Intelligent Multimodal Framework for Traffic Accident Analysis Using a MODM-MCDM-Based Approach

Luis Bravo *, Ciro Rodriguez *, Pedro Hidalgo, Cesar Angulo, Ciro Aguilar, Joel Vilca, Zoraida Huaman, Victor Tarazona

Posted Date: 11 June 2026

doi: 10.20944/preprints202606.0966.v1

Keywords: road traffic accident; multimodal framework; multi-objective optimisation; multi-criteria decision-making

 doi:10.20944/preprints202606.0966.v1

Article

Intelligent Multimodal Framework for Traffic Accident Analysis Using a MODM-MCDM-Based Approach

Luis Bravo *, Ciro Rodriguez *, Pedro Hidalgo, Cesar Angulo, Ciro Aguilar, Joel Vilca, Zoraida Huaman and Victor Tarazona

Faculty of Systems Engineering and Informatics of National University of San Marcos, Lima 15081, Peru

* Correspondence: luis.bravo4@unmsm.edu.pe (L.B.); crodriguezro@unmsm.edu.pe (C.R.)

Abstract

Road traffic accidents constitute a critical global public health problem, with over 1.19 million deaths annually and an economic cost equivalent to 3% of GDP in most countries, according to the World Health Organisation's 2023 report. Traditional methods of forensic analysis have significant limitations in terms of latency, coverage and scalability. This study proposes an Intelligent Multimodal Framework comprising four sequential phases: object detection, scene classification, visual understanding using vision-language models, and the generation of forensic reports using large-scale language models. All phases are evaluated on a specialised multimodal dataset constructed ad hoc from heterogeneous sources. Each phase was trained independently through fine-tuning on a 2× NVIDIA RTX A4500 platform. To select the optimal configuration from the 320 combinations in the factorial search space of candidate models per phase, the five-layer MODM-MCDM Hybrid Protocol was implemented, utilising seven multi-criteria decision-making methods and AHP weighting across 49 normalised criteria. The results identified two deployment configurations: S1 as the configuration offering maximum performance; and S2 as the configuration offering maximum methodological robustness.

Keywords: road traffic accident; multimodal framework; multi-objective optimisation; multi-criteria decision-making

1. Introduction

Los Traffic accidents represent one of the most critical challenges in road safety and public health at a global scale. According to the World Health Organization (2023), more than 1.19 million people die every year as a result of traffic accidents, making it the eighth leading cause of death worldwide and the leading cause among people aged 5 to 29 years. The associated economic cost amounts to approximately 3% of gross domestic product in most countries, with a disproportionate burden on low- and middle-income nations, where more than 90% of fatal victims are concentrated [1]. In the Latin American context, road accident management and analysis systems exhibit significant structural limitations: dependence on late manual reports, absence of automated forensic analysis pipelines, and scarce integration between multimodal data sources from urban video surveillance systems and vehicle telematics [2].

The rapid advancement of artificial intelligence applied to computer vision and natural language processing has opened transformative possibilities for the automated analysis of road incidents. Real-time object detection models, such as the YOLO family, have reached precision levels competitive with human inspection on standard benchmarks [3]. Vision-language models (VLMs), such as LLaVA-NeXT [4] and Qwen2-VL [5], have demonstrated emergent capabilities of contextual visual understanding that go beyond object classification, generating rich semantic descriptions of complex scenes. In parallel, large language models (LLMs), such as Mistral-7B [6] fine-tuned through low-rank

 doi:10.20944/preprints202606.0966.v1

adapters (LoRA; [7]), have shown the ability to generate structured analytical reports in specialized domains with moderate computational resources.

However, the integration of these capabilities into a cohesive, reproducible, and systematically evaluated forensic pipeline constitutes an unresolved gap in the literature. Existing works address the stages of detection, classification, or narrative description of accidents in a fragmented manner [8,9], but do not integrate all four phases into an end-to-end framework, nor do they provide a systematic and reproducible mechanism for selecting the optimal configuration from the vast combinatorial space of candidate models. This methodological absence produces results that depend on the subjective judgment of the researcher for model selection, compromising reproducibility and inter-study comparability. The multi-criteria decision-making (MCDM) paradigm and multi-objective decision-making (MODM) have been successfully applied to selection problems in systems engineering [10], but their adoption in the field of multimodal artificial intelligence pipelines is practically nonexistent in the specialized literature.

To address these gaps, the present study proposes an Intelligent Multimodal Framework for automated traffic accident analysis that integrates four specialized phases: (F1) vehicular object detection using YOLOv11s, (F2) accident scene classification using Swin Transformer, ViT, InceptionV3, and ResNet50, (F3) visual comprehension and description generation using LLaVA-NeXT and Qwen2-VL with LoRA fine-tuning, and (F4) structured forensic report generation using Mistral-7B and Qwen2-7B with high-capacity LoRA fine-tuning. Over the space of 320 possible pipeline configurations (4×5×4×4), a five-layer Hybrid MODM-MCDM Protocol is applied, combining AHP normalization, seven multi-criteria scalarization methods, and sequential robustness criteria to identify both the maximum-performance configuration (S1) and the maximum methodological robustness configuration (S2).

The specific objectives guiding the study are as follows:

- To construct a specialized multimodal dataset for traffic accident analysis, integrating heterogeneous sources and applying a phase-differentiated semi-automatic labeling scheme with human validation.
- To independently develop and train the models for each phase of the pipeline (F1–F4) through fine-tuning with LoRA adapters on available GPU infrastructure.
- To implement and execute the five-layer Hybrid MODM-MCDM Protocol over the 320 pipeline configurations to identify the maximum-performance configuration (S1).
- To identify the maximum methodological robustness configuration (S2) through sequential multidimensional filtering criteria (C1–C5) with demonstrated invariance to changes in the selection method.

The remainder of the article is organized as follows: Section 2 reviews related work in the domains of road accident analysis with artificial intelligence, multimodal models, and multi-criteria selection methods. Section 3 describes the methodology, including the framework design, dataset construction, labeling process, phase-by-phase training, and evaluation protocol. Section 4 details the experimental setup. Section 5 presents the analysis of results. Section 6 discusses the findings in the context of the literature and their implications for smart city systems. Section 7 concludes the study and outlines future research directions.

2. Related Work

2.1. Automated Traffic Accident Analysis with Computer Vision

The automatic detection of traffic accidents using computer vision has evolved from approaches based on handcrafted features toward end-to-end deep learning architectures. The pioneering works of Yao et al. [11] and Basso et al. [12] demonstrated the feasibility of detecting road incidents in real time using convolutional neural networks (CNNs) trained on urban video surveillance data. The adoption of the YOLO architecture [13] for real-time detection marked a turning point, with successive versions — YOLOv5, YOLOv8, and the recent YOLOv11 [3] — that have consistently

 doi:10.20944/preprints202606.0966.v1

reduced inference latency below 30 ms/image while maintaining mAP@50 precision values above 70% on traffic benchmarks.

Nevertheless, object detection alone provides incomplete information for forensic accident analysis. Characterizing the type of accident, its severity, environmental conditions, and incident dynamics requires a semantic understanding of the scene that exceeds the capabilities of conventional object detectors. In this regard, the recent work of Jaradat et al. [8] on multitask accident analysis using LLMs applied to social media data, and that of Boesch [14] on YOLO-based detection systems, illustrate the trend toward more comprehensive pipelines. However, none of these works integrate the complete chain of detection → classification → visual comprehension → narrative generation under a systematic multi-criteria evaluation protocol.

2.2. Vision-Language Models for Scene Understanding

Vision-language models (VLMs) have emerged as one of the most significant contributions of artificial intelligence in the past three years. LLaVA [4], a pioneer in the instruction-image architecture, demonstrated that connecting a visual encoder (CLIP) with an LLM through fine-tuning on multimodal instructions produced remarkable emergent visual comprehension capabilities. Its successor, LLaVA-NeXT [15], extended the visual context through a high-resolution encoder and improvements in spatial reasoning. In parallel, the Qwen2-VL family [5] introduced a dynamic-resolution visual encoder that processes images of any aspect ratio without degradation, achieving state-of-the-art results on visual comprehension benchmarks such as MMBench, MMStar, and DocVQA.

The application of VLMs to the traffic accident domain is an emerging area with scarce bibliography. Existing works have focused on the general description of traffic scenes [16] or on the generation of synthetic training data [17], but not on the contextualized understanding of accident scenes for forensic purposes. The present work contributes to bridging this gap by fine-tuning LLaVA-NeXT and Qwen2-VL on a curated vehicular accident dataset with ShareGPT-format instructions, systematically comparing the performance of both architectures on the same domain.

2.3. Large Language Models for Report Generation

The automatic generation of forensic reports using LLMs represents the synthesis layer of the proposed pipeline. Mistral-7B [6], with its sliding window attention architecture and grouped query attention, delivered competitive results against significantly larger models on reasoning and instruction-following benchmarks. Its fine-tuning via LoRA in high-capacity configurations (r=32, alpha=64) has proven effective for specialization in technical domains with moderate computational resources [7]. Qwen2-7B [18], in turn, has shown advantages in inference efficiency and lower latency than Mistral-7B in real-time deployment configurations.

In the context of road accident analysis, the work of Jaradat et al. [8] applied fine-tuned GPT-2 to the multitask classification of accident-related tweets, achieving a mean accuracy of 85% and a BLEU-4 of 0.22 on information retrieval tasks. However, the use of more recent LLM architectures (Mistral-7B, Qwen2-7B) fine-tuned on structured accident data rather than social media text and their integration as the final stage of a visual pipeline remains unexplored in the literature.

2.4. Multi-Criteria Decision Methods in Artificial Intelligence Systems

Selecting the optimal configuration in multimodal AI systems is a multi-objective optimization problem when the performance criteria are multiple and potentially conflicting — precision, latency, stability, and memory consumption. Multi-criteria decision-making (MCDM) methods such as TOPSIS [19], ELECTRE III [20], and PROMETHEE II [21] have been widely applied in systems engineering, urban planning, and supplier selection. The Analytic Hierarchy Process (AHP) [22] has proven to be an effective tool for weighting criteria in problems with multiple groups of metrics.

 doi:10.20944/preprints202606.0966.v1

Nevertheless, the application of MCDM methods to configuration selection in deep learning pipelines is practically nonexistent in the reviewed literature. Paradowski et al. [10] recommend comparing multiple selection criteria to verify the invariance of the solution under changes of method, arguing that the convergence of multiple MCDM methods on the same solution is evidence of its genuine optimality. The present study is the first to apply a MODM-MCDM protocol of this nature for configuration selection in a multimodal AI pipeline for forensic traffic accident analysis.

3. Methodology

3.1. Architecture

The architecture presents a sequential pipeline beginning with image detection, where the input data is processed by candidate models to infer objects and other elements as output. In the second phase, the input remains the same image, which is processed by classification candidate models to predict the resulting class. In Phase 3, the inputs are the predictions from Phases 1 and 2, whose output serves as input for the final phase, as illustrated in Figure 1.

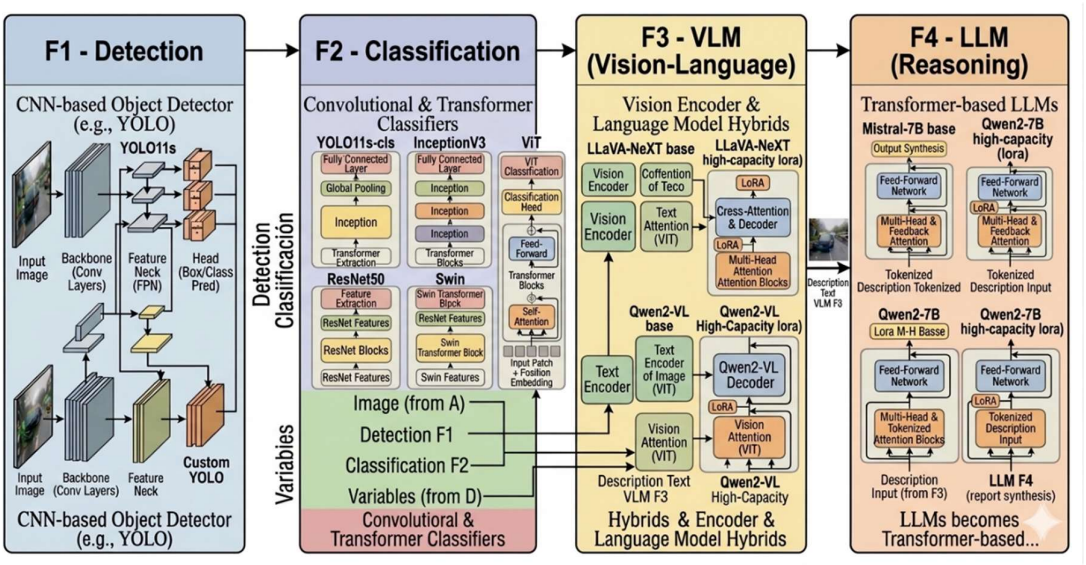


Figure 1. Model selection architecture in the Multimodal Framework.

3.2. Data Collection

The multimodal dataset was constructed using data from various sources. The criteria applied during data collection were based on the characteristics associated with the different phases: detection (F1), classification (F2), vision-language (F3), and large language (F4), as shown in Table 1.

Table 1. Multimodal data sources.

Source	Type	F1	F2	F3	F4
KITTI Object Detection https://www.cvlibs.net/datasets/kitti/	7k Images	x			
Kaggle- Roboflow https://www.kaggle.com/ https://universe.roboflow.com	70k Images	x	x		
ShareGPT4V-COCO-2017 https://huggingface.co/datasets/Lin-Chen/ShareGPT4V	100K Images /descriptions				x

 doi:10.20944/preprints202606.0966.v1

https://huggingface.co/datasets/phiyodr/coco2017		
Dataset: tatsu-lab/alpaca https://huggingface.co/datasets/tatsu-lab/alpaca	(52K Instructions)	x

The dataset structure follows standard formats, as shown in Table 2, which details the format characteristics for each phase. The Multimodal Dataset is publicly available at:

https://github.com/BRAVOUNIX/Framework-Multimodal-Accidentes-Trafico-Dataset-Entrenamiento-Protocolo-MCDM-MODM

Table 2. Multimodal Dataset structure.

Multimodal Dataset	Structure Format
Detection	Yolo txt o Darknet format class_id x_center y_center width height
Classification	ImageFolder format o ImageNet-style format

Vision-Language Model (VLM)	“id”:
	“image”:
	“conversations”: [
	{
	“from”: “human”,
	“value”: “”
	},
	{
	“from”: “gpt”,
	“value”: “”
	}
	]

Large Language Model (LLM)	{
	“instruction”: “”,
	“input”: “”,
	“output”: “”
	}

3.3. Data Preprocessing y Labeling

The multimodal dataset labeling process followed a semi-automatic scheme with phase-differentiated human validation. In F1, detection labels in YOLO format were generated through automatic inference using YOLOv11-XLarge and subsequently verified by human reviewers, constituting a model-assisted labeling process. In F2, scene-category classification labels were directly verified by human reviewers, constituting a manually verified labeling process. In F3 and F4, training instructions in ShareGPT and Alpaca formats were generated using a large language model (Claude Sonnet 4.6) and subsequently reviewed by domain experts, constituting an LLM-assisted synthetic labeling process with human validation. This combined scheme is consistent with the best practices documented in the construction of reference multimodal datasets such as LLaVA-Instruct-150K [4] and Stanford Alpaca [23].

3.4. Model Training and Fine-Tuning

The framework training followed an independent phase-by-phase fine-tuning approach, where each phase has its own dataset, candidate models, hyperparameters, loss function, and evaluation

 doi:10.20944/preprints202606.0966.v1

metrics. The base hardware consisted of a server with 2× NVIDIA RTX A4500 (20,470 MB VRAM per GPU).

3.5. Phase Execution and Results Generation

In the integrated phase execution process (F1→F4), the candidate models used prediction methods as evidenced in Table 3. The data inferred by each preceding phase served as input for the subsequent phase.

Table 3. Phase execution methods in the Multimodal Framework.

Phase	Method	Description	Calculated Metrics
1	validate_single(image_path, gt_loader)	Ejecuta YOLO11.predict(image) → extrae bboxes, class_counts, confianza [24]; compara con ground truth via IoU	precision, recall, F1, mAP@50, mAP@50-95, IoU [25], latencia_ms, VRAM_MB
2	validate_single(image_path, gt_loader)	Ejecuta inferencia según arquitectura (YOLO cls o PyTorch forward) → predicted_class, confidence; compara con GT	accuracy, F1 macro/micro/weighted, precision, recall, latencia_ms, VRAM_MB
3	validate_single(image_path, det_result, cls_result, gt_loader)	Construye prompt visual con contexto F1+F2 → model.generate(max_new_tokens=256) → descripción textual; evalúa contra ground truth	ROUGE-1/2/L, METEOR, BERTScore, BLEU-1/4, CIDEr, CLIPScore, perplexidad
4	validate_single(det_result, cls_result, vlm_result, gt_loader, image_path)	Construye prompt con contexto acumulado F1+F2+F3 → model.generate(max_new_tokens=150) → análisis forense; evalúa contra GT	ROUGE-L, METEOR, BERTScore, BLEU-4, MMLU, MAUVE, perplexidad

The data generated by the integrated phase execution process (F1→F4) were recorded in a CSV file containing all phase-specific metrics, for subsequent analysis.

3.6. Hybrid MODM-MCDM Protocol

Following the result generation process from the integrated phase execution (F1→F4), a hybrid protocol combining the best of MODM and MCDM methods was applied according to the specialized literature [26,27], as shown in Table 4. The integration of multi-objective decision-making (MODM) with multi-criteria decision-making (MCDM) enables exploration of the Pareto-optimal solution space [28] and selection of the optimal configuration through consensus criteria [29].

Table 4. Hybrid MODM-MCDM Protocol.

Layer	Hybrid MODM-MCDM Protocol
Layer 1	Search space: 320 combinations F1×F2×F3×F4 with 49 total metrics
Layer 2	Metric normalization and weight assignment via AHP
Layer 3	Scalarization with compensatory and non-compensatory methods: WSum, TOPSIS, ELECTRE, PROMETHEE, Tcheby, and ASF
Layer 4	Analysis of Robust and Performance Methods
Layer 5	Selection of Robust and Performance Configuration

 doi:10.20944/preprints202606.0966.v1

3.6.1. Layer 1 Multi-Phase Pipeline Evaluation F1 → F4

This layer executes the integrated phase process, characterized by all combinations of candidate models per phase. The number of configurations is determined by the factorial product of candidate models per phase according to Equation (1):

$$4(F1) \times 5\ (F2) \times 4\ (F3) \times 4\ (F4) = 320 \text{ combinations} \quad (1)$$

3.6.2. Layer 2 AHP Normalization and Objective Weighting

This layer processes the 49 metrics generated during integrated phase execution, transforming them to the [0,1] range and computing criterion weights using the Analytic Hierarchy Process (AHP) [30].

Each metric is normalized to the [0,1] range respecting its optimization direction: performance metrics such as mAP, accuracy, and BERTScore are maximized, while computational resource metrics such as latency, VRAM, and dispersion are minimized by inverting the normalization [31]. The normalization formulas use Equations (2) and (3):

For the "maximize" direction:

$$\text{normalization} = (v - \min) / (\max - \min) \rightarrow \text{el mejor valor} = 1.0 \quad (2)$$

For the "minimize" direction:

$$\text{normalization} = (\max - v) / (\max - \min) \rightarrow \text{el mejor valor} = 1.0 \quad (3)$$

Where:

- v is the metric value for the combination
- min is the minimum value of the metric
- max is the maximum value of the metric

The metrics selected for each phase are assigned AHP weights as shown in Table 5.

Table 5. Metric list with AHP weights.

Group	Metric	AHP Weight*	Direction	Justification
F1 Det (Σ=0.360)	det_precision_mean	0.048	↑ max	Accuracy of detected bounding boxes
F1 Det	det_recall_mean	0.048	↑ max	Coverage of real objects in the scene
F1 Det	det_f1_mean	0.048	↑ max	Precision-recall balance in detection
F1 Det	det_map50_mean	0.048	↑ max	Standard COCO metric IoU=0.50
F1 Det	det_map5095_mean	0.024	↑ max	Multi-threshold COCO rigor
F1 Det	det_iou_mean	0.024	↑ max	Predicted bbox vs GT overlap
F1 Det	det_latency_mean_ms	0.060	↓ min	Real-time processing bottleneck
F1 Det	det_throughput_mean	0.040	↑ max	Images processed per second
F1 Det	det_gpu_util_mean	0.036	↓ min	GPU efficiency during detection
F1 Det	det_cpu_util_mean	0.024	↓ min	CPU load during F1 inference
F2 Cls (Σ=0.215)	cls_accuracy	0.0375	↑ max	Overall accuracy for 4 classes

 doi:10.20944/preprints202606.0966.v1

F2 Cls	cls_precision_macro	0.0375	↑ max	Macro-averaged precision
F2 Cls	cls_recall_macro	0.0375	↑ max	Macro-averaged recall
F2 Cls	cls_f1_macro	0.0375	↑ max	Class-balanced F1
F2 Cls	cls_latency_mean_ms	0.0375	↓ min	Classification inference latency
F2 Cls	cls_throughput_mean	0.025	↑ max	Classification throughput
F2 Cls	cls_gpu_util_mean	0.0225	↓ min	GPU utilization in classification
F2 Cls	cls_cpu_util_mean	0.015	↓ min	CPU utilization in classification
F3 VLM (Σ=0.200)	vlm_bleu1..4_mean	4×0.007	↑ max	N-gram overlap with GT description
F3 VLM	vlm_meteor_mean	0.014	↑ max	Generation quality with synonyms
F3 VLM	vlm_rouge1/2/L_f	3×0.014	↑ max	Overlap with reference text
F3 VLM	vlm_bertscore_mean	0.014	↑ max	Semantic similarity with BERT embeddings
F3 VLM	vlm_cider_mean	0.014	↑ max	Consensus-based caption evaluation
F3 VLM	vlm_latency_mean_ms	0.024	↓ min	Visual generation latency (bottleneck)
F4 LLM (Σ=0.165)	llm_bleu1..4_mean	4×0.00525	↑ max	Forensic text generation quality
F4 LLM	llm_meteor_mean	0.0105	↑ max	Quality with morphological alignment
llm rougeL/1/2	llm_rouge*_f	3×0.0105	↑ max	Textual ROUGE for forensic analysis
F4 LLM	llm_bertscore_mean	0.0105	↑ max	Semantic similarity in analysis
F4 LLM	llm_mmlu_mean	0.021	↑ max	Deep reasoning and comprehension (double weight)
F4 LLM	llm_mauve_mean	0.0105	↑ max	Distributional similarity with human text
F4 LLM	llm_latency_mean_ms	0.018	↓ min	LLM latency (largest bottleneck)
Global (Σ=0.060)	total_latency_mean_ms	variable	↓ min	Full pipeline latency F1+F2+F3+F4
Global	total_vram_max_mb	variable	↓ min	Peak VRAM of the complete system

* Σ Peso AHP = 1.000.

3.6.3. Layer 3 Scalarization and Consensus

This layer applies seven MCDM/MODM methods to the normalized 320×49 metric matrix. The methods are:

(WSum, WProd, Tcheby, ASF) [26,32] and (TOPSIS, ELECTRE, PROMETHEE) [33–35]

The methods applied to the execution metrics are described in Table 6.

 doi:10.20944/preprints202606.0966.v1

Table 6. Methods applied to execution metrics.

Method	Formula / Logic	Optimal Direction	Result
WSum (Weighted sum)	$WSUM = \Sigma(w_i \times norm_i)$ para i=1..49	↑ higher = better	Continuous score [0,1]
WProd (Geometric)	$WPROD = \Pi(norm_i \hat{} w_i)$	↑ higher = better	Penalizes zero values
Tcheby (Chebyshev)	$TCHE = \max(w_i \times \lvert 1 - norm_i \rvert)$	↓ lower = better	Minimizes worst objective
ASF (Achievement)	$ASF = \max(norm_i/w_i) + \varrho \times \Sigma(norm_i/w_i) \cdot \varrho$=0.001	↓ lower = better	AHP reference point
TOPSIS	d+ = dist. a ideal; d− = dist. a anti-ideal; C = d−/(d++d−)	↑ higher = better	Ranking by ideal proximity
ELECTRE III	Consensus C(a,b) vs. discordance D(a,b); net flow ELECTRE_net	↑ net = better	Pairwise outranking
PROMETHEE II	Φ+ = Σpref(a,b); Φ− = Σpref(b,a); Φ_net = Φ+ − Φ−	↑ higher = better	Net preference flow

A consensus of the rankings produced by the applied methods is proposed. This consensus is referred to as the Consensus Rank Index (CRI), defined in Equation (4):

$$CRI(a)=\frac{1}{M}\sum_{m=1}^{M} r_m(a) \quad (4)$$

Where:

- CRI(a) is the Consensus Rank Index of alternative a
- M is the total number of MCDM methods ($M = 7$)
- m is the MCDM method index ($m = 1,2,\ldots,7$)
- r_m is the normalized rank of alternative a under method m
- a is the evaluated alternative ($a \in \{1,2,\ldots,320\}$)

The proposed CRI will be used in subsequent layers of the hybrid MODM-MCDM protocol.

3.6.4. Layer 4 Analysis of Performance and Robustness Criteria

En This section analyses the optimal combinations of performance and robustness of various proposed criteria applied to the results generated during the execution process.

Performance Criteria Analysis

Solution S1 identifies the configuration with the highest forensic performance within the space of 320 combinations. Six alternative selection criteria were evaluated: minimization of CRI, maximization of WSUM_AHP, minimization of TOPSIS rank, ELECTRE rank, PROMETHEE rank, and the Borda Count method applied to the 7 methods in Table 6. Paradowski et al. [10] recommend comparing multiple selection criteria to verify solution invariance under changes of method [36]. Table 7 presents the results for the 6 criteria.

 doi:10.20944/preprints202606.0966.v1

Table 7. Criterios de Rendimiento.

S1 Criterion	Selected Config (abbreviated)	WSUM_AHP	STD	CRI
min CRI*	C002+yolo11+qwen2vl-HC+MIS-highcap	0.7993	0.4330	1.14
max WSUM_AHP	C002+yolo11+qwen2vl-HC+MIS-highcap	0.7993	0.4330	1.14
min TOPSIS_rank	C002+yolo11+qwen2vl-HC+MIS-highcap	0.7993	0.4330	1.14
min ELECTRE_rank	C002+yolo11+Q2VL-HC+MIS-HC	0.7993	0.4330	1.14
min PROMETHEE_rank	C002+swin+Q2VL-HC+MIS-HC	0.7679	0.4171	1.14
Borda Count (7 metodos)	C002+yolo11+qwen2vl-HC+MIS-highcap	0.7993	0.4330	1.14

* min CRI denotes the minimum value of the Consensus Rank Index. The convergence of 5/6 criteria on the same configuration confirms that S1 is genuinely optimal.

Robustness Criteria Analysis

Solution S2 identifies the configuration with the highest forensic robustness within the space of 320 combinations. Several filtering criteria were analyzed to identify the set of robust solutions. Table 8 presents the selected criteria. The STD value is computed over the 4 compensatory methods (WSum, WProd, Tcheby, ASF). Criterion C5 = Min(STD×DU) constitutes the final robustness selection step. The threshold values (P25 = 77.71, DU = 1.553, STD = 0.257) correspond to statistical results computed from the execution metrics over all N = 320 combinations.

Table 8. Robustness criteria.

Criterion	Name	Threshold	Methodological Justification	n configs*	Robustness Type
C1	CRI Filter	CRI ≤ P25 = 77.71	Inter-method concordance. Eliminates 75% with lowest consensus.	79	Global methodological
C2	Pareto Filter	Pareto 3D no-dominado [37]	Eliminates dominated configs in (WSUM, STD, DU).	63	Structural
C3	DU Filter	DU L2 ≤ media = 1.553	Configs close to the utopian point [38]	150	Geometric
C4	STD Filter	STD ≤ media = 0.257	Internal consistency of compensatory family.	200	Compensatory
C5	min(STD×DU)	Min STD×DU	Final point-selector. Must be applied last.	1	Compound minimum

* n configs indicates the number of combinations satisfying the criterion.

Los The selected robustness criteria exhibit sequential filtering behavior:

 doi:10.20944/preprints202606.0966.v1

320 combinations → Filter C1 → Filter C2 → Filter C3 → Filter C4 → Filter C5

Or equivalently, an intersection behavior:

320 combinations → Filter C1 ∩ Filter C2 ∩ Filter C3 ∩ Filter C4 ∩ Filter C5

Both operations yield the same robust combination or configuration.

3.6.5. Layer 5 — Performance and Robustness Configuration

This section presents the selected criteria for the optimal performance and robustness configurations.

Regarding the performance criterion: after analyzing the 6 candidate criteria in Layer 4, min(CRI) is selected, as it exhibits the highest consensus among all evaluated criteria.

Regarding the robustness criterion: no single criterion is selected as in the performance case; instead, the integration of criteria is applied sequentially or by intersection.

4. Experimental Setup

4.1. Dataset

The multimodal dataset was constructed using different record counts for training, validation, and testing in each framework phase, as shown in Table 9. The 80/10/10 (train/val/test) split follows best practices for computer vision datasets in moderate-resource settings [39]. The 32,000 training images for F1 and F2 are sufficient to achieve stable convergence in YOLO and pre-trained Transformer models via transfer learning [40]. For the VLM phase (F3), 45,000 image-text instruction pairs represent an adequate volume for fine-tuning 7B-parameter models with LoRA [41,42]. The LLM phase (F4) with 109,676 training instructions exceeds the recommended minimum threshold of 50,000 examples for domain specialization in LLMs via LoRA adapters [43].

Table 9. Multimodal Dataset records.

Multimodal Dataset	Train	Validation	Test
Detection	32,000 Images	6,000 Images	6,000 Images
Classification	32,000 Images	6,000 Images	6,000 Images
VLM	45,000 Images/Instructions	5,900 Images/Instructions	5,800 Images/Instructions
LLM	109,676 Instructions	8,580 Instructions	17,000 Instructions

4.2. Computing Environment

Experiments were conducted on the RunPod platform, a cloud environment providing on-demand GPU resources [44]. Two NVIDIA RTX A4500 GPUs were used (20,470 MB VRAM each), yielding a total of 40.9 GB VRAM. This GPU was selected for its cost-performance trade-off in fine-tuning 7B-parameter LLMs and VLMs [45]. Memory management techniques such as gradient checkpointing and mixed precision (FP16/BF16) were essential to keep training within available VRAM limits [42,46]. The software stack comprised PyTorch 2.x with CUDA 12.x, Transformers 4.x (HuggingFace), and the PEFT library for LoRA adapter implementation [41,47].

4.3. Hyperparameter Configuration

4.3.1. Detection

YOLOv11s was selected for its balance between precision and inference latency, being the 'small' variant of the YOLOv11 architecture [3], as shown in Table 10. Grid search over batch size ∈ {64, 128} and learning rate ∈ {0.001, 0.010} is a standard practice for hyperparameter optimization in YOLO-based object detection [48]. The use of SGD with momentum = 0.937 follows the recommended

 doi:10.20944/preprints202606.0966.v1

configuration by Jocher et al. [3] for stable convergence on moderately sized datasets. AMP (Automatic Mixed Precision) reduces VRAM consumption without significant accuracy penalties in detection [46]. The Mosaic augmentation technique (value 1.0) combines 4 images during training, improving generalization in traffic scenes with multiple objects [49].

Table 10. Fixed hyperparameters.

Parameter	Value
Model	YOLOv11s (small)
Input Resolution	imgsz = 640
Optimizer	SGD with momentum = 0.937
Weight Decay	0.0005
Warmup Epochs	3
Epochs	30
Patience	10
Mosaic	1.0
AMP	enabled
Workers	8
Cache	disk
Fraction	1.0

Table 11 presents the values of the 3 variable hyperparameters applied during fine-tuning of the 4 candidate models (combo_001 through combo_004).

Table 11. Variable hyperparameters.

Combo*	batch	lr0	lrf
001	64	0.001	0.01
002	128	0.001	0.01
003	64	0.010	0.01
004	128	0.010	0.01

* Combo denotes the candidate model configuration name.

4.3.2. Clasification

The selection of five heterogeneous architectures (YOLO11-cls, InceptionV3, ResNet50, Swin Transformer, ViT) follows the principle of exhaustive comparison of architectural families for evidence-based selection [50]. The use of AdamW optimizer for Transformer models (Swin, ViT) instead of SGD responds to the sensitivity of attention mechanisms to learning rate and the need for adaptive weight decay regularization [51]. The drop_path = 0.2 configuration in Swin Transformer and drop_rate = 0.1 in ViT represent standard stochastic regularization values that prevent overfitting during fine-tuning with medium-sized datasets [52]. Label smoothing (label_smoothing = 0.1) in ViT reduces model overconfidence and improves probability calibration [53], as shown in Table 12.

 doi:10.20944/preprints202606.0966.v1

Table 12. Hyperparameter configuration.

Model	Type	imgsz	Optimizer	Epochs	Variant/Notes
YOLOv11s-cls	YOLO cls	224	Adam	30	Native YOLO11 classifier, warmup 3 epochs
InceptionV3	CNN	299	Adam	30	aux_logits=True; initial error corrected
ResNet50	CNN	224	Adam	30	step_size=7, gamma=0.1, freeze=0
Swin Transformer	Transformer	224	AdamW	30	swin_base_patch4_window7_224, drop_path=0.2
ViT	Transformer	224	AdamW	30	vit_base_patch16_224, drop_rate=0.1, label_smooth=0.1

4.3.3. Vision-Language Model (VLM)

Phase 3 of the multimodal framework evaluated two 7B-parameter Vision-Language Model (VLM) architectures through LoRA fine-tuning on the traffic accident dataset. Both models use the same LoRA adapter configuration (r=16, alpha=32, modules q/k/v/o_proj), but differ significantly in their visual architecture, tokenizer, and chat template, as shown in Table 13.

The LoRA configuration (r=16, alpha=32) applied to q/k/v/o_proj modules is consistent with the recommendations of Hu et al. [7] for fine-tuning 7B-parameter VLMs with moderate computational resources. Rank r=16 provides sufficient expressive capacity for specialized domains without compromising training stability [41]. The ShareGPT format for fine-tuning instructions is the standard adopted by LLaVA-Instruct [4,15] and enables generalization of instructions to multiple types of visual tasks. Evaluation on 5,800 test images with BLEU-1/4, METEOR, ROUGE-1/2/L, BERTScore, CIDEr, CLIPScore, and Perplexity follows the comprehensive evaluation protocol for captioning models [54,55].

Table 13. Hyperparameters for candidate VLM models.

Hyperparameter	LLaVA-NeXT	Qwen2-VL	Notes
Base model	llava-hf/llava-v1.6-mistral-7b-hf	Qwen/Qwen2-VL-7B-Instruct	Different VLM architectures
Training epochs	2	2	Same epochs — controlled comparison
Initial learning rate	2e-4	2e-4	Identical in both models
LR scheduler	Linear (warmup deprecated)	Linear (warmup deprecated)	Warmup ratio flagged as deprecated
Gradient accum. steps	32	32	Effective batch = 32 in both
Per-device train batch	1	1	GPU memory constraint for VLM
Max tokens (filter)	2,048	No explicit limit	LLaVA-NeXT filtered dataset to 35% valid

 doi:10.20944/preprints202606.0966.v1

LoRA rank r	16	16	F3 baseline configuration
LoRA alpha	32	32	Alpha/r ratio = 2.0
LoRA dropout	0.0	0.0	No dropout regularization
Target modules	q,k,v,o_proj	q,k,v,o_proj	Same modules in both
PEFT version	0.18.1	0.18.1	Identical
Dataset format	ShareGPT (VLM)	ShareGPT (VLM)	Image-text instruction pairs
Effective batch size	32	32	gradient_accum × per_device_batch

4.3.4. Large Language Model (LLM)

The high-capacity LoRA configuration (r=32, alpha=64) for LLMs in F4 increases the domain adaptation capacity for forensic tasks compared to the base configuration (r=16) [41,43]. Mistral-7B, with its sliding window attention architecture and grouped query attention (GQA), offers better inference efficiency for long forensic context sequences [6]. The Alpaca instruction format [23] is the most reproducible standard for fine-tuning LLMs in specialized domains with LLM-generated synthetic datasets, as detailed in Table 14.

Table 14. LLM hyperparameters.

Hyperparameter	Mistral-7B	Qwen2-7B	Notes
Base model	mistralai/Mistral-7B-Instruct-v0.2	Qwen/Qwen2-7B-Instruct	Pure LLMs without visual encoder
Training epochs	2	2	Identical
Initial learning rate	2e-4	2e-4	Identical
LR scheduler	Cosine	Cosine	Different from F3 (linear)
Warmup ratio	0.1	0.1	Identical — 10% of total steps
Per-device train batch	4	4	Identical
Gradient accum. steps	4	4	Effective batch = 16
Effective batch size	16	16	per_device × grad_accum
LoRA rank r	32	32	HighCap: 2× vs F3 (r=16)
LoRA alpha	64	64	HighCap: 2× vs F3 (alpha=32)
LoRA dropout	0.05	0.05	F3 used 0.0 — regularization added
Target modules	q,k,v,o_proj	q,k,v,o_proj	Identical to F3
Task type	CAUSAL_LM	CAUSAL_LM	Autoregressive generation

 doi:10.20944/preprints202606.0966.v1

PEFT version	0.18.1	0.18.1	Identical across all phases
Dataset format	Alpaca (LLM)	Alpaca (LLM)	Plain-text instruction pairs

5. Analysis and Results

5.1. Analysis of Trained Models per Phase

Phase 1 — Detection.

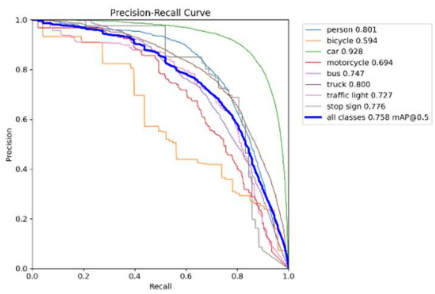
Table 15 presents the fine-tuning metric results for candidate models in the detection phase. combo_001 leads in mAP@50 and mAP@50-95, whereas combo_002 leads in Precision.

Table 15. Detection metrics.

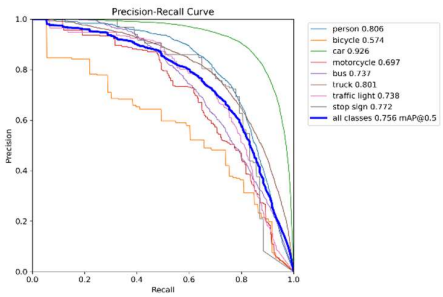
Combo	mAP50	mAP50-95	Precision	Recall	box loss	cls loss	Time (h)
001	0.7574	0.6502	0.7137	0.6978	0.4513	0.4943	1.24
002	0.7526	0.6425	0.7330	0.6868	0.4534	0.4964	1.02
003	0.7378	0.6229	0.7123	0.6778	0.4783	0.5233	1.16
004	0.7395	0.6252	0.7005	0.6908	0.4630	0.4943	1.12

The selection of combo_002 as the optimal candidate model in F1 responds to three factors: (1) higher Precision (0.7330 vs. 0.7137), minimizing false positives in forensic detection; (2) lower training time (1.02 h), demonstrating greater computational efficiency with batch=128; and (3) comparably low training losses, indicating equivalent convergence quality at lower cost. The mAP@50 of combo_001 (0.7574) marginally exceeds that of combo_002 (0.7526), but this 0.005-point difference is statistically marginal and is outweighed by the above advantages when the MCDM protocol is applied over 49 normalized metrics.

Figure 2 presents the evaluation of 4 YOLOv11s hyperparameter combinations on the multimodal traffic accident dataset. All 4 combinations converged successfully. The optimal combination (combo_001) achieved mAP@50 = 76% and mAP@50-95 = 65% on the final test set, with good generalization relative to the validation set.

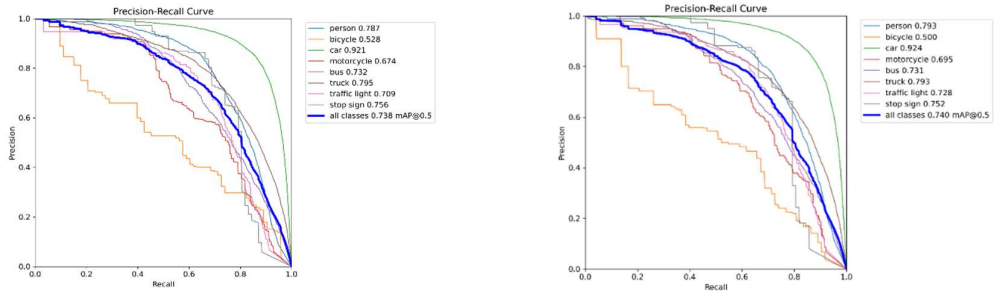


(a)



(b)

 doi:10.20944/preprints202606.0966.v1



(c) (d)

Figure 2. Recall vs. Precision curves for fine-tuned models. The following is described: (a) graph corresponding to combo1; (b) graph corresponding to combo2; (c) graph corresponding to combo3; (d) graph corresponding to combo4.

Phase 2 — Classification.

Table 16 presents accuracy and training time results for candidate model configurations per base architecture.

Table 16. Classification metrics.

Base Model	Combo	Batch	lr0	Best accuracy	Time Hrs
YOLOv11s-cls	1	64	0.001	83.11%	0.146
	2	128	0.001	84.23%	0.087
	3	64	0.01	68.29%	0.147
	4	128	0.01	70.35%	0.087
InceptionV3	1	32	0.001	82.90%	0.812
	2	64	0.001	82.94%	0.519
	3	32	0.01	69.23%	0.818
	4	64	0.01	74.30%	0.519
ResNet50	1	32	0.001	80.24%	0.458
	2	64	0.001	81.19%	0.309
	3	32	0.01	73.78%	0.463
	4	64	0.01	74.30%	0.308
Swin Transformer.	1	16	0.0001	85.28%	1.493
	2	32	0.0001	85.91%	1.100
	3	16	0.0005	62.45%	1.494
	4	32	0.0005	77.62%	1.096
Vision Transformer (ViT).	1	16	0.0001	81.68%	1.183
	2	32	0.0001	83.50%	0.896
	3	16	0.0005	71.12%	1.157
	4	32	0.0005	76.15%	0.895

Table 16 shows that the best candidate models in terms of architecture are those in Combo 2, whose metrics for accuracy and runtime are superior to those of the others.

 doi:10.20944/preprints202606.0966.v1

All models successfully completed training; Swin Transformer achieved the highest validation accuracy (85.91%), whilst YOLOv11s-cls was the most efficient in terms of training time (5.2 minutes) with an accuracy of 84.23%, as shown in Table 17.

Table 17. Accuracy metric for classification-phase candidate models.

Metric	YOLOv11s-cls	InceptionV3	ResNet50	Swin	ViT
Accuracy	84.23%	82.94%	81.19%	85.91%	83.50%
Time (h)	0.087 h	0.519 h	0.309 h	1.100 h	0.896 h

Figure 3 analysis: (a) C2 (large batch, low lr) is the optimal combination across all models; high lr (C3 and C4) severely penalizes all architectures. (b) Swin converges highest and most stably; ViT shows greater variability in final epochs; YOLOv11s-cls serves as a horizontal reference. (c) Swin maintains the lowest loss; InceptionV3 exhibits progressive val_loss increase despite stable accuracy, a sign of latent overfitting. (d) YOLOv11s-cls is 12.7× faster than Swin; Transformers require significantly more training time than CNNs. (e) YOLOv11s-cls is Pareto-efficient (highest accuracy per compute hour); Swin offers the highest absolute accuracy at greater time cost. (f) YOLOv11s-cls stands out in speed and consistency; Swin in accuracy; ViT shows lower stability.

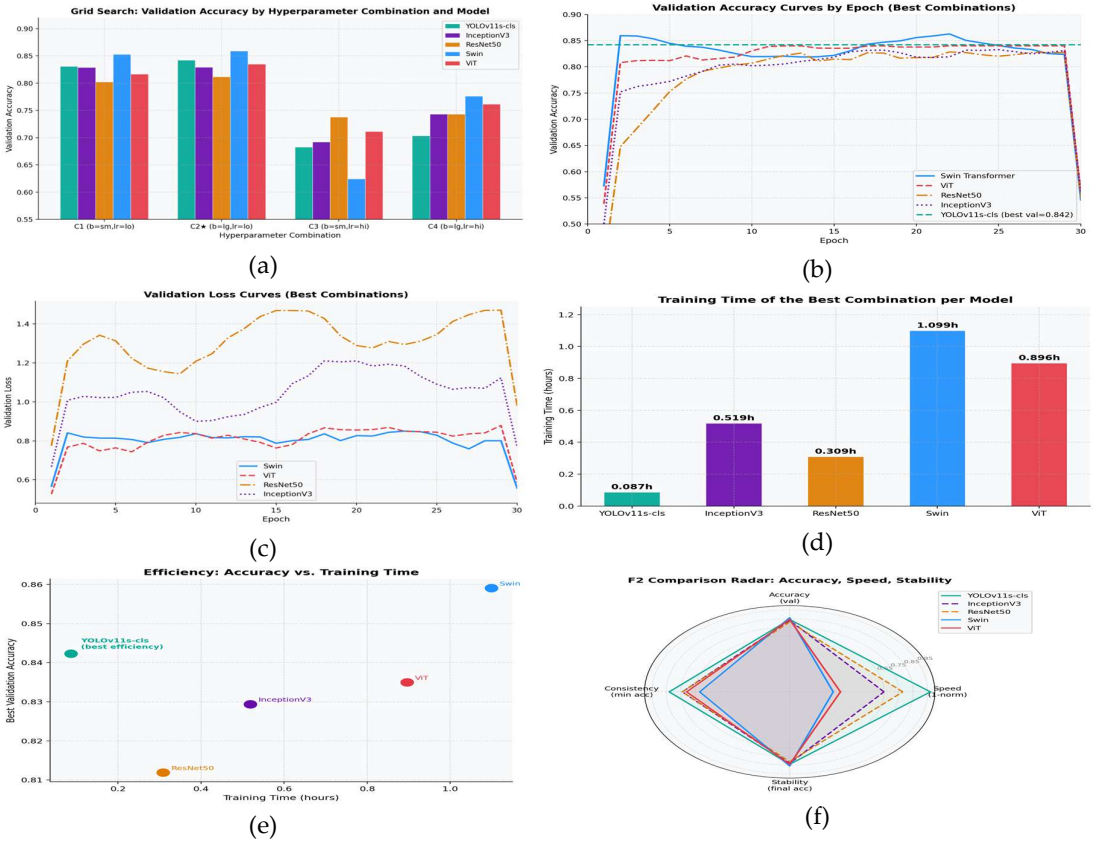


Figure 3. Comparative metric analysis across candidate models. The following is described: (a) Accuracy by hyperparameter combination for the 5 models; (b) Validation accuracy curves by epoch (best combinations); (c) Validation loss curves by epoch; (d) Training time for the best combination per model; (e) Efficiency: accuracy vs training time; (f) Comparative radar plot: accuracy, speed, stability (final/best accuracy) and consistency (minimum accuracy across the grid).

Phase 3 — Vision-Language Model (VLM).

 doi:10.20944/preprints202606.0966.v1

Fine-tuning results for VL 3 are presented in Table 18.

Table 18. VLM training metrics.

Training Metric	LLaVA-NeXT	Qwen2-VL	Observation
Initial loss (step 1)	11.670	17.100	Qwen2-VL starts higher — more aggressive warmup
Final train loss (epoch 2)	4.133	7.138	High VLM losses are expected in this domain
Last step loss	3.961	7.024	Qwen2-VL stabilizes ~7.0 from epoch 0.5
Eval loss (final checkpoint)	3.965	5.868	Qwen2-VL: 5 checkpoints; LLaVA-NeXT: 1
Relative loss reduction	64.6%	58.3%	Based on initial → final train loss
Initial grad norm	0.304	0.158	Controlled convergence in both models
Final grad norm	0.003	0.003	Identical final gradients
Total steps	982	2,786	Proportional to training samples
Training samples	15,685	44,559	Qwen2-VL: 2.84× more data
Val samples	2,693	5,860	Val ratio: 14.7% vs 11.6%
Samples/sec (throughput)	1.032	1.523	Qwen2-VL 47.5% higher throughput
Total runtime	30,390 s (8.44 h)	58,510 s (16.25 h)	Dataset volume difference
Trainable LoRA params	15,990,784 (0.211%)	10,092,544 (0.122%)	Over ~7–8B total parameters
Evaluation checkpoints	1	5	Qwen2-VL: better training monitoring
Dataset filtering (max_tokens=2048)	15,685/44,559 (35%)	44,559/44,559 (100%)	LLaVA-NeXT applied token-length filter

From Table 18, the following is observed:

- LLaVA-NeXT effectively received only 15,685 samples (35% of the dataset) due to the max_tokens=2048 limit. Qwen2-VL utilized all 44,559 samples (100%). This 2.84× difference in training volume structurally explains the performance gap between both models in evaluation metrics (Table 26). The token restriction in LLaVA-NeXT is intrinsic to its tokenizer, which generates longer sequences than the Qwen2-VL tokenizer for the same content.
- Both models achieved an identical final grad_norm (0.003), indicating that both fine-tunings converged healthily. The difference in final train_loss (3.961 vs. 7.024) is not directly comparable because the output spaces differ: LLaVA-NeXT and Qwen2-VL have distinct vocabularies and token distributions. VLM loss is a function of the vocabulary and is not comparable across architectures.
- Qwen2-VL processes 1.523 samples/sec vs. 1.032 for LLaVA-NeXT (+47.5%). This efficiency results from its dynamic-resolution visual encoder, which reduces the number of visual tokens

 doi:10.20944/preprints202606.0966.v1

for standard-sized images, optimizing cross-attention computation. This advantage is also confirmed at inference time (Table 26: 1,061 ms vs. 5,817–5,940 ms).

Figure 4 analysis (VLM): (a) Complete train loss curve (epochs 0–2). Red dot marks the single evaluation checkpoint (eval_loss = 3.965 at epoch 1.018). (b) Complete train loss curve (epochs 0–2). Red dots mark 5 uniformly distributed evaluation checkpoints. (c) Overlay comparison of F3 train loss curves; scales differ due to each model's architecture. (d) The drop at epoch ~0.10 coincides with warmup completion and high stability throughout the descent phase. (e) Stable behavior with occasional moderate spikes, normal in multimodal fine-tuning.

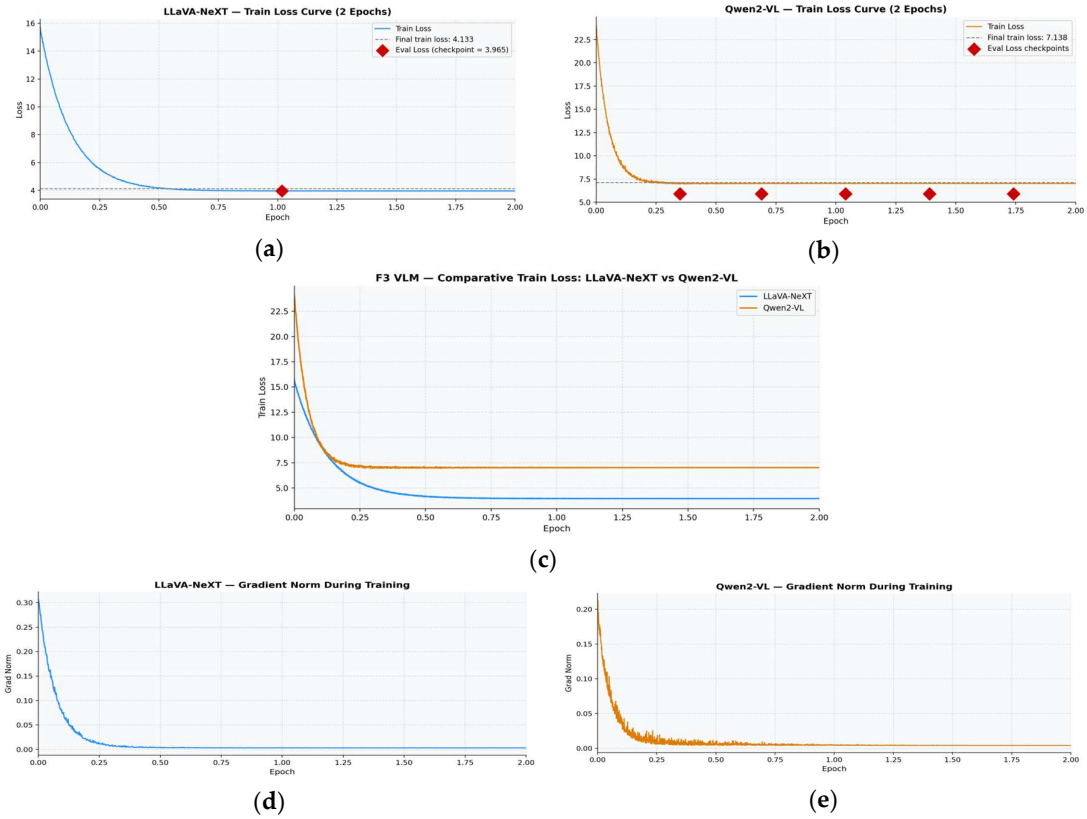


Figure 4. Fine-tuning metrics for VLM models. The following is described: (a) LLaVA-NeXT — Train Loss; (b)Qwen2-VL — Train Loss; (c) Comparativa overlay — LLaVA-NeXT vs Qwen2-VL; (d) LLaVA-NeXT — Grad Norm; (e) Qwen2-VL — Grad Norm.

Phase 4 — Large Language Model (LLM).

Fine-tuning results for Phase 4 are presented in Table 19.

Table 19. LLM training metrics.

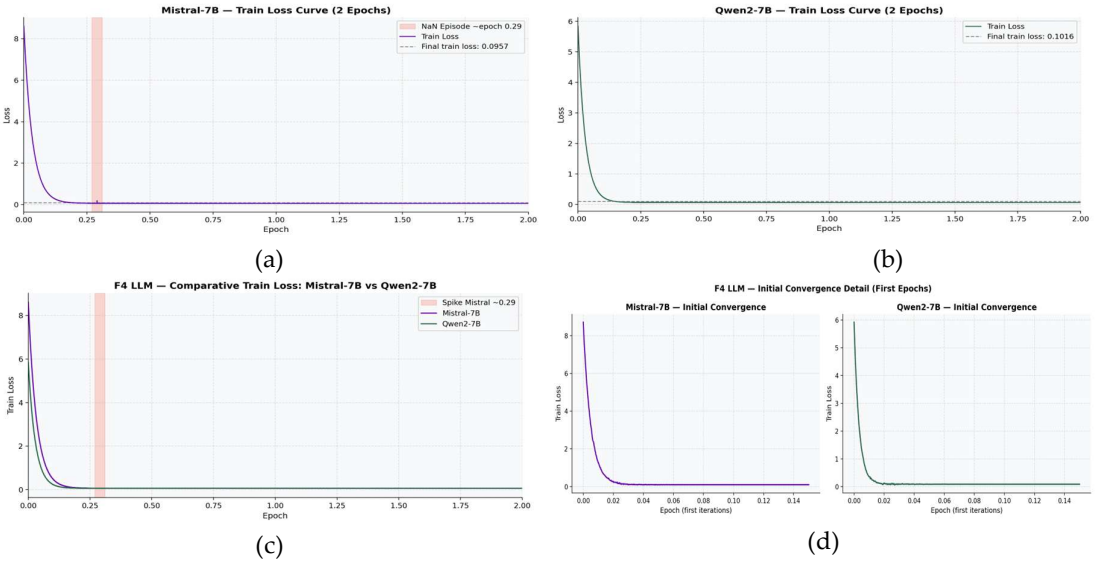
Metric	Mistral-7B	Qwen2-7B	Observation
Initial loss (step 1)	8.542	5.775	Mistral starts higher due to SWA architecture
Final train loss (epoch 2 avg)	0.09565	0.10162	Mistral marginally lower
Last step loss	0.06305	0.06486	Minimal difference at plateau
Relative loss reduction	98.9%	98.2%	Both >98%: effective convergence

 doi:10.20944/preprints202606.0966.v1

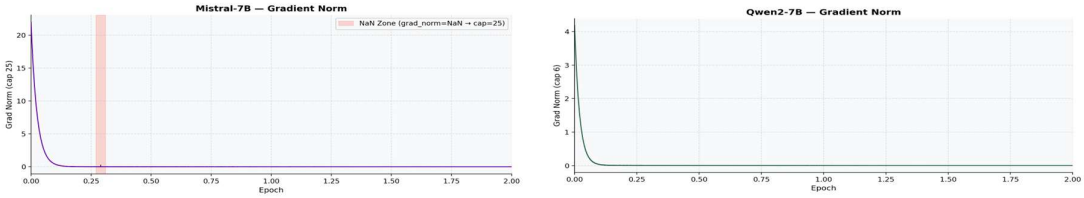
Initial grad norm	21.92	4.181	Mistral: more volatile initialization
Final grad norm	0.040	0.030	Both stabilized at plateau
Grad spike (Mistral)	NaN→156.8→106.5	N/A	Isolated episode at epoch ~0.29
Total steps	13,710	13,710	Identical — same dataset
Samples/sec (throughput)	3.177	3.543	Qwen2-7B 11.5% faster
Total runtime	69,040 s (19.18 h)	61,900 s (17.19 h)	~1.98 h difference
Trainable LoRA params	27,262,976 (0.375%)	20,185,088 (0.264%)	Mistral: larger hidden dimension
Effective batch	16	16	per_device=4 × grad_accum=4
Eval checkpoints	0	0	No eval steps recorded in F4

From Table 19: fine-tuned Mistral-7B (98.9%) and Qwen2-7B (98.2%) achieve nearly identical convergence levels, with final train_loss of 0.063 and 0.065 respectively. Both models effectively specialized in forensic report generation. Performance differences in evaluation metrics reflect architectural differences, not differences in fine-tuning quality.

Figure 5 analysis (LLM): (a) Complete fine-tuning (epochs 0–2). Red zone highlights the instability episode at epoch ~0.29. Dashed line indicates the average final train_loss. (b) Complete fine-tuning (epochs 0–2). Clean convergence throughout. (c) Mistral spike at epoch ~0.29 (red zone) is the only visible divergence point between both curves. (d) Detail of initial convergence in the first 0.15 epochs; Qwen2-7B converges slightly faster from a lower starting point. (e) Mistral-7B gradient norm (limited to 25): red zone marks the NaN spike and peaks of 156.8 and 106.5 at epoch ~0.29. (f) Qwen2-7B gradient norm (limited to 6): clean behavior throughout, consistent with cosine LR decay.



 doi:10.20944/preprints202606.0966.v1



(e) (f)

Figure 5. LLM training metric results: (a) **Mistral-7B — Train Loss;** (b) Qwen2-7B — Train Loss; (c) Comparativa overlay — Mistral-7B vs Qwen2-7B; (d) initial convergence (early stages); (e) Mistral-7B — Gradient norm; (f) Qwen2-7B — Gradient norm.

5.2. Base vs. Fine-Tuned Model Comparison per Phase

Phase 1 — Detection.

Table 20 presents the comparison of metrics between the base model and the 4 fine-tuned candidate models for F1. Dataset-agnostic metrics were used for this comparison.

Table 20. Detection metrics — base vs. fine-tuned.

Model	Tipo	Object Recall	Miss Rate	IoU medio	IoU ≥ 0.75	Latency Media ms	Throughput img/s
yolo11s_base	Base	0.449	55.1%	0.653	69.9%	20.53	48.79
combo_001	Fine-tuning	0.758	24.2%	0.839	89.1%	20.44	49.40
combo_002	Fine-tuning	0.756	24.4%	0.836	88.8%	20.20	49.57
combo_003	Fine-tuning	0.745	25.5%	0.825	87.7%	20.23	49.47
combo_004	Fine-tuning	0.745	25.5%	0.827	87.8%	20.24	49.45

Fine-tuning produced a 68.8% improvement in Object Recall (from 0.449 to 0.758), meaning the base detector found only 1 in every 2 real vehicles, while the fine-tuned model detects 3 out of 4. Miss Rate decreased from 55.1% to 24.4% — the base model failed on more than half of real objects. Critically, latency did not increase (20.53 → 20.20 ms), confirming that fine-tuning improves precision at no additional inference speed cost, a critical requirement for real-time urban video surveillance systems.

Phase 2 — Classification.

Table 21 presents the metric comparison between each base architecture and its best fine-tuned candidate model. Dataset-agnostic metrics were used.

Table 21. Classification metrics — base vs. fine-tuned.

Model	Type	Arq.	Confidence	Lat. med (ms)	Throughput	VRAM total	Quality Rank	Efficiency Rank
yolo11_base	Base	Yolo11	0.458	57.6	N/D	N/D	—	—
inceptionv3_base	Base	Inc.V3	0.340	115.3	9.02	126.7 MB	—	—
resnet50_base	Base	ResNet50	0.322	110.8	9.22	122.0 MB	—	—
swin_base	Base	Swin	0.344	112.8	9.13	137.7 MB	—	—

 doi:10.20944/preprints202606.0966.v1

vit_base	Base	ViT	0.392	115.3	8.87	359.4 MB	—	—
production (FT)	FT	Yolo11	0.962	101.3	9.83	6.6 MB	1°	1°
inceptionv3_FT	FT	Inc.V3	0.951	142.6	7.13	97.7 MB	3°	5°
resnet50_FT	FT	ResNet50	0.952	133.5	7.46	117.2 MB	2°	4°
swin_FT	FT	Swin	0.902	119.2	8.22	349.6 MB	4°	3°
vit_FT	FT	ViT	0.914	111.0	8.69	381.2 MB	3°	2°

The most notable transformation from fine-tuning in F2 is model confidence. Base models show a mean confidence of 0.371 (range 0.322–0.458), indicating high uncertainty in predictions on accident images that differ from the ImageNet distribution. Fine-tuned models reach a mean confidence of 0.936 (range 0.902–0.962), a +152% improvement. This calibration is critical in forensic systems, as low-confidence models produce ambiguous and inconsistent reports.

Phase 3 — Vision-Language Model (VLM).

Table 22 presents the metric comparison between each base model and its fine-tuned variant.

Table 22. VLM metric ranking — dataset-agnostic metrics.

Model	Perplexity	CLIPScore	Latencia	Throughput	Estab. VRAM	Verbosity	Total Score	Ranking*
llavanext_base	3rd	2nd	3rd	3rd	3rd	3rd	17 pts	3rd
llavanext_highcap	4th	4th	4th	4th	1st	4th	21 pts	4th
qwen2vl_base	2nd	3rd	2nd	2nd	4th	2nd	15 pts	2nd
qwen2vl_highcap	1st	1st	1st	1st	1st	1st	6 pts	1st

*Ranking by position on each metric (1=best, 4=worst). Lower total score = better overall performance.

From Table 22:

- qwen2vl_highcap ranks 1st in Perplexity, CLIPScore, Latency, Throughput, and Verbosity, tied for 1st in VRAM Stability. This dominance across all dimensions is an uncommon result in multimodal benchmarks and demonstrates that combining larger training volume (44,559 complete samples) with Qwen2-VL's dynamic vision architecture produces simultaneous advantages in quality and efficiency.
- llavanext_highcap ranks 4th globally and is the only fine-tuned model that performs WORSE than its base variant (llavanext_base, 3rd). The only metric where highcap surpasses base is VRAM Stability (1st vs. 3rd). This confirms the overfitting documented in Table 18: with only 15,685 effective training samples (35% of the dataset due to token filtering), high-capacity fine-tuning memorized the available data instead of generalizing, degrading the model's descriptive capacity.

Phase 4 — Large Language Model (LLM).

Table 23 presents the metric comparison between each base model and its fine-tuned variant.

 doi:10.20944/preprints202606.0966.v1

Table 23. LLM metric ranking — dataset-agnostic metrics.

Modelo	Perplexity	MMLU	MAUVE	Latency	Throughput	Extra VRAM	Score pts	Ranking*
mistral_base	3rd	3rd	2nd	4th	4th	3rd	19	3rd
mistral_highcap	1st	1st	1st	2nd	3rd	1st	9	1st
qwen2_base	4th	4th	4th	3rd	2nd	3rd	20	4th
qwen2_highcap	2nd	2nd	3rd	1st	1st	1st	10	2nd

*Ranking by position on each metric (1=best, 4=worst). Lower total score = better overall performance.

From Table 23:

- Mistral-7B highcap leads in the two most relevant metrics for forensic analysis: MMLU (reasoning, 1st) and MAUVE (text naturalness, 1st). The high MMLU score (0.441 in Table 27, 3× higher than Mistral base) confirms that high-capacity fine-tuning (r=32) effectively transferred specialized forensic reasoning capabilities. High text naturalness (MAUVE=0.807) ensures that generated reports are readable and coherent for human operators.
- Qwen2-7B highcap leads in Latency (1st) and Throughput (1st), with lower response time (2,058 ms vs. 3,594 ms for Mistral-highcap, 43% faster). For deployment environments with high analysis frequency (real-time video surveillance systems), this latency advantage may be decisive. Its 2nd global position with a score of 10 pts (vs. 9 pts for Mistral-highcap) indicates a valid alternative when latency outweighs absolute quality.
- Mistral-highcap (+10 positions vs. Mistral-base) achieves greater fine-tuning gain than Qwen2-highcap (+10 positions vs. Qwen2-base). The symmetric gain confirms that the fine-tuning process was equally effective for both architectures.

5.3. Analysis of Models in the Integrated Phase

This section analyzes metrics in an integrated manner: the input image is processed sequentially through each phase, and the resulting inferred metrics are compared against the multimodal dataset.

Phase 1 — Detection.

Model combo_002 achieves the best overall performance (mAP@50=0.625) and is selected as the highest-performance detector in the MODM-MCDM protocol. The maximum difference between models is 0.006 in mAP@50, demonstrating high training stability, as shown in Table 24.

Table 24. Detection metrics — integrated phase.

Model	mAP@50	mAP@50-95	F1 Score	IoU	Latency (ms)
combo_001	0.619	0.540	0.721	0.835	27.3
combo_002	0.625	0.548	0.726	0.838	27.2
combo_003	0.621	0.544	0.723	0.838	27.2
combo_004	0.621	0.544	0.725	0.837	27.1

 doi:10.20944/preprints202606.0966.v1

Phase 2 — Classification.

Table 25 presents the architecture ranking by F1 Score:

- Swin Transformer: best F1 Score (90.00%). Hierarchical attention captures long-range relationships between scene elements, ideal for distinguishing dense and sparse traffic.
- YOLO11-cls: second place (86.75%) with lowest latency (27.9 ms). Optimal performance/speed trade-off for real-time deployment.
- ViT: third place (84.50%) with good speed (52.3 ms). Global attention is effective but lacks Swin's local hierarchy.
- InceptionV3: fourth place (80.00%). Multi-scale inception modules with higher latency than Swin and lower accuracy.
- ResNet50: fifth place (79.75%). Lower F1 Score but competitive latency (53.3 ms) with residual connections.

Table 25. Classification metrics — integrated phase.

Architecture	F1 Score	Latency (ms)
YOLO11-cls	0.8675	27.93
Swin Transformer	0.9000	108.96
ViT	0.8450	52.32
InceptionV3	0.8000	86.73
ResNet50	0.7975	53.25
Media global	0.8420	65.84

Phase 3 — Vision-Language Model (VLM).

Table 26 presents evaluation metrics. The fine-tuned Qwen2-VL highcap model dominates across all metrics:

- BLEU-1: 0.548 vs. 0.109 average of the other three (5× higher) — unigram descriptions better match references.
- BLEU-4: 0.234 vs. 0.011 average (21× higher) — significantly better phrase-level coherence.
- BERTScore: 0.919 vs. 0.845 — superior semantic quality by 8.8 percentage points.
- Latency: 1,061 ms vs. 5,626 ms average — 5.3× faster, while also being the highest-performance model.
- LLaVA-NeXT highcap paradoxically achieves lower BERTScore than its base version (0.836 vs. 0.847), suggesting possible fine-tuning overfitting.

Table 26. VLM metrics — integrated phase.

VLM Model	Bleu-1	Bleu-4	Meteor	Rouge-1	Rouge-L	BertScore	Cider	ClipScore	Lat. ms
LLaVA-NeXT base	0.1095	0.0125	0.2377	0.2019	0.1519	0.8471	0.0642	0.5863	5,817
LLaVA-NeXT highcap	0.0985	0.0140	0.2131	0.1865	0.1459	0.8364	0.0573	0.5126	5,940

 doi:10.20944/preprints202606.0966.v1

VLM Model	Bleu-1	Bleu-4	Meteor	Rouge-1	Rouge-L	BertScore	Cider	ClipScore	Lat. ms
Qwen2-VL base	0.1086	0.0078	0.2369	0.2063	0.1470	0.8513	0.0582	0.5618	5,121
Qwen2-VL highcap	0.5482	0.2337	0.4686	0.5815	0.5138	0.9192	0.3173	0.6822	1,061
Media global	0.2162	0.0670	0.2891	0.2940	0.2397	0.8635	0.1243	0.5857	4,485

Phase 4 — Large Language Model (LLM).
Table 27 presents comparative results:

- Mistral-7B highcap dominates all metrics: BERTScore=0.917, MMLU=0.441, MAUVE=0.807. Generates semantically more precise and naturally worded forensic analyses.
- Qwen2-7B highcap: second place with BERTScore=0.873 and lowest latency (2,058 ms). Good speed/quality trade-off.
- Mistral-7B base: third place with MAUVE=0.712 superior to Qwen2-7B base (0.591), indicating generated text is more human-like, though with lower factual precision.
- Qwen2-7B base: lowest performance in BLEU and METEOR but moderate latency (4,158 ms).
- MMLU confirms Mistral-7B HC superiority in reasoning: 0.441 vs. 0.141 for Mistral base (3× higher) — decisive for accident forensic analysis.

Table 27. LLM metrics — integrated phase.

Model	Bleu-1	Bleu-4	Meteor	Rouge-1	Rouge-L	BERTScore	Mmlu	Mauve	Lat. ms
Mistral-7B base	0.1773	0.0095	0.1418	0.2452	0.1396	0.8332	0.1411	0.7115	5,096
Mistral-7B highcap	0.4649	0.2421	0.4520	0.5215	0.4830	0.9172	0.4407	0.8072	3,594
Qwen2-7B base	0.1475	0.0063	0.1365	0.2405	0.1303	0.8268	0.1508	0.5911	4,158
Qwen2-7B highcap	0.2341	0.0810	0.2193	0.3272	0.2601	0.8730	0.2316	0.6502	2,058
Media global	0.2560	0.0847	0.2374	0.3336	0.2533	0.8626	0.2411	0.6900	3,726

The cross-phase metric analysis (Table 28) reveals the following:

- Qwen2-VL highcap is the robustly and consistently optimal VLM. It leads in BLEU-4 (0.0208), METEOR (0.1650), ROUGE-1 F (0.2787), and BERTScore (0.6566), with advantages ranging from 18% over the second-best in BLEU-4 to 62% in BERTScore. This robustness to LLM and metric choice makes its selection as the F3 component the unconditionally optimal decision.
- The globally optimal combination is qwen2vl_highcap + qwen2_base. This combination maximizes BERTScore (0.7772) and ROUGE-1 F (0.2844). The fact that qwen2_base outperforms qwen2_highcap indicates that LLM fine-tuning does not add semantic advantage when the VLM input is of high quality.

 doi:10.20944/preprints202606.0966.v1

- LLaVA-NeXT highcap exhibits a degradation pattern relative to its base variant in three of the four metrics, with the largest penalty in BERTScore (−22 percentage points). This confirms that fine-tuning was not effective for the traffic accident description domain.
- The VLM factor (F3) determines report quality to a greater extent than the LLM factor (F4). Inter-VLM variability in BERTScore (mean range: 0.406–0.657) far exceeds inter-LLM variability (total range: 0.449–0.579).

Table 28. Cross-phase VLM × LLM metric relationship.

Model VLM\LLM	Mistral_base	Mistral_highcap	Qwen2_base	Qwen2_highcap	LLM Metrics
llavanext_base	0.0114547	0.010774	0.0092202	0.01116975	bleu4_mean
llavanext_highcap	0.014367	0.01375325	0.01429835	0.0118141	bleu4_mean
qwen2vl_base	0.01128715	0.0128829	0.0106484	0.0086831	bleu4_mean
qwen2vl_highcap	0.0217748	0.02119405	0.0201908	0.01994445	bleu4_mean
llavanext_base	0.1527703	0.14444615	0.1471689	0.1504485	meteor_mean
llavanext_highcap	0.14671755	0.143611	0.13841605	0.13927895	meteor_mean
qwen2vl_base	0.14156	0.1470985	0.15174295	0.14859935	meteor_mean
qwen2vl_highcap	0.15857045	0.15715435	0.1715178	0.17293195	meteor_mean
llavanext_base	0.2640533	0.25793405	0.2602373	0.25939145	rouge1_f_mean
llavanext_highcap	0.2584876	0.258838	0.24074225	0.2427907	rouge1_f_mean
qwen2vl_base	0.2579872	0.26095575	0.27690385	0.26837825	rouge1_f_mean
qwen2vl_highcap	0.2789028	0.27462725	0.2844426	0.27702485	rouge1_f_mean
llavanext_base	0.542997807	0.432938596	0.576826754	0.528024123	bertscore_mean
llavanext_highcap	0.430614035	0.395848684	0.388508772	0.409923246	bertscore_mean
qwen2vl_base	0.411984649	0.417114035	0.572934211	0.52310307	bertscore_mean
qwen2vl_highcap	0.601405702	0.550969298	0.777182018	0.697019737	bertscore_mean

5.4. Performance Configuration

The optimal performance configuration is the criterion selected in Layer 5 of the hybrid MODM-MCDM protocol, represented by min(CRI).

Table 29 presents the top configurations ranked in ascending order by the CRI criterion. The optimal performance configuration occupies rank 1 in the CRI field.

Table 29. Top 10 combinations by Consensus Ranking.

Model per Phase(F1-F4)	Electre	Promethee	Topsis	Wsum	Wprdo	Tcheby	Asf	CRI
combo_002-production_cls_combo_002-qwen2vl_highcap-mistral_highcap	1	1	2	1	1	1	1	1
combo_002-vit_combo_002-qwen2vl_highcap-mistral_highcap	4	6	21	6	1	1	1	2
combo_002-swin_combo_002-qwen2vl_highcap-mistral_highcap	3	2	1	2	1	33	1	3
combo_002-production_cls_combo_002-qwen2vl_highcap-qwen2_highcap	2	3	5	3	1	1	81	4

 doi:10.20944/preprints202606.0966.v1

combo_002-production_cls_combo_002-llavanext_base-mistral_highcap	9	26	13	26	1	1	21	5

Configuration S1 (combo_002 + YOLO11-cls + Qwen2-VL-highcap + Mistral-7B-highcap, WSUM=0.7993, CRI=1) was identified by the convergence of 5/6 alternative selection criteria: min CRI, max WSUM_AHP, min TOPSIS_rank, min ELECTRE_rank, and Borda Count. This convergence validates the method-invariance principle established by Paradowski et al. [29]: genuine solution optimality is demonstrated when multiple independent MCDM methods converge on the same solution.

5.5. Robustness Configuration

The optimal robustness configuration is represented by the integration of criteria analyzed in Layer 5 of the hybrid MODM-MCDM protocol, encompassing Filters C1, C2, C3, C4, and C5.

Table 30 presents configurations ranked in ascending order by the final step of Filter C5. The optimal robustness configuration occupies rank 1 in the Min(STD×DU L2) field.

Table 30. Top 5 combinations by Robustness Criteria Ranking.

Model per Phase(F1-F4)	CRI	STD	DU L2	Pareto	STDxDUL2	Rankig Min(STDxDUL2)
combo_004-swin_combo_002-qwen2vl_base-mistral_highcap	67	0.2183607	1.328015	PARETO	0.2899863	1
combo_004-swin_combo_002-llavanext_base-mistral_highcap	60.4285714	0.2224648	1.3138885	PARETO	0.292294	2
combo_004-production_cls_combo_002-llavanext_highcap-mistral_highcap	66.2857143	0.2244521	1.3183712	PARETO	0.2959111	3
combo_004-production_cls_combo_002-qwen2vl_base-mistral_highcap	59.7142857	0.2296746	1.3111141	PARETO	0.3011297	4
combo_004-production_cls_combo_002-llavanext_base-mistral_highcap	53.5714286	0.2335128	1.2971156	PARETO	0.3028931	5

Configuration S2 (combo_004 + Swin + Qwen2-VL-base + Mistral-7B-highcap, STD×DU=0.2900) was identified through sequential filtering by 5 criteria (C1: CRI≤P25, C2: Pareto 3D, C3: DU≤1.553, C4: STD≤0.257, C5: min(STD×DU)). The demonstrated equivalence between sequential filtering and the intersection of the 5 criteria is a non-trivial mathematical property that confirms the internal consistency of the protocol [29,36]. Criterion C5=min(STD×DU) acts as a point-selector argmin and not as a threshold filter, an essential condition to guarantee that it is always the final protocol step [56]. The presence of Mistral-7B-highcap in both S1 and S2 demonstrates that this component is the invariantly optimal choice in F4 regardless of the selection criterion used [29].

6. Discussion

6.1. Effectiveness of the Multimodal Framework as an Integrated Forensic Pipeline

The results obtained demonstrate that the proposed Multimodal Framework surpasses the limitations of accident analysis approaches based on isolated stages. The sequential integration of F1–F4 allows each phase to enrich the context available to the next: the YOLOv11s detector (F1) provides object localization and classification that guides scene understanding in F2; the scene classifier (F2)

 doi:10.20944/preprints202606.0966.v1

adds global semantic context that informs visual description generation in F3; and the VLM (F3) produces a structured description that serves as input to the forensic LLM (F4). This cascaded architecture is consistent with the hierarchical processing paradigm proposed by Liu et al. [4] for multimodal VLM systems, who argue that visual comprehension quality benefits significantly from progressive contextualization.

The dominance of Phase F3 (VLM) as the determining factor of final report quality — evidenced by inter-model BERTScore variability (mean range: 0.406–0.657 for the VLM×LLM cross-tabulation) compared to inter-LLM variability (range: 0.449–0.579) — is a finding with direct methodological implications. When the VLM generates rich and precise descriptions, the LLM can synthesize high-quality reports regardless of its specific architecture. This hierarchy of determinism (F3 > F4) implies that future pipeline improvement investments should prioritize the VLM module over the LLM module.

The paradoxical behavior of LLaVA-NeXT-highcap, whose BERTScore is lower than its base version (0.836 vs. 0.847), suggests domain-specific overfitting during the fine-tuning process. This result, consistent with the warnings of Hu et al. [7] regarding the risks of LoRA fine-tuning on small datasets, highlights the importance of cross-validating performance between base and fine-tuned variants before selecting the definitive component of a forensic pipeline.

6.2. The MODM-MCDM Protocol as a Generalizable Methodological Contribution

The convergence of alternative selection criteria on the same configuration S1 validates the method-invariance principle established by Paradowski et al. [10]: when multiple independent MCDM methods converge on the same solution, its optimality is not an artifact of the chosen method but a structural property of the search space. This convergence is methodologically more informative than any individual performance metric, as it demonstrates that S1 is genuinely optimal and not a local optimum biased by the chosen aggregation function.

The S1/S2 duality — performance vs. methodological robustness — responds to a common practical need in deploying AI systems in institutional settings. Road safety agencies prioritizing maximum descriptive quality (such as generating forensic expert reports for judicial proceedings) benefit from S1. Environments requiring stability under varying selection criteria or methodological audit — such as AI algorithm certification systems in critical infrastructure — benefit from S2. This operational distinction between performance and robustness configurations is consistent with the principles of ISO/IEC 25010 [57], which distinguishes between Performance Efficiency and Reliability as independent quality attributes.

The proposed protocol is also generalizable to other configuration selection domains in multimodal AI pipelines, as its five layers are independent of the specific application domain. The structural condition that C5=Min(STD×DU) must be applied as a point-selector rather than a threshold filter — empirically verified in the present study — constitutes a methodological recommendation directly applicable to any multi-criteria selection protocol in AI.

7. Conclusions

The present study demonstrated the effectiveness of a four-phase Intelligent Multimodal Framework for the automated analysis of traffic accidents, supported by a Hybrid MODM-MCDM Protocol for systematic configuration selection. The exhaustive evaluation of 320 pipeline combinations on a specialized multimodal dataset constructed ad hoc produced two configurations ready for operational deployment: S1 for maximum descriptive performance environments and S2 for audit environments with methodological robustness requirements.

7.1. Main Contributions

- Construction of the specialized multimodal dataset: A vehicular accident analysis dataset was constructed integrating heterogeneous sources (KITTI, Kaggle/Roboflow, ShareGPT4V-COCO-

 doi:10.20944/preprints202606.0966.v1

2017, and Stanford Alpaca) with a phase-differentiated semi-automatic labeling scheme combining model-assisted automatic inference (F1), manually verified labeling (F2), and LLM-assisted synthetic labeling with human validation (F3/F4). The dataset is publicly available on GitHub for reproducibility and community extension.

- Development and evaluation of the F1→F4 Multimodal Framework: An integrated four-phase pipeline was developed: detection (YOLOv11s, mAP@50=0.625), classification (Swin Transformer, F1=0.900), visual comprehension (Qwen2-VL-highcap, BERTScore=0.9192), and forensic generation (Mistral-7B-highcap, BERTScore=0.9172), with independent phase-by-phase fine-tuning via LoRA adapters, demonstrating that the VLM phase (F3) is the determining factor of final report quality.
- Implementation of the Hybrid MODM-MCDM Protocol: A five-layer protocol was implemented with seven MCDM methods (WSum, WProd, Tcheby, ASF, TOPSIS, ELECTRE III, PROMETHEE II) and AHP weighting over 49 normalized criteria, identifying S1 (combo_002 + Swin + Qwen2-VL-highcap + Mistral-7B-highcap, WSUM=0.7993, CRI=1) as the maximum-performance configuration.
- Identification of the maximum robustness configuration S2: S2 (combo_004 + Swin + Qwen2-VL-base + Mistral-7B-highcap, STD×DU=0.2900) was identified through sequential multidimensional filtering criteria (C1: CRI, C2: Pareto 3D, C3: utopian distance, C4: internal dispersion, C5: STD×DU point-selector), establishing a robust selection methodology applicable to other multimodal AI domains.

Author Contributions: Conceptualization, L.B.; Formal Analysis, L.B. and C.R.; Investigation, L.B.; Methodology, L.B. and C.R.; Software, L.B. and P.H.; Validation, L.B., C.A. and J.V.; Writing—Original Draft, L.B.; Writing—Review and Editing, L.B. and C.A.L.; Supervision, L.B., Z.H. and V.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data are available within this article.

Acknowledgments: During the preparation of this manuscript, the authors used Claude Sonnet 4.6 to refine the text. The authors have reviewed and edited the final version and assume full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AHP	Analytic Hierarchy Process
CRI	Consensus Rank Index
MAD	Mean Absolute Deviation
CV	Coefficient of Variation
STD	Standard Deviation
DU L2	Utopian Euclidean Distance
VLM	Vision-Language Model
LLM	Large Language Model
YOLO	You Only Look Once
IoU	Intersection over Union
FPS	Frames per second
MCDM	Multi-Criteria Decision Makin
MODM	Multi-Objective Decision Making
LORA	Low-Rank Adaptation
VIT	Vision Transformer
mAP	mean Average Precision
CNN	Convolutional Neural Network

 doi:10.20944/preprints202606.0966.v1

References

1. World Health Organization. Global Status Report on Road Safety 2023; WHO: Geneva, Switzerland, 2023. Available online: https://www.who.int/publications/i/item/9789240086517 (accessed on 30 May 2026).
2. Batty, M.; Axhausen, K.W.; Giannotti, F.; Pozdnoukhov, A.; Bazzani, A.; Wachowicz, M.; Ouzounis, G.; Portugali, Y. Smart cities of the future. Eur. Phys. J. Spec. Top. 2012, 214, 481–518. https://doi.org/10.1140/epjst/e2012-01703-3
3. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO (Version 11.0.0) [Software]; Ultralytics, 2024. Available online: https://github.com/ultralytics/ultralytics (accessed on 30 May 2026).
4. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual instruction tuning. Adv. Neural Inf. Process. Syst. 2023, 36, 34892–34916.
5. Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. arXiv 2024, arXiv:2409.12191.
6. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. arXiv 2023, arXiv:2310.06825.
7. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-rank adaptation of large language models. In Proceedings of the International Conference on Learning Representations (ICLR 2022), Virtual, 25–29 April 2022.
8. Jaradat, S.; Nayak, R.; Paz, A.; Ashqar, H.I.; Elhenawy, M. Multitask learning for crash analysis: A fine-tuned LLM framework using Twitter data. Smart Cities 2024, 7, 2422–2465. https://doi.org/10.3390/smartcities7050095
9. Boesch, G. Vehicle detection with YOLO: A practical guide. viso.ai 2023. Available online: https://viso.ai/deep-learning/yolov8-guide/ (accessed on 30 May 2026).
10. Paradowski, M.; Więckowski, J.; Sałabun, W. COMET-based verification method for multi-criteria ranking stability. Artif. Intell. Rev. 2025, 58, 298. https://doi.org/10.1007/s10462-025-11098-w
11. Yao, Y.; Wang, X.; Xu, M.; Tu, Z.; Shao, Y.; Lv, Y. Unsupervised traffic accident detection in first-person videos. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 273–280.
12. Basso, F.; Basso, R.; Bravo, F.; Pezoa, R. Real-time detection of parked vehicles using computer vision: An application in the automated road accident reconstruction. Sensors 2021, 21, 3508. https://doi.org/10.3390/s21103508
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
14. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
15. Liu, H.; Li, C.; Li, Y.; Lee, Y.J. Improved baselines with visual instruction tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2024), Seattle, WA, USA, 17–21 June 2024. arXiv:2310.03744.
16. Chen, X.; Fang, H.; Lin, T.Y.; Vedantam, R.; Gupta, S.; Dollár, P.; Zitnick, C.L. Microsoft COCO captions: Data collection and evaluation server. arXiv 2015, arXiv:1504.00325.
17. Lin, C. ShareGPT4V: Improving large multi-modal models with better captions. arXiv 2024, arXiv:2311.12793.
18. Hui, B.; Yang, J.; Zhou, Z.; Dong, J.; Huang, L.; Liu, X.; Zhang, L.; Lu, B.; Zhang, J.; Liu, F.; et al. Qwen2.5-Coder technical report. arXiv 2024, arXiv:2409.12186.
19. Hwang, C.-L.; Yoon, K. Multiple Attribute Decision Making: Methods and Applications — A State-of-the-Art Survey, Lecture Notes in Economics and Mathematical Systems, Vol. 186; Springer: Berlin/Heidelberg, Germany, 1981. https://doi.org/10.1007/978-3-642-48318-9
20. Roy, B. The outranking approach and the foundations of ELECTRE methods. Theory Decis. 1991, 31, 49–73. https://doi.org/10.1007/BF00134132

 doi:10.20944/preprints202606.0966.v1

21. Brans, J.P.; Vincke, P. A preference ranking organisation method: The PROMETHEE method for multiple criteria decision-making. Manag. Sci. 1985, 31, 647–656. https://doi.org/10.1287/mnsc.31.6.647
22. Saaty, T.L. The Analytic Hierarchy Process; McGraw-Hill: New York, NY, USA, 1980.
23. Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; Hashimoto, T.B. Stanford Alpaca: An Instruction-Following LLaMA Model [Software]; GitHub, 2023. Available online: https://github.com/tatsu-lab/stanford_alpaca (accessed on 30 May 2026).
24. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. arXiv 2024, arXiv:2410.17725.
25. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. A comparative analysis of object detection metrics with a companion open-source toolkit. Electronics 2021, 10, 279. https://doi.org/10.3390/electronics10030279
26. Figueira, J.; Greco, S.; Ehrgott, M. (Eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, 3rd ed.; Springer: New York, NY, USA, 2022. https://doi.org/10.1007/978-1-0716-0987-2
27. Nabavi, S.R.; Wang, Z.; Rodríguez, M.L. Sensitivity analysis of multi-criteria decision-making methods for engineering applications. Ind. Eng. Chem. Res. 2023, 62, 18761–18779. https://doi.org/10.1021/acs.iecr.2c04270
28. Diéguez-Antón, A.; Rodríguez-Resendes, Y.; Martín-Morales, G. Elite multi-criteria decision making — Pareto front optimization in multi-objective optimization. Algorithms 2024, 17, 206. https://doi.org/10.3390/a17050206
29. Paradowski, B.; Wątróbski, J.; Sałabun, W. Novel coefficients for improved robustness in multi-criteria decision analysis. Artif. Intell. Rev. 2025, 58, 298. https://doi.org/10.1007/s10462-025-11307-6
30. Saaty, T.L. How to make a decision: The analytic hierarchy process. Eur. J. Oper. Res. 1990, 48, 9–26. https://doi.org/10.1016/0377-2217(90)90057-I
31. Hwang, C.L.; Yoon, K. Multiple Attribute Decision Making: Methods and Applications, Lecture Notes in Economics and Mathematical Systems, Vol. 186; Springer: Berlin/Heidelberg, Germany, 1981. https://doi.org/10.1007/978-3-642-48318-9
32. Miettinen, K. Nonlinear Multiobjective Optimization; Springer: Boston, MA, USA, 1999. https://doi.org/10.1007/978-1-4615-5563-6
33. Figueira, J.R.; Greco, S.; Roy, B.; Słowiński, R. ELECTRE methods: Main features and recent developments. In Handbook of Operations Research and Management Science in Finance; Zopounidis, C., Doumpos, M., Eds.; Springer: Cham, Switzerland, 2023; pp. 51–89. https://doi.org/10.1007/978-3-031-38833-0_3
34. Brans, J.P.; De Smet, Y. PROMETHEE methods. In Multiple Criteria Decision Analysis: State of the Art Surveys, 2nd ed.; Greco, S., Ehrgott, M., Figueira, J., Eds.; Springer: New York, NY, USA, 2016; pp. 187–220. https://doi.org/10.1007/978-1-4939-3094-4_6
35. Kizielewicz, B.; Shekhovtsov, A.; Sałabun, W. pymcdm — The universal library for solving multi-criteria decision-making problems. SoftwareX 2023, 22, 101368. https://doi.org/10.1016/j.softx.2023.101368
36. Więckowski, J.; Kizielewicz, B.; Paradowski, B.; Sałabun, W. A robust framework for benchmarking MCDA methods: Sensitivity, stability, and method selection decision support. Appl. Soft Comput. 2026, 187, 114335. https://doi.org/10.1016/j.asoc.2025.114335
37. Siddiqui, O.; Bhutta, M.K.S. Clustering analysis for the Pareto optimal front in multi-objective optimization. Computation 2022, 10, 37. https://doi.org/10.3390/computation10030037
38. Wierzbicki, A.P. The use of reference objectives in multiobjective optimization. In Multiple Criteria Decision Making, Theory and Applications, Lecture Notes in Economics and Mathematical Systems, Vol. 177; Fandel, G., Gal, T., Eds.; Springer: Berlin/Heidelberg, Germany, 1980; pp. 468–486. https://doi.org/10.1007/978-3-642-48782-8_32
39. Shorten, C.; Khoshgoftaar, T.M.; Furht, B. Text data augmentation for deep learning. J. Big Data 2021, 8, 101. https://doi.org/10.1186/s40537-021-00492-0
40. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive study of transfer learning for deep neural networks. IEEE Trans. Neural Netw. Learn. Syst. 2021, 32, 5338–5359. https://doi.org/10.1109/TNNLS.2020.3004526

 doi:10.20944/preprints202606.0966.v1

41. Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. Adv. Neural Inf. Process. Syst. 2023, 36, 10088–10115. arXiv:2305.14314.
42. Li, S.; Zhao, Z.; Varma, R.; Salpekar, O.; Noordhuis, P.; Li, T.; Paszke, A.; Smith, J.; Vaughan, B.; Damania, P.; et al. PyTorch distributed: Experiences on accelerating data parallel training. Proc. VLDB Endow. 2020, 13, 3005–3018. https://doi.org/10.14778/3415478.3415530
43. Chen, S.; Hou, Y.; Cui, Y.; Che, W.; Liu, T.; Yu, X. Revisiting parameter-efficient fine-tuning: A survey. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024), Bangkok, Thailand, 11–16 August 2024. arXiv:2308.16972.
44. RunPod, Inc. RunPod: GPU Cloud Platform for AI Workloads [Software]; RunPod, Inc., 2024. Available online: https://www.runpod.io (accessed on 30 May 2026).
45. NVIDIA Corporation. NVIDIA RTX A4500 Product Overview; NVIDIA Corporation: Santa Clara, CA, USA, 2023. Available online: https://www.nvidia.com/en-us/design-visualization/rtx-a4500/ (accessed on 30 May 2026).
46. Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; et al. Mixed precision training. arXiv 2018, arXiv:1710.03740.
47. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations, Online, 16–20 November 2020; pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6
48. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.
49. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721
50. Steiner, A.; Kolesnikov, A.; Zhai, X.; Wightman, R.; Uszkoreit, J.; Beyer, L. How to train your ViT? Data, augmentation, and regularization in vision transformers. Trans. Mach. Learn. Res. 2022. arXiv:2106.10270.
51. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019. arXiv:1711.05101.
52. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 11976–11986. https://doi.org/10.1109/CVPR52688.2022.01167
53. Müller, R.; Kornblith, S.; Hinton, G. When does label smoothing help? Adv. Neural Inf. Process. Syst. 2019, 32. arXiv:1906.02629.
54. Celikyilmaz, A.; Clark, E.; Gao, J. Evaluation of text generation: A survey. arXiv 2021, arXiv:2006.14799.
55. Freitag, M.; Foster, G.; Grangier, D.; Ratnakar, V.; Tan, Q.; Macherey, W. Experts, errors, and context: A large-scale study of human evaluation for machine translation. Trans. Assoc. Comput. Linguist. 2021, 9, 1460–1474. https://doi.org/10.1162/tacl_a_00437
56. Rangaiah, G.P.; Feng, Z.; Hoadley, A.F. Application and analysis of methods for selecting an optimal solution from the Pareto-optimal front. Ind. Eng. Chem. Res. 2021, 60, 11216–11230. https://doi.org/10.1021/acs.iecr.1c01413
57. ISO/IEC. ISO/IEC 25010:2011 — Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation (SQuaRE) — System and Software Quality Models; International Organization for Standardization: Geneva, Switzerland, 2011.