# Strategic Blueprint for Zalo AI Challenge 2025: The RoadBuddy Autonomous Assistant

## 1. Executive Summary

The Zalo AI Challenge 2025, specifically the RoadBuddy track, represents a watershed moment in the intersection of computer vision, legal reasoning, and edge-constrained artificial intelligence within the Vietnamese technological landscape. The mandate is clear yet technically formidable: engineer a driving assistant capable of ingesting high-dimensional dashcam video data (5 to 15 seconds in duration), processing it through a neuro-symbolic reasoning pipeline, and delivering legally grounded answers to traffic inquiries strictly adhering to the Vietnamese legal framework. This must be achieved within a stringent inference window of 30 seconds on consumer-grade hardware—specifically a single NVIDIA RTX 3090 or A30 GPU—while adhering to a model parameter ceiling of 9 billion.

This report articulates a comprehensive, research-backed strategy to secure a winning position in this competition. Our analysis suggests that a monolithic approach—relying on a single, end-to-end Vision-Language Model (VLM)—is fundamentally ill-suited to the dual pressures of high-fidelity perception (detecting small traffic signs) and precise legal citation (avoiding hallucination of Decree 100/2019 or Law 36/2024). Standard VLMs, when constrained to 9B parameters, typically lack the visual resolution to discern distant signage in wide-angle footage and the long-context retention required to memorize the entirety of the Vietnam Road Traffic Order and Safety Law.

Consequently, we propose the **"Lenses & Law" Architecture**: a hybrid, modular-augmented pipeline that decouples perception from reasoning. This system synthesizes a lightweight, high-recall detection front-end (YOLOv11-Nano) with a retrieval-augmented generation (RAG) core utilizing **Qwen2.5-VL-7B-Instruct** or **MiniCPM-V 2.6** as the executive reasoning engine. By treating traffic sign recognition as a retrieval task rather than a generation task, we dramatically reduce the cognitive load on the VLM, allowing it to focus on complex causal reasoning and legal interpretation. Furthermore, to satisfy the 30-second latency constraint, we introduce a dynamic keyframe selection mechanism driven by optical flow analysis, rejecting redundant temporal data to maximize the information density of the tokens fed into the VLM.

The following sections detail the theoretical underpinnings, architectural decisions, and engineering optimizations required to realize this vision. We draw extensively from recent benchmarks in multimodal learning, advancements in quantization (AWQ), and the specific legal texts governing Vietnamese road traffic to construct a robust, hallucination-resistant, and high-performance system.

## 2. Operational Context and Constraint Decomposition

Success in the RoadBuddy challenge requires a rigorous deconstruction of the operational environment. The constraints imposed—time, memory, and model size—are not merely hurdles but defining features that dictate the architectural topology. We must analyze the hardware capabilities of the RTX 3090/A30 against the computational demands of processing video data and generating text.

### 2.1 Hardware Envelope: The RTX 3090 and A30

The target deployment environment is specified as a single NVIDIA RTX 3090 or A30 GPU. Understanding the microarchitecture of these cards is essential for optimizing the inference pipeline. The RTX 3090 is built on the Ampere architecture, featuring 24 GB of GDDR6X memory with a massive bandwidth of 936 GB/s.1 This high bandwidth is a critical asset for memory-bound applications like Large Language Model (LLM) inference, where the bottleneck is often moving weights from memory to compute units rather than the computation itself.

However, the 24 GB VRAM limit poses a significant challenge for a 9B parameter model processing video. A standard 7B model in half-precision (FP16) consumes approximately 14-15 GB of VRAM for weights alone. This leaves less than 10 GB for the KV cache (which grows linearly with context length), the visual encoder's activation maps (which are heavy for video), and any auxiliary models like the object detector or vector database. Without quantization, the system would likely encounter Out-Of-Memory (OOM) errors when processing the 128K context window capabilities of modern VLMs or handling multi-frame video inputs.1

The A30, alternatively, offers similar memory capacity (24 GB HBM2) but with significantly higher memory bandwidth (933 GB/s) and distinct Tensor Core optimizations for inference. While the A30 is a data center card optimized for throughput, the RTX 3090 is a consumer card with raw power. Our strategy must be agnostic to the subtle differences but optimized for the shared 24 GB VRAM constraint.

### 2.2 The 30-Second Inference Budget

The 30-second latency limit is the most stringent constraint. In the context of video QA, latency is composed of four distinct phases:

1. **Video Decoding & Preprocessing:** Converting compressed MP4/H.264 data into raw tensor frames. On a CPU, decoding a 15-second 1080p video can take 1–3 seconds.
2. **Visual Encoding:** Passing frames through the Vision Transformer (ViT). For a model like Qwen2-VL, which uses a deep ViT (e.g., SigLIP or similar), encoding a single image can take 50–200ms. Encoding 16 frames sequentially could consume 3–4 seconds.
3. **Retrieval (RAG):** Querying the vector database. With an optimized index (e.g., HNSW), this is negligible (<100ms).
4. **Autoregressive Generation:** The LLM generates the answer token by token. A 7B model on an RTX 3090 can generate 40–60 tokens per second.3 A detailed legal answer might require 300 tokens, consuming 5–8 seconds.

Summing these up ($3s + 4s + 0.1s + 8s \approx 15.1s$), we theoretically fit within the 30s limit. However, this assumes an optimized pipeline. Naive implementations using Hugging Face transformers defaults often result in 2x-3x slower performance due to lack of continuous batching and unoptimized attention kernels. Therefore, the use of a high-performance serving engine like **vLLM** is non-negotiable.4

### 2.3 The 9B Parameter Ceiling

The limitation to 9 billion parameters places the solution in the realm of Small Language Models (SLMs). While 70B+ models like GPT-4 or Qwen-72B exhibit emergent reasoning and vast world knowledge, 7B models often struggle with complex, multi-step reasoning and "long-tail" knowledge recall.

In the context of Vietnamese traffic law, a 7B model pre-trained primarily on English or Chinese web data will likely hallucinate when asked about specific articles of Law 36/2024. It may conflate Vietnamese rules with US traffic laws (e.g., "Right turn on red," which is generally prohibited in Vietnam unless signaled). This necessitates a strategy where the model is not the *source* of legal knowledge but the *processor* of retrieved legal context. We must shift the burden of knowledge from the model's weights (parametric memory) to the input context window (non-parametric memory via RAG).

## 3. Legal Knowledge Engineering and RAG Strategy

The core requirement of the RoadBuddy challenge is to answer questions using **Vietnamese law**. This is not a general driving advice task; it is a legal compliance task. The system must cite specific regulatory documents. We must rigorously analyze the target legal corpus to structure our Retrieval-Augmented Generation (RAG) system effectively.

### 3.1 Analysis of the Legal Corpus

The legal framework for Vietnamese road traffic has recently undergone a significant overhaul. The challenge explicitly references two key documents:

1. **Law No. 36/2024/QH15 on Road Traffic Order and Safety:** Passed by the National Assembly on June 27, 2024, and effective from January 1, 2025.6 This law separates traffic rules from infrastructure management (which remains in the Road Law).
   * *Key Changes:* It enforces a **zero-tolerance alcohol ban** (Article 9, Clause 2), strictly prohibiting driving with *any* alcohol concentration in blood or breath.7 This is a critical distinction from older laws or international standards that might allow small amounts. A model trained on data prior to mid-2024 might hallucinate the old 50mg/100ml limit unless corrected by RAG.
   * *Points System:* It introduces a license point deduction system (12 points total). The system must be aware of which violations trigger point deductions versus simple fines.6
   * *Child Safety:* New rules for children under 10 years old and under 1.35m height, prohibiting them from sitting in the front seat.8
2. **QCVN 41:2024/BGTVT (National Technical Regulation on Road Signs):** This technical regulation defines the visual appearance and legal meaning of every traffic sign and road marking in Vietnam.9
   * *Complexity:* It contains hundreds of signs (Prohibitory P.x, Command R.x, Warning W.x). A generic VLM will identify a "red circle" but may not distinguish between "P.103a" (No cars) and "P.103b" (No right turn for cars). The distinction is purely visual but legally vast.

### 3.2 Structuring the Legal Vector Database

To enable the VLM to use this knowledge, we cannot simply "dump" the PDF text into the prompt context limit; doing so would dilute attention and exceed token limits. We must structure a specialized **Legal Vector Database**.

#### 3.2.1 Hybrid Indexing Strategy

We propose a dual-index approach using a vector database like **Qdrant** or **ChromaDB**, which supports multimodal payloads.

**Index A: The Statutory Text Index (Law 36/2024)**

* **Chunking Strategy:** We will chunk the law by **Article**. Each chunk will contain the full text of the Article, its Title, and the Chapter it belongs to.
* **Metadata:** {'article\_id': '9', 'chapter': 'General Rules', 'keywords': ['alcohol', 'drunk', 'ban']}.
* **Embedding Model:** We require a multilingual embedding model that excels in Vietnamese legal semantic retrieval. **bge-m3** (BAAI/bge-m3) is the current SOTA for multilingual retrieval, supporting long contexts (8192 tokens) and dense/sparse hybrid retrieval.11
* **Sparse Retrieval:** We will also maintain a BM25 (keyword) index. If a user asks about "Article 9," dense vector search might drift to semantically similar articles, whereas BM25 will lock onto the keyword "Article 9" precisely.

**Index B: The Visual Sign Registry (QCVN 41)**

* **Content:** This index contains the standard reference images of all traffic signs defined in QCVN 41.
* **Embedding:** We use **CLIP (ViT-L/14)** or **SigLIP** to generate embeddings for every reference sign image.12
* **Workflow:** When the system detects a sign in the dashcam video, it crops the sign, computes its visual embedding, and queries this Visual Sign Registry. The result is the exact sign code (e.g., "P.102") and its legal meaning, which is then fed as text to the VLM. This bridges the gap between pixel data and legal text.

### 3.3 Prompt Engineering for Legal Compliance

The retrieved legal text must be injected into the VLM using a strict system prompt to enforce compliance.

* **System Prompt Template:**"You are a legal expert in Vietnamese Road Traffic Law. You are provided with a video analysis and a set of retrieved legal articles from Law 36/2024/QH15 and QCVN 41:2024. Your task is to answer the user's question. You must cite the specific Article and Clause that supports your answer. If the video shows a violation, identify it strictly according to the retrieved laws. Do not use external knowledge if it conflicts with the provided context."

This formulation leverages "In-Context Learning" to align the model's output with the specific constraints of the 2025 challenge, mitigating the risk of hallucinating outdated 2008 laws.

## 4. Visual Perception Strategy: The "Lenses"

A driving assistant is only as good as its eyes. The Zalo challenge involves wide-angle dashcam footage where critical details—traffic signs, lane markings, and traffic light states—are often small and fleeting. Relying solely on the VLM's native encoder to process the entire video frames is inefficient and perceptually inadequate.

### 4.1 The "Small Object" Dilemma

Standard VLMs typically resize input images to a fixed resolution, often $336 \times 336$ or $448 \times 448$ pixels. In a $1920 \times 1080$ dashcam video, a traffic sign might be $50 \times 50$ pixels. When downsampled to a $448 \times 448$ grid, this sign becomes a $10 \times 10$ feature map blob, effectively destroying the text or symbol required for identification.13 To read a "Max Speed 60" sign, the model needs high-resolution access to that specific region.

### 4.2 The Modular Perception Pipeline

We propose a **Coarse-to-Fine Perception Module** that acts as a pre-processor before the VLM.

#### 4.2.1 Object Detection with YOLOv11

We will deploy a lightweight object detection model, **YOLOv11-Nano** or **YOLOv10-Small**, trained specifically on Vietnamese traffic datasets (VNTS).

* **Why YOLOv11?** It offers the best trade-off between latency and accuracy. The Nano version runs in <10ms on an RTX 3090.13
* **Target Classes:** We do not need to classify every sign perfectly at this stage. We only need to *detect* them. We define a simplified class ontology: Prohibitory, Warning, Command, Information, Traffic Light, Road Marking.
* **Objective:** High Recall. It is acceptable to propose a false positive (which the VLM will later discard) but fatal to miss a true positive.

#### 4.2.2 The Crop-and-Zoom Mechanism

Upon detecting a traffic sign with the YOLO model, the system performs a "virtual zoom":

1. **Coordinate Extraction:** Get the bounding box $(x, y, w, h)$ from the high-resolution source frame (1080p).
2. **Context Expansion:** Expand the box by 10-20% to capture context (e.g., the pole or a supplementary sign below it).
3. **Crop & Stack:** Crop these regions. If multiple signs are present, stitch them into a single "Sign Summary Image."
4. **VLM Input:** The VLM receives two visual inputs:
   * **Context Stream:** A sequence of downsampled full frames (for understanding vehicle motion, lane position).
   * **Detail Stream:** The high-resolution "Sign Summary Image" (for reading specific text and symbols).

This strategy allows a 7B model to achieve OCR performance comparable to a 72B model, as it is "looking" at the critical pixels with full fidelity.

### 4.3 Dynamic Keyframe Selection

Processing video at 30 fps is redundant. We need to select the *most informative* frames to stay within the 30s budget. Uniform sampling (e.g., every 30th frame) risks missing a sign that is visible for only 0.5 seconds.

* **Algorithm:** We implement a **Semantic Triggered Sampling** method.
  + *Base Rate:* Sample 1 frame per second (FPS) for general context.
  + *Trigger:* If the YOLO detector's confidence for a traffic\_sign class peaks or if the bounding box area exceeds a threshold (indicating the car is close to the sign), we forcefully sample that frame.
  + *Deduplication:* Use optical flow or histogram comparison to reject frames that are $>95\%$ similar to the previous keyframe (e.g., stopped at a red light).15

## 5. Vision-Language Model Architecture

The reasoning engine—the "Brain"—must synthesize the visual cues from the perception module and the legal facts from the RAG module to answer the user's question.

### 5.1 Comparative Model Selection

We evaluated three primary candidates available in the <9B parameter class as of late 2024/early 2025.

#### 5.1.1 Qwen2.5-VL-7B-Instruct

* **Architecture:** Qwen2.5-7B LLM + Naive Dynamic Resolution Vision Encoder.16
* **Strengths:**
  + **Naive Dynamic Resolution:** It handles images of varying aspect ratios natively, mapping them to dynamic token counts. This is ideal for our "Sign Summary Image" which might be a tall strip of stacked signs.
  + **Strong OCR:** Benchmarks on DocVQA and MathVista show it outperforms competitors in reading text.17 This is vital for reading auxiliary traffic plates (e.g., "6h-9h").
  + **Long Video Support:** Capable of processing 20min+ videos, implying robust positional embeddings for our 15s clips.
* **Weaknesses:** Video inference throughput in vLLM can be lower than image inputs due to complex attention patterns.4

#### 5.1.2 MiniCPM-V 2.6 (8B)

* **Architecture:** SigLIP-400M + Qwen2-7B.
* **Strengths:**
  + **Token Density:** It encodes high-res images (1.8M pixels) into only 640 tokens.18 This is significantly more efficient than standard ViT encodings, offering a speed advantage for the 30s limit.
  + **Edge Optimization:** Designed for end-side devices (iPad, etc.), suggesting low VRAM footprint.
* **Weaknesses:** While fast, its reasoning depth on complex causal traffic scenarios (e.g., "Who has the right of way in this roundabout?") might lag behind the specialized Qwen2.5-VL.

#### 5.1.3 InternVL2-8B

* **Architecture:** InternViT-300M + InternLM2.5-7B.
* **Strengths:** SOTA on generic video benchmarks (Video-MME).19
* **Weaknesses:** The visual encoder is heavy (300M params vs. smaller alternatives), potentially impacting latency.

### 5.2 Selection: Qwen2.5-VL-7B

We select **Qwen2.5-VL-7B-Instruct** as the primary candidate. The decisive factor is its **OCR capability** and **Dynamic Resolution**. In traffic sign analysis, reading the text on a sign is often the difference between a correct and incorrect legal interpretation. The potential latency disadvantage can be mitigated via quantization (AWQ) and the efficient keyframe strategy outlined in Section 4.3.

## 6. Data Strategy and Curation

Data is the fuel for the AI engine. We must construct a robust training pipeline that bridges the gap between generic visual pre-training and specific Vietnamese legal reasoning.

### 6.1 Dataset Aggregation

We will leverage a hierarchy of datasets to train the Perception Module (YOLO) and fine-tune the VLM.

#### 6.1.1 The VLSP 2025 MLQA-TSR Dataset

This is the gold standard for the challenge. It contains multimodal legal question-answering pairs specifically for Vietnamese traffic signs.9

* **Composition:** Images of traffic signs + Textual Questions + Legal Articles + Correct Answers.
* **Utility:** This will be the primary validation set and the core of the instruction-tuning dataset for the VLM.

#### 6.1.2 VNTS (Vietnam Road Signs Dataset)

A comprehensive dataset for object detection.20

* **Size:** ~3,200 images.
* **Classes:** Detailed breakdown of classes (e.g., Speed limit (50km/h), No Parking, Pedestrian Crossing).
* **Utility:** Training the YOLOv11 detector. We will map the specific classes (e.g., Speed limit 50) to broader super-classes (Prohibitory) for the detector, leaving the fine-grained reading to the VLM.

#### 6.1.3 Zalo Traffic Sign Detection 2020

A legacy dataset from a previous challenge.21

* **Size:** 4,500 images.
* **Features:** Contains challenging conditions (night, rain, occlusion).
* **Utility:** Augmenting the YOLO training set to improve robustness against environmental noise.

### 6.2 Synthetic Data Generation (The Data Flywheel)

To teach the VLM "reasoning" rather than just "recognition," we need data that explains the *why*. We will generate this synthetically.

* **Source Material:** Unlabeled dashcam videos from Vietnamese sources (YouTube channels, open datasets).
* **The "Teacher":** We use a powerful proprietary model (e.g., **GPT-4o** or **Qwen-Max**) to annotate these videos.
  + *Prompt:* "Analyze this video frame-by-frame. Identify every traffic sign. Explain its meaning under Vietnam Law 36/2024. Formulate a difficult multiple-choice question about the driver's obligation in this scene and provide the correct answer with legal citation."
* **Distillation:** This process generates thousands of high-quality (Video, Question, Answer, Chain-of-Thought) tuples. We use this synthetic dataset to fine-tune the Qwen2.5-VL-7B model, effectively distilling the reasoning capabilities of the large model into our constrained 7B architecture.

## 7. Systems Engineering and Optimization

Meeting the 30-second inference limit on a single GPU requires optimization at the kernel and serving level. We cannot rely on standard Python inference loops.

### 7.1 Inference Engine: vLLM

We will utilize **vLLM** (Virtual Large Language Model) as the serving backend. vLLM is the industry standard for high-throughput inference due to its **PagedAttention** algorithm, which optimizes memory management for the KV cache.5

* **Video Handling in vLLM:** Recent updates to vLLM have added support for multi-modal inputs, including Qwen2-VL. However, processing video tokens can be slow if not configured correctly.
* **Configuration:**  
  Bash  
  vllm serve Qwen/Qwen2.5-VL-7B-Instruct-AWQ \  
   --quantization awq \  
   --dtype half \  
   --limit-mm-per-prompt video=1,image=4 \  
   --max-model-len 8192 \  
   --gpu-memory-utilization 0.9  
  + *Limit MM:* We explicitly limit the number of video chunks and image crops to prevent OOM.
  + *Max Model Len:* Reducing context from 32k to 8k saves significant VRAM for the KV cache, sufficient for 15s videos.

### 7.2 Activation-Aware Weight Quantization (AWQ)

Running the model in FP16 consumes ~15GB VRAM. We will use **INT4 AWQ**.

* **Mechanism:** AWQ identifies the 1% of salient weights that are critical for performance and keeps them in higher precision, while quantizing the rest to 4-bit.
* **Benefit:** This reduces the model size to ~5.5 GB.
* **Performance:** On the RTX 3090, INT4 inference is significantly faster than FP16 due to higher effective memory bandwidth usage (moving 4-bit weights is 4x faster than 16-bit).2 This serves to double the token generation speed, crucial for the autoregressive generation phase.

### 7.3 Docker Containerization

The solution must be deployed as a Docker container.

* **Base Image:** vllm/vllm-openai:latest or a custom build with specific CUDA 12.1 optimizations.
* **Dependency Management:** All models (YOLO, Qwen-AWQ, Embedding models) must be baked into the Docker image to ensure offline functionality. We will use a multi-stage build to keep the image size manageable, although the model weights alone will make it large (~10GB).
* **Runtime Command:**  
  Bash  
  docker run --gpus all --shm-size=12g -v /path/to/models:/models roadbuddy-submission  
    
  (Note: --shm-size is critical for NCCL/PyTorch communications).

## 8. Experimental Validation Plan

To ensure the strategy is robust, we define a rigorous validation protocol using the metrics specified in the VLSP/Zalo challenges.

### 8.1 Metrics

1. **Retrieval F2-Score:** For the RAG component. We prioritize Recall (F2) over Precision because failing to retrieve the relevant law is a catastrophic failure, whereas retrieving an extra irrelevant law is manageable.11
2. **Accuracy:** For the final QA generation.
3. **Inference Latency:** Measured as the wall-clock time from receiving the video path to returning the JSON response. Must be $< 30.0$ seconds.

### 8.2 Ablation Studies

We will perform the following experiments to tune the system:

* **Keyframe Count vs. Accuracy:** Test 4, 8, 16 keyframes. Hypothesis: 8 frames is the diminishing returns point for traffic QA.
* **Crop vs. No-Crop:** Evaluate the impact of the "Zoom-In" mechanism on OCR accuracy. We expect a >15% accuracy gain on small sign recognition.
* **Quantization Impact:** Compare FP16 vs. AWQ INT4. Hypothesis: Accuracy drop is < 2%, but latency improves by 40%.

## 9. Conclusion

The Zalo AI Challenge 2025 RoadBuddy track is a test of architectural ingenuity under constraint. Our proposed solution moves beyond the simplistic application of a pretrained model. Instead, it engineers a symbiotic system:

1. **Perception:** A dedicated YOLOv11 "eye" that foveates on critical traffic signs, overcoming the resolution limits of standard VLMs.
2. **Knowledge:** A structured Legal RAG system that bridges the gap between pixel data and the rigid statutes of Law 36/2024 and QCVN 41.
3. **Reasoning:** An optimized Qwen2.5-VL-7B engine, accelerated via AWQ and vLLM, to synthesize these inputs into legally compliant answers.

By rigorously optimizing for the 9B parameter and 30-second limitations through quantization, dynamic keyframe selection, and modular design, this strategy provides a definitive roadmap to achieving state-of-the-art performance in the RoadBuddy challenge. The integration of specific Vietnamese datasets (VLSP, VNTS) and the generation of synthetic reasoning traces further ensures that the system is not just a generic AI, but a specialized, compliant Vietnamese driving assistant.

## 10. Data Tables and Implementation References

### Table 1: Model Candidate Comparison (<9B Parameters)

| **Model Name** | **Parameter Count** | **Vision Encoder** | **Video Capability** | **OCR Performance** | **Recommended Use** |
| --- | --- | --- | --- | --- | --- |
| **Qwen2.5-VL-7B** | 7.6B | Naive Dynamic Res | **High** (Long context) | **SOTA** (DocVQA) | **Primary Reasoner** |
| **MiniCPM-V 2.6** | 8B | SigLIP-400M | Moderate | High (Token Efficient) | Fallback / Speed |
| **InternVL2-8B** | 8B | InternViT-300M | High (Video-MME) | Moderate | Research |
| **LLaVA-NeXT-Video** | 7B | CLIP-ViT-L | High | Moderate | Baseline |

### Table 2: Dataset Utilization Strategy

| **Dataset Name** | **Source** | **Size** | **Usage Strategy** |
| --- | --- | --- | --- |
| **VLSP 2025 MLQA-TSR** | VLSP Challenge 9 | ~3,000 QA Pairs | **Validation / Fine-tuning** (Gold Standard) |
| **VNTS** | Kaggle 20 | 3,200 Images | **YOLO Training** (Detection) |
| **Zalo 2020** | Zalo AI 21 | 4,500 Images | **Augmentation** (Robustness) |
| **Law 36/2024 Corpus** | Official Text 8 | 89 Articles | **RAG Knowledge Base** (Text Index) |
| **QCVN 41 Sign Registry** | Official Reg 9 | ~400 Sign Images | **RAG Knowledge Base** (Visual Index) |

### Table 3: Estimated Latency Budget (RTX 3090, Qwen2.5-VL-AWQ)

| **Pipeline Stage** | **Operation** | **Estimated Time** | **Optimization** |
| --- | --- | --- | --- |
| **1. Preprocessing** | Video Decode + Optical Flow | 2.5s | OpenCV CUDA |
| **2. Detection** | YOLOv11 Inference (16 frames) | 0.5s | TensorRT / Batched |
| **3. Encoding** | Visual Encoder (ViT) | 4.0s | vLLM / Pipeline Parallel |
| **4. Retrieval** | Vector Search (Qdrant) | 0.2s | HNSW Index |
| **5. Generation** | LLM Decoding (256 tokens) | 6.5s | AWQ INT4 / vLLM |
| **Overhead** | Python/Docker Overhead | 1.0s | Efficient Code |
| **Total** |  | **~14.7s** | **Well within 30s Limit** |

#### Works cited

1. vLLM Performance Tuning: The Ultimate Guide to xPU Inference Configuration | Google Cloud Blog, accessed January 5, 2026, <https://cloud.google.com/blog/topics/developers-practitioners/vllm-performance-tuning-the-ultimate-guide-to-xpu-inference-configuration>
2. Qwen 2.5 14B Instruct AWQ vLLM - Kaggle, accessed January 5, 2026, <https://www.kaggle.com/code/cemalemrealbayrak/qwen-2-5-14b-instruct-awq-vllm>
3. RTX4090 vLLM Benchmark: Best GPU for LLMs Below 8B on Hugging Face, accessed January 5, 2026, <https://www.databasemart.com/blog/vllm-gpu-benchmark-rtx4090>
4. [Performance]: Multi-Modal Benchmark on NVIDIA A100 – Qwen2.5-VL / MiniCPM-V-4 / InternVL3\_5-4B / InternVL3\_5-2B · Issue #24728 · vllm-project/vllm - GitHub, accessed January 5, 2026, <https://github.com/vllm-project/vllm/issues/24728>
5. vLLM - Qwen, accessed January 5, 2026, <https://qwen.readthedocs.io/en/latest/deployment/vllm.html>
6. Traffic Regulations and Debates: Vietnam's Evolving Road Laws, accessed January 5, 2026, <https://www.thevietnamese.org/2025/03/traffic-regulations-and-debates-vietnams-evolving-road-laws/>
7. CSDLVBQPPL Bộ Tư pháp - Road traffic, accessed January 5, 2026, <https://moj.gov.vn/vbpq/en/lists/vn%20bn%20php%20lut/view_detail.aspx?itemid=10506>
8. Law No. 36/2024/QH15 dated June 27 of 2024 on Road Traffic Order and Safety, accessed January 5, 2026, <https://thuviennhadat.vn/van-ban-phap-luat-viet-nam/law-36-2024-qh15-road-traffic-order-and-safety-620124.html>
9. VLSP 2025 MLQA-TSR Challenge: Vietnamese Multimodal Legal Question Answering on Traffic Sign Regulation - arXiv, accessed January 5, 2026, <https://arxiv.org/html/2510.20381v1>
10. VLSP 2025 MLQA-TSR Challenge: Vietnamese Multimodal Legal Question Answering on Traffic Sign Regulation - ACL Anthology, accessed January 5, 2026, <https://aclanthology.org/2025.vlsp-1.48.pdf>
11. VLSP 2025 MLQA-TSR Challenge: Vietnamese Multimodal Legal Question Answering on Traffic Sign Regulation - ResearchGate, accessed January 5, 2026, <https://www.researchgate.net/publication/396847809_VLSP_2025_MLQA-TSR_Challenge_Vietnamese_Multimodal_Legal_Question_Answering_on_Traffic_Sign_Regulation>
12. GitHub - phungpx/LexiSignVQA: VLSP2025, accessed January 5, 2026, <https://github.com/phungpx/LexiSignVQA>
13. An Enhanced Algorithm for Detecting Small Traffic Signs Using YOLOv10 - MDPI, accessed January 5, 2026, <https://www.mdpi.com/2079-9292/14/5/955>
14. Towards Real-Time Vietnamese Traffic Sign Recognition on Embedded Systems | Request PDF - ResearchGate, accessed January 5, 2026, <https://www.researchgate.net/publication/388017530_Towards_Real-Time_Vietnamese_Traffic_Sign_Recognition_on_Embedded_Systems>
15. A Keyframe Extraction Method for Assembly Line Operation Videos Based on Optical Flow Estimation and ORB Features - MDPI, accessed January 5, 2026, <https://www.mdpi.com/1424-8220/25/9/2677>
16. Qwen/Qwen2.5-VL-7B-Instruct - Hugging Face, accessed January 5, 2026, <https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct>
17. Qwen/Qwen2-VL-72B-Instruct-AWQ - Hugging Face, accessed January 5, 2026, <https://huggingface.co/Qwen/Qwen2-VL-72B-Instruct-AWQ>
18. minicpm-v - Ollama, accessed January 5, 2026, <https://ollama.com/library/minicpm-v>
19. OpenGVLab/InternVL2-8B - Hugging Face, accessed January 5, 2026, <https://huggingface.co/OpenGVLab/InternVL2-8B>
20. Vietnam Traffic Signs - Kaggle, accessed January 5, 2026, <https://www.kaggle.com/datasets/maitam/vietnamese-traffic-signs/versions/4/data>
21. Zalo Traffic Sign - Dataset Ninja, accessed January 5, 2026, <https://datasetninja.com/zalo-traffic-sign>
22. Using Docker - vLLM, accessed January 5, 2026, <https://docs.vllm.ai/en/stable/deployment/docker/>