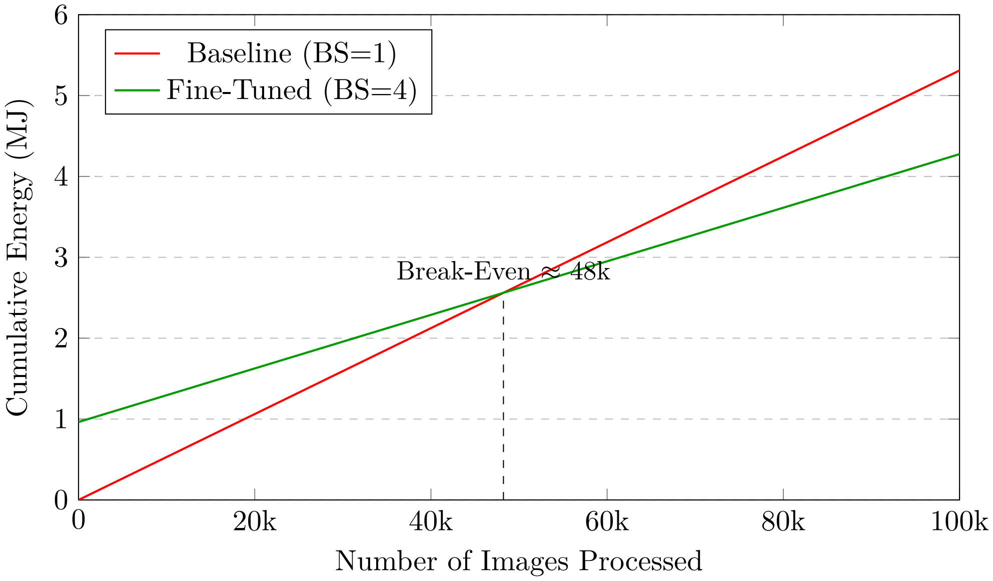


Figure 6.3: Break-Even Analysis of Energy Consumption. The initial energy debt of fine-tuning (y-intercept) is recovered after approximately 48,000 images, after which the fine-tuned model becomes more energy-efficient than the baseline.

6.8.3 Operational Implications

This result has significant implications for industrial deployment. For a large-scale insurance provider processing millions of claims annually, the "energy debt" of fine-tuning is recovered after processing just **48,231 images**. Beyond this point, the fine-tuned model is effectively "carbon negative" relative to the baseline, generating a net energy saving of ≈20 kJ for every 1,000 subsequent images. This demonstrates that targeted training is a prerequisite for sustainable large-scale inference.

6.9 Discussion and Limitations

6.9.1 The Role of Data Quality

The results strongly support the "Data-Centric" paradigm. The Curated dataset, despite being smaller than the Naive dataset, produced a superior model in less training time. This suggests that the limiting factor in VLM adaptation is not the quantity of data, but the consistency of the semantic-visual alignment. The removal of ambiguous labels allowed the model to converge on a more robust internal representation, as evidenced by the 16% improvement in the "Scratch" category.

6.9.2 Limitations

While the results are promising, several limitations remain:

- **2D vs. 3D Understanding:** The model operates on single 2D images. Some damage types, such as subtle deformations in body panels, are inherently 3D phenomena that rely on depth perception and motion parallax (e.g., moving a camera around a reflection). A single viewpoint may remain ambiguous even for a fine-tuned model.

- **Severe Class Imbalance:** Despite stratification, categories like “Roof Damage” and “Tire Damage” remain rare. While performance on “Tire Damage” was high due to its distinct visual signature, “Roof Damage” performance remained volatile. Future work could explore synthetic data generation to augment these rare classes.

6.9.3 Summary

In summary, the experimental results demonstrate that a data-centric approach to fine-tuning, combined with strategic operational optimizations, can significantly enhance both the performance and sustainability of VLMs. The Qwen-VL-7B model, when fine-tuned on a curated dataset and deployed with optimized batching and resolution settings, achieves a compelling balance between high-fidelity damage assessment and energy efficiency. These findings provide a robust foundation for the conclusions and future research directions discussed in the final chapter.

CHAPTER 7

Conclusions and Future Research

This chapter synthesizes the key findings and contributions of this thesis, addresses research questions, and demonstrates the effectiveness of the proposed sustainable VLM framework for the assessment of vehicle damage. It concludes with future research directions aimed at enhancing the framework's capabilities and exploring new applications.

7.1 Conclusion

This thesis addressed the dual challenge of accuracy and sustainability in automated vehicle damage assessment. Using large VLMs, a framework was established that not only maximizes predictive performance, but also ensures energy-efficient deployment in industrial settings. The methodology combined a rigorous benchmarking phase with a data-centric fine-tuning approach, validated through extensive experimentation on a custom car damage dataset.

The experimental results confirmed that the Qwen-VL-7B architecture offered the optimal balance between performance and efficiency. It achieved a zero-shot weighted F1-score of 61.05%, outperforming larger models while consuming significantly less energy. Furthermore, the "Data-Centric" approach demonstrated that a smaller, curated dataset could yield better results than a larger, noisy baseline (F1: 68.35% vs. 66.77%), while also reducing training energy by 19.5%.

Significantly, high accuracy was shown to not require excessive computational waste. Through the implementation of an optimized configuration (Batch Size 4, 400 × 400 resolution), a 58.7% reduction in inference energy consumption was achieved compared to the standard baseline. The "Break-Even Analysis" further revealed that the energy cost of fine-tuning was recovered after processing approximately 48,000 images, making the solution net-positive for the environment in long-term deployment.

Answering the Research Questions

The research questions addressed in this thesis are as follows:

- **RQ1:** How does the performance of a compact multimodal model fine-tuned on a dataset of car damage images and text descriptions, compare to general-purpose VLMs in detection accuracy, processing time, and energy usage?
- **RQ2:** What impact does a structured data curation pipeline have on model generalization and data efficiency versus naïve aggregation?
- **RQ3:** To what extent can energy-efficient model adaptation techniques minimize energy consumption while maintaining or improving model accuracy?

RQ1: Performance of Compact vs. General VLMs. As detailed in the benchmarking analysis (Chapter 6, Section 6.1), Qwen-VL-7B emerged as the superior architecture. It significantly outperformed larger or similarly sized models like Phi-3.5-Vision and LLaVA-1.5, proving that parameter count is not the sole determinant of performance. The high-resolution visual encoder of the Qwen architecture was particularly effective at detecting fine-grained damage features.

RQ2: Impact of Data Curation. The fine-tuning experiments (Chapter 6, Section 6.3) provided strong evidence for the Data-Centric AI hypothesis. The model trained on the rigorously filtered "Curated" dataset achieved higher accuracy and faster convergence than the model trained on the larger "Naive" dataset. This confirmed that removing ambiguous samples helped the model form sharper decision boundaries, improving both generalization and training efficiency.

RQ3: Energy-Efficient Adaptation. The efficiency analysis (Chapter 6, Section 6.6) directly addressed this question. It was found that by optimizing batch sizes and reducing input resolution to 400 × 400, energy consumption could be cut by nearly 60% with only an acceptable trade-off in accuracy. Additionally, the study showed that vision-only adaptation was the most effective fine-tuning strategy for this domain, minimizing computational overhead while maximizing learning.

In summary, this thesis demonstrated that integrating data-centric fine-tuning with operational parameter optimization offered a powerful, sustainable solution for deploying VLM-based damage assessment tools.

7.2 Future Research

Although the proposed framework demonstrates significant potential, several avenues for future research emerge from its current limitations. These directions aim to further enhance the system's capabilities and broaden its applicability.

7.2.1 Multi-View and 3D Analysis

The current reliance on single-view 2D images limits the assessment of depth-dependent damages, such as the severity of a dent. Future work should focus on integrating multi-view consistency checks or 3D reconstruction techniques. This would allow the model to understand the volumetric nature of damage, providing a more comprehensive and accurate assessment.

7.2.2 Synthetic Data Augmentation

Addressing the class imbalance observed in rare categories like “Roof Damage” remains a challenge. Future research could utilize Generative AI to create synthetic photorealistic examples of these rare damages. This would resolve the data scarcity issue without the prohibitive cost of manual data collection, potentially stabilizing performance across all categories.

7.2.3 Model Quantization and Edge Deployment

Although this study focused on server-grade GPUs, the ultimate goal is the deployment on edge devices. Future studies should investigate 4-bit or 8-bit quantization techniques (e.g., QLoRA or GPTQ) to further reduce the memory footprint. This would enable these powerful 7B models to run directly on smartphones or Internet of Things (IoT) devices, maximizing accessibility and energy efficiency.

7.2.4 Hybrid Architectures

The success of the vision-only adaptation suggests that the visual encoder is the critical component. Future research could explore hybrid architectures that decouple detection and description tasks. For example, a lightweight CNN (like YOLOv8) could be used for rapid damage localization, feeding cropped regions of interest into a VLM for detailed severity assessment. This “cascade” approach could potentially offer the speed of object detectors with the reasoning capabilities of VLMs.

Overview of Generative AI Tools Used

I confirm that GPT-4, Writefull (via Overleaf), and DeepL were used solely as supplementary aids for language improvement and translation of key terms in this thesis. These tools were exclusively utilized for checking grammar, spelling, and improving sentence structure based on my original text. No content was generated purely from a prompt and included without substantial input and revision.

List of Figures

List of Tables

Acronyms

AI artificial intelligence. 2–4, 8, 9, 11, 14, 19, 23, 53, 54, 56, 60, 64, 65

CLIP Contrastive Language-Image Pre-training. 35, 36

CNN convolutional neural network. 11, 12, 65

FNOL First Notice of Loss. 11

GAN generative adversarial network. 13

GDP gross domestic product. 1

GPU graphics processing unit. 4, 11, 22, 38, 40, 43, 44, 47–49, 65, 69

IoT Internet of Things. 65

LLaVA Large Language and Vision Assistant. 2, 3, 7, 9, 27, 35, 52, 53, 64, 71

LLM Large Language Model. 9, 10, 14, 22, 49, 55, 70

LoRA Low-Rank Adaptation. 17, 22, 39, 40, 48–50, 70

Mask R-CNN Mask Region-Based Convolutional Neural Network. 12

ML machine learning. 8, 21, 27

MMLU Massive Multitask Language Understanding. 10

NVML NVIDIA Management Library. 4, 17, 22, 23, 27, 38, 40, 41, 47, 48, 69

OCR Optical Character Recognition. 24, 36

PEFT parameter-efficient fine-tuning. 16, 17, 21, 22, 25

PTQ post-training quantization. 15, 16, 24, 69

QAT quatization-aware training. 15

QLoRA Quantized LoRA. 17, 22, 25, 65

ResNet Residual Network. 12, 13

ViT Vision Transformer. 35, 36

VLM vision-language model. 1–4, 7–10, 13, 19, 22, 24, 27, 34, 35, 39, 44, 51–55, 62–65, 70, 71

VQA Visual Question Answering. 8, 35, 36

VRAM Video Random Access Memory. 22, 39, 44, 50

WHO World Health Organization. 1

YOLO You Only Look Once. 12, 13, 23, 65

Bibliography

[ABW+23] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023.

[ag21] ayush grover. car-dects-new dataset. `https://universe.roboflow.com/ayush-grover/car-dects-new`, dec 2021. visited on 2025-06-25.

[AJA+24] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp A. Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024.

[AMNA24] Beecha Aditya, Mohammad Moizuddin, Mekala Naveen, and A. Amulya. Insurance amount prediction based on accidental car damage level using cnn. *International Journal of Software Engineering and Technology*, 2024.

[ASJP23] Azin Asgarian, Rohit Saha, Daniel Jakubovitz, and Julia Peyre. Autofraudnet: A multimodal network to detect fraud in the auto insurance industry. *CoRR*, abs/2301.07526, 2023.

[BBC+23] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *CoRR*, abs/2309.16609, 2023.

[BBY+24] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond, 2024.

[BCL+25] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025.

[BCTC19] Hédi Ben-Younes, Rémi Cadène, Nicolas Thome, and Matthieu Cord. BLOCK: bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI*

2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, pages 8102–8109. AAAI Press, 2019.

[BMR$^+$20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[CAP23] CAPSTONE. Car damage detection dataset. `https://universe.roboflow.com/capstone-nh0nc/car-damage-detection-t0g92`, aug 2023. visited on 2025-06-25.

[Cha22] Rahul Deb Chakladar. First notice of loss made simple: Redefining claims management for digital customers. *European Journal of Engineering and Technology Research*, pages 152–154, 12 2022.

[CJ25] Arpit Chauhan and Amol Joglekar. Automated vehicle damage analysis and cost prediction with mask r-cnn for insurance and repair optimization. *International Journal of Scientific Research and Engineering Development*, 8(4), July 2025.

[CWZZ17] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, abs/1710.09282, 2017.

[Dam23] Car Damage. Car damage images dataset. `https://universe.roboflow.com/car-damage-kadad/car-damage-images`, mar 2023. visited on 2025-06-25.

[DPHZ23] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[EHP$^+$25] Cooper Elsworth, Keguo Huang, David Patterson, Ian Schneider, Robert Sedivy, Savannah Goodman, Ben Townsend, Parthasarathy Ranganathan, Jeff Dean, Amin Vahdat, Ben Gomes, and James Manyika. Measuring the environmental impact of delivering ai at google scale, 2025.

[EYEW22] Fevziye Irem Eyiokur, Dogucan Yaman, Hazim Kemal Ekenel, and Alexander Waibel. Exposure correction model to enhance image quality. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 675–685. IEEE, 2022.

[FJZL17] Mengjuan Fei, Zhaojie Ju, Xiantong Zhen, and Jing Li. Real-time visual tracking based on improved perceptual hashing. *Multim. Tools Appl.*, 76(3):4617–4634, 2017.

[GACZ25] Mohsen Gholami, Mohammad Akbari, Kevin Cannons, and Yong Zhang. CASP: compression of large multimodal models based on attention sparsity. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 9372–9381. Computer Vision Foundation / IEEE, 2025.

[GLX$^+$23] Youdi Gong, Guangzhen Liu, Yunzhi Xue, Rui Li, and Lingzhong Meng. A survey on dataset quality in machine learning. *Inf. Softw. Technol.*, 162:107268, 2023.

[GPM$^+$20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, 2020.

[GWW19] Wenzhong Guo, Jianwen Wang, and Shiping Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.

[GZZ$^+$23] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models, 2023.

[Ham50] Richard W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[HHt23] Daniel Han, Michael Han, and Unsloth team. Unsloth. `http://github.com/unslothai/unsloth`, 2023.

[HLW$^+$24] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *CoRR*, abs/2402.11530, 2024.

[HNO$^+$24] Md Jahid Hasan, Agustinus Nalwan, Kok-Leong Ong, Hamed Jahani, Yee Ling Boo, Kha Cong Nguyen, and Mahmudul Hasan. Groundingcardd: Text-guided multimodal phrase grounding for car damage detection. *IEEE Access*, 12:179464–179477, 2024.

[HSW+22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[Hua25] Ana Huamán. Opencv: Laplace operator, 2025. Accessed: 2025-06-25.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[Kag25] Kaggle Inc. Kaggle datasets, 2025. Accessed: 2025-06-25.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.

[LHLW24] Zhangheng Li, Junyuan Hong, Bo Li, and Zhangyang Wang. Shake to leak: Fine-tuning diffusion models can amplify the generative privacy risk. *CoRR*, abs/2403.09450, 2024.

[LHN+25] Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, Shengen Yan, Huazhong Yang, and Yu Wang. MBQ: modality-balanced quantization for large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 4167–4177. Computer Vision Foundation / IEEE, 2025.

[LLLL24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26286–26296. IEEE, 2024.

[LLW+25] Mengqi Lei, Siqi Li, Yihong Wu, Han Hu, You Zhou, Xinhu Zheng, Guiguang Ding, Shaoyi Du, Zongze Wu, and Yue Gao. Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception. *CoRR*, abs/2506.17733, 2025.

[LLWL23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[LT24] Songtao Li and Hao Tang. Multimodal alignment and fusion: A survey. *CoRR*, abs/2411.17040, 2024.

[LVL23] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15, 2023.

[LZG+25] Wenzhuo Liu, Fei Zhu, Haiyang Guo, Longhui Wei, and Cheng-Lin Liu. Llava-c: Continual improved visual instruction tuning. *CoRR*, abs/2506.08666, 2025.

[Man21] Hamza Manssor. Car damage assessment dataset. `https://www.kaggle.com/datasets/hamzamanssor/car-damage-assessment`, 2021. Accessed: 2025-06-25.

[MZF+25] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models. *CoRR*, abs/2504.05299, 2025.

[Nay25] Bhabani Nayak. The evolution and architecture of multimodal ai systems. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 11(1):1007–1017, January 2025.

[NVI25] NVIDIA Corporation. *NVML API Reference Guide (vR580)*. NVIDIA Corporation, Aug 2025.

[Ope23] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[Org23] World Health Organization. Road traffic injuries, 2023.

[PGL+21] David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021.

[PNA+25] Nitesh Patnaik, Navdeep Nayak, Himani Bansal Agrawal, Moinak Chinmoy Khamaru, Gourav Bal, Saishree Smaranika Panda, Rishi Raj, Vishal Meena, and Kartheek Vadlamani. Small vision-language models: A survey on compact architectures and techniques. *CoRR*, abs/2503.10665, 2025.

[Pro17] Paul Prondeau. The car connection picture dataset. `https://www.kaggle.com/datasets/prondeau/the-car-connection-picture-dataset`, 2017. Accessed: 2025-06-25.

[PSUL25] Eileen Paula, Jayesh Soni, Himanshu Upadhyay, and Leonel Lagos. Comparative analysis of model compression techniques for achieving carbon efficient ai. *Scientific Reports*, 15:23461, Jul 2025.

[PZCGPM+24] Sergio A. Pérez-Zarate, Daniel Corzo-García, Jose L. Pro-Martín, Juan A. Álvarez García, Miguel A. Martínez del Amor, and David Fernández-Cabrera. Automated car damage assessment using computer vision: Insurance company use case. *Appl. Sci.*, 14(20):9560, 2024.

[R22] Lakshmi Narayanan R. car_defect_2 dataset. `https://universe.roboflow.com/lakshmi-narayanan-r/car_defect_2`, jan 2022. visited on 2025-06-25.

[RDGF16] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788. IEEE Computer Society, 2016.

[Rob25] Roboflow, Inc. Roboflow: Dataset management and hosting for computer vision, 2025. Accessed: 2025-06-25.

[rod22] rodney.virtualassistant@gmail.com. Car damages dataset. `https://universe.roboflow.com/rodney-virtualassistant-gmail-com/car-damages-z0aas`, jun 2022. visited on 2025-06-25.

[Saw24] Yash Sawant. car-dent-detection2 dataset. `https://universe.roboflow.com/yash-sawant/car-dent-detection2`, jul 2024. visited on 2025-06-25.

[SFA+22] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100, 2022.

[SGM19] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics, 2019.

[Sol24] LexisNexis Risk Solutions. U.s. auto insurance trends report highlights new generational risks in drivers and vehicles that continue to contribute to higher claim frequencies, 2024.

[TGLM20] Neil C. Thompson, Kristjan H. Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning. *CoRR*, abs/2007.05558, 2020.

[THKG20] Fadi A. Thabtah, Suhel Hammoud, Firuz Kamalov, and Amanda H. Gonsalves. Data imbalance in classification: Experimental evaluation. *Inf. Sci.*, 513:429–441, 2020.

[TWG$^+$24] Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression. *CoRR*, abs/2402.05964, 2024.

[VAA$^+$25] B. Vanathi, R. Akshaya, P. Alfina, V. Gayathri, and S. Lekhasri. Damage detection using yolov8 ai for vehicle assessment. *International Journal of Innovative Science and Research Technology (IJISRT)*, 25(3):982–987, 2025.

[VSBS25] Priyansh Verma, Kaustubh Shankar, Rahul Biswas, and B. K. Singh. Automatic damaged vehicle estimator using deep learning algorithms. *International Journal of Research Publication and Reviews*, 2025.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[WBT$^+$24] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *CoRR*, abs/2409.12191, 2024.

[WCJ+25] Luping Wang, Sheng Chen, Linnan Jiang, Shu Pan, Runze Cai, Sen Yang, and Fei Yang. Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artif. Intell. Rev.*, 58(8):227, 2025.

[WGC+23] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. Multimodal large language models: A survey. In Jingrui He, Themis Palpanas, Xiaohua Hu, Alfredo Cuzzocrea, Dejing Dou, Dominik Slezak, Wei Wang, Aleksandra Gruca, Jerry Chun-Wei Lin, and Rakesh Agrawal, editors, *IEEE International Conference on Big Data, BigData 2023, Sorrento, Italy, December 15-18, 2023*, pages 2247–2256. IEEE, 2023.

[WLW23] Xinkuang Wang, Wenjing Li, and Zhongcheng Wu. Cardd: A new dataset for vision-based car damage detection. *IEEE Trans. Intell. Transp. Syst.*, 24(7):7202–7214, 2023.

[WMMX17] Haohan Wang, Aaksha Meghawat, Louis-Philippe Morency, and Eric P. Xing. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *2017 IEEE International Conference on Multimedia and Expo, ICME 2017, Hong Kong, China, July 10-14, 2017*, pages 949–954. IEEE Computer Society, 2017.

[XSW+25] Jinda Xu, Yuhao Song, Daming Wang, Weiwei Zhao, Minghua Chen, Kangliang Chen, and Qinya Li. Quality over quantity: Boosting data efficiency through ensembled multimodal data curation. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 21761–21769. AAAI Press, 2025.

[YCC23] Jie You, Jae-Won Chung, and Mosharaf Chowdhury. Zeus: Understanding and optimizing GPU energy consumption of DNN training. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 119–139, Boston, MA, April 2023. USENIX Association.

[YCD+23] Jiaxi Yang, Kui Chen, Kai Ding, Chongning Na, and Meng Wang. Auto insurance fraud detection with multimodal learning. *Data Intell.*, 5(2):388–412, 2023.

[Zew25] Adam Zewe. Explained: Generative ai's environmental impact, Jan 2025.

[Zhu25] Helen Zhuravel. Ai car damage detection: How it works and why it matters, May 2025.

[ZLL+24] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Trans. Assoc. Comput. Linguistics*, 12:1556–1577, 2024.

[ZYHD20] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE J. Sel. Top. Signal Process.*, 14(3):478–493, 2020.