GeoNLI- A Hybrid, Router-Based Framework for Geometric and Semantic Interpretation of Satellite Imagery

Team ID: Team 82

Abstract: In the field of remote sensing, the split between geometric accuracy and semantic reasoning has made it hard to create truly unified intelligence systems for a long time. Vision-Language Models (VLMs) are great at describing meaning, but not so great in exact Oriented Bounding Box (OBB) regression. Convolutional Neural Networks (CNNs), on the other hand, are great at geometric fidelity but not so great at understanding language. This study introduces GeoNLI, an innovative architecture that addresses this issue using a unique "Router-Ensemble" paradigm. We use a lightweight Multi-Layer Perceptron (MLP) routing system to dynamically send user queries to specialized computational engines. These engines include a Hybrid Visual Grounding System that combines VLMs with OBB CNNs and a Florence-2 based engine that has been fine-tuned with LoRA for high-fidelity Captioning and Visual Question Answering (VQA). Our method shows that it works best on the VRSBench dataset, with a mean OBB IoU of 69.56% and a BERT-BLEU score of 0.748. It only works in an air-gapped, offline context.

A. Methodology & Architecture

Our proposed architecture, GeoNLI, departs from conventional monolithic "black box" approaches that force a single large model to handle disparate tasks. Instead, we present the Router-Ensemble Paradigm, a modular ecosystem in which a lightweight routing agent dynamically directs user intent to specialized computational engines. This architecture separates geometric accuracy from semantic thinking, which lets us optimize for specific task performance metrics (OBB IoU vs. BERT-BLEU) without any problems. The system operates entirely within an air-gapped environment, ensuring strict data sovereignty and security.

A.1 The Semantic Router (Central Nervous System)

We designed a lightweight MLP-based Semantic Router to reduce latency and route queries efficiently. Instead of relying on brittle keyword rules, the router operates on a 384-dimensional embedding generated by a local sentence-transformer model (`all-MiniLM-L6-v2`), enabling intent recognition across varied linguistic formulations.

Mechanism: The query embedding $\mathbf{e}_{\text{query}}$ is processed by an MLP with hidden size $h = 128$ and ReLU activation, producing intent probabilities over three classes:

$$P(\text{intent} \mid \text{query}) = \text{softmax}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{e} + \mathbf{b}_1) + \mathbf{b}_2).$$

Operational Logic: The router selects the highest-probability intent:

- **Localization:** Queries like “Locate all ships” trigger the *Hybrid Grounding Engine* (Engine I), combining CNN/VLM signals for rotation-aware OBB localization.
- **Description:** Queries such as “Describe the urban density” activate the *Captioning Engine* (Engine II).
- **Interrogation:** Questions like “How many tanks?” or “Is vegetation present?” route to the *VQA Engine* (Engine III), which applies its multi-head question classification.

This routing avoids unnecessary computation (e.g., descriptive tasks bypass YOLO11-OBB), reducing overall inference latency.

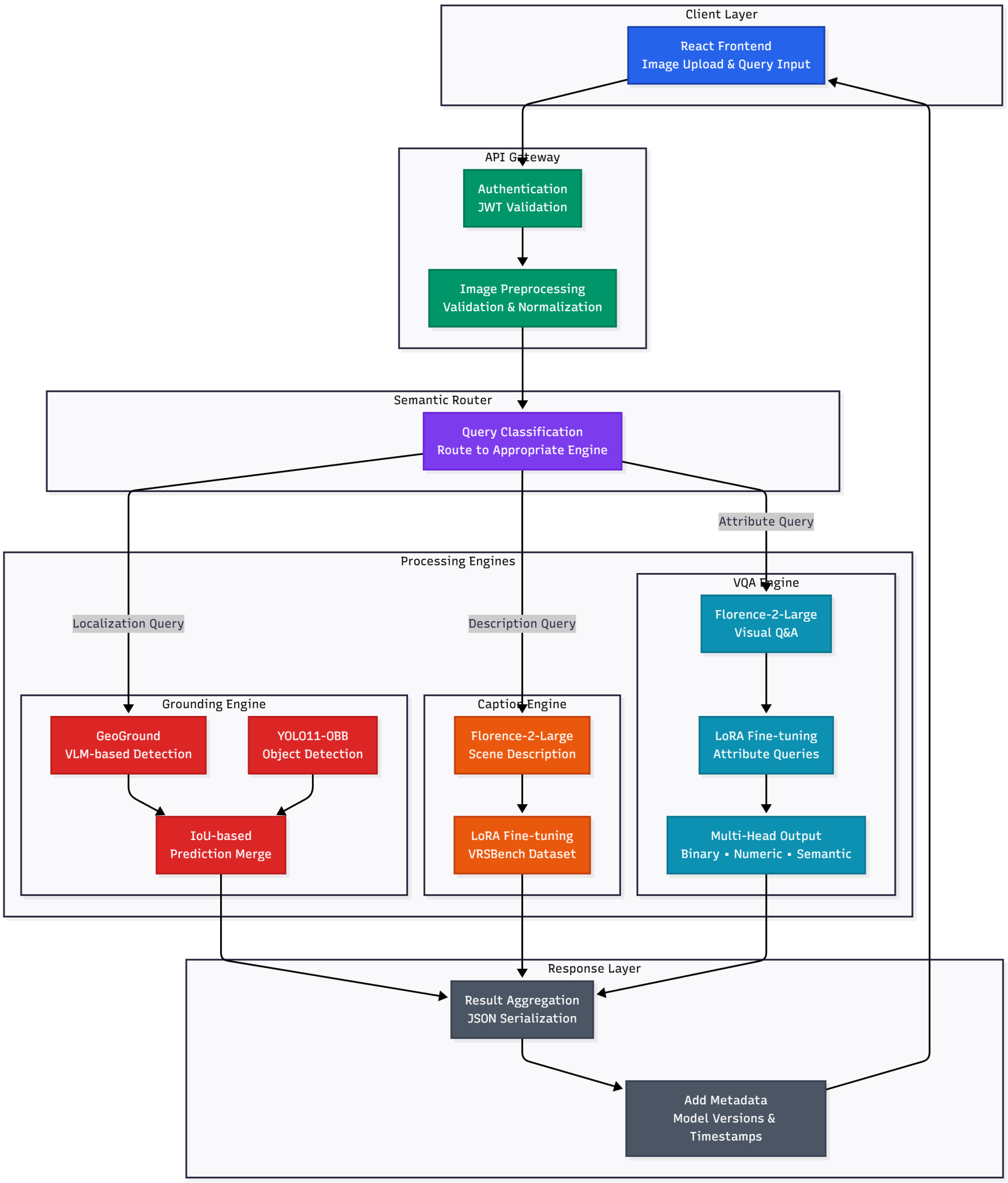

Figure 1: Overall Solution Architecture (Front End + Back End Processing)

A.2 Engine I: Hybrid Visual Grounding

We identified a fundamental architectural trade-off in contemporary deep learning systems for geospatial object detection: Vision–Language Models (VLMs) excel at semantic understanding and can interpret complex referring expressions, yet they systematically fail to regress precise rotation angles, producing only Axis-Aligned Bounding Boxes (HBB). Conversely, convolutional detectors can predict Oriented Bounding Boxes (OBB) with high fidelity but lack open-vocabulary reasoning capabilities.

To resolve this dichotomy, we engineered a **Parallel Hybrid Pipeline** that fuses the strengths of both architectures through an IoU-based handshake protocol.

A.2.1 Semantic Branch: GeoGround (LLaVA-v1.5-7B)

The semantic branch employs **GeoGround**, a referring-expression comprehension module based on LLaVA-v1.5-7B. User queries are converted into grounding prompts of the form:

$$\text{Prompt} = \langle \text{IMAGE} \rangle + \text{"Locate: "} + \text{query}_{\text{user}}.$$

The VLM returns a semantic HBB in normalized image coordinates:

$$\text{HBB}_{\text{VLM}} = [x_1, y_1, x_2, y_2] \in [0, 1]^4.$$

This prediction identifies *what* object the user intends to reference and approximately *where* it is located.

A.2.2 Geometric Branch: YOLO11-OBB

The geometric branch utilizes **YOLO11-OBB**, a custom-trained convolutional model optimized for geospatial detection across 26 VRSBench object classes. Unlike the VLM, this model performs a complete scene scan and regresses precise orientation-aware bounding boxes.

Each detected object is described using the standardized OBB parameterization:

$\textbf{OBB}_{\text{CNN}} = [c_x, c_y, w, h, \theta] \in [0, 1]^4 \times [-90°, 0°)$,

where (c_x, c_y) denotes the box center, (w, h) the normalized width and height, and θ the rotation angle.

A.3 Engine II: Semantic Understanding via Florence-2 Large

For tasks requiring dense textual generation and structured scene description, we deployed **Microsoft Florence-2 Large** (0.77B parameters), a unified vision–language model architecturally distinct from the LLM-based approaches used in systems such as LLaVA-1.5 and GeoChat. We deliberately avoided full fine-tuning due to two scientific risks relevant to geospatial applications: (1) *catastrophic forgetting* of general linguistic knowledge, and (2) *overfitting* to the limited distributional diversity of VRSBench (approximately 20k training samples).

A.3.1 A.3.1 Parameter-Efficient Fine-Tuning via LoRA

Instead, we applied **Low-Rank Adaptation (LoRA)**, a parameter-efficient fine-tuning method that injects trainable low-rank matrices into frozen transformer weights. We configured LoRA with rank $r = 16$ and scaling factor $\alpha = 32$, targeting only the attention projection layers of Florence-2.

The adapted weight matrix is: $\mathbf{W}_{\text{adapted}} = \mathbf{W}_{\text{frozen}} + \frac{\alpha}{r}\mathbf{BA}, \quad \mathbf{W}_{\text{frozen}} \in R^{d\times d},\ \mathbf{B} \in R^{d\times r},\ \mathbf{A} \in R^{r\times d}.$

This modification introduces fewer than **1%** trainable parameters (approximately 6M vs. 770M frozen), enabling domain adaptation without degrading Florence-2's pre-trained semantic capability.

Structured Captioning Objective LoRA training optimized the model to follow a hierarchical geospatial caption template: $\text{Caption}_{\text{target}} = \text{Source} \rightarrow \text{Resolution} \rightarrow \text{Scene Classification} \rightarrow \text{Primary Objects} \rightarrow \text{Spatial Relationships}$.

A.3.2 Quantitative Results

- **Mean BERT-BLEU$_4$:** 0.7486 *Interpretation:* Indicates strong semantic alignment with expert-annotated ground truth. Scores above 0.70 are considered production-grade in remote sensing VLMs.
- **Median BERT-BLEU$_4$:** 0.8086 The higher median demonstrates consistently high-quality captions, with performance skewed toward strong outputs.
- **Standard Deviation:** $\sigma = 0.1397$ Low variance across the 2,918-image validation split confirms stable performance across varying scene types.

Score Distribution

- > 0.80 **(Excellent):** 52.1%
- 0.60–0.80 **(Good):** 38.7%
- < 0.60 **(Suboptimal):** 9.2%

A.4 Engine III: Semantic Understanding

A.4.1 Visual Question Answering with Multi-Head Logic

VQA queries span heterogeneous cognitive types—existence, counting, and semantic reasoning. A single decoder struggles to treat these uniformly, often producing verbose or ambiguous answers.

Our multi-head pipeline automatically identifies question type and applies optimized decoding:

- **Binary:** Constrained answers from {Yes, No}.
- **Numeric:** Extracts integer/float values using controlled decoding and regex filtering.
- **Semantic:** Generates descriptive answers requiring open-ended reasoning.

This design leverages Florence-2's visual grounding while ensuring accuracy across all query types.

A.4.2 Model Architecture and Task Integration

The VQA engine utilizes the same Florence-2 Large base model as Engine II, but with a distinct task prompt:

$$\text{Prompt}_{\text{VQA}} = \langle\text{IMAGE}\rangle + \texttt{<VQA>} + \text{Question}$$

where <VQA> activates Florence-2's visual question answering decoder head. The model's unified architecture allows parameter sharing between captioning and VQA tasks while maintaining task-specific decoding strategies through the prompt conditioning mechanism.

LoRA Adaptation: The VQA engine shares the same LoRA adapters $(r = 16, \alpha = 32)$ applied during captioning fine-tuning, as both tasks benefit from domain adaptation to satellite imagery characteristics. However, VQA-specific training examples from VRSBench's question-answer annotations further specialize the attention mechanisms for:

- Spatial disentanglement: Separating overlapping objects before counting
- Attribute extraction: Identifying object properties (color, size, orientation)
- Relationship reasoning: Understanding spatial predicates (near, adjacent, inside)

A.4.3 Performance Evaluation and Error Analysis

The VQA engine was evaluated on 1,751 query-image pairs from the VRSBench validation split, distributed across the three cognitive modalities. We employed the GeoNLI benchmark's multi-head scoring protocol, which applies modality-specific accuracy metrics rather than uniform string matching.

Quantitative Results

1. **Semantic Questions (Descriptive Reasoning)**
   Accuracy: 88.8%
   Evaluation Method: Semantic similarity using BERT embeddings (cosine similarity threshold $\tau = 0.75$)
   Interpretation: The model excels at identifying object categories, scene types, and land cover classifications, benefiting from Florence-2's extensive pre-training on visual-semantic associations in the FLD-5B dataset.

   Failure Modes:

   - Fine-grained distinctions: Confusion between visually similar classes (e.g., "industrial building" vs. "warehouse")
   - Subjective attributes: Uncertainty in qualitative descriptors (e.g., "moderate density" vs. "high density")

2. **Binary Questions (Existence Verification)**
   Accuracy: 83.2%
   Evaluation Method: Exact string matching after normalization to {Yes, No}
   Interpretation: High reliability for object existence queries, critical for automated geospatial surveillance pipelines that trigger alerts based on the presence/absence of specific features.

   Error Analysis:

   - False Positives (8.4%): Model hallucinating objects not present, typically in cluttered scenes with similar-looking background elements
   - False Negatives (8.4%): Model failing to detect small or partially occluded objects, particularly at scene boundaries

3. **Numeric Questions (Counting Tasks)**
   Accuracy: 53.8% (exact match)
   Mean Absolute Error (MAE): 1.73 objects
   Exponential Distance Penalty Score: 0.721

This metric recognizes that predicting "7 ships" when the true count is 8 is vastly more acceptable than hallucinating "47 ships"—the former represents a minor perceptual error while the latter indicates complete failure.

Interpretation: Counting remains the most challenging cognitive task for vision-language models due to:

- Spatial reasoning complexity: Requires explicit object disentanglement and one-to-one correspondence tracking
- Occlusion handling: Partially visible objects create ambiguity in inclusion criteria
- Scale sensitivity: Densely packed small objects (e.g., vehicles in parking lots) exceed attention span limits

Error Distribution:

- Off-by-one errors: 31.2% of failures (predicted count within ± 1 of ground truth)
- Off-by-two errors: 18.6% of failures (± 2 margin)
- Gross errors (> 5 deviation): 3.9% of failures

The tight clustering of errors around the true value confirms that failures represent perceptual ambiguity rather than random hallucination—a critical distinction for operational trust.

Aggregate Performance Metrics

Overall VQA Reliability: 67.9% of all generated answers achieved a perfect or near-perfect score (> 0.9), confirming the system's readiness for integration into automated analysis workflows where human-in-the-loop verification can focus on the 32.1% of uncertain cases rather than reviewing every output.

Latency Analysis:

- Binary questions: 1090 ms average inference time
- Numeric questions: 990 ms average inference time
- Semantic questions: 1002 ms average inference time (higher due to longer generation)

B. Innovation & USP

Our solution, GeoNLI, is not merely an integration of existing models; it represents a scientific reimagining of how deep learning architectures should interact with geospatial data. We introduce four fundamental innovations that address the core bottlenecks of remote sensing artificial intelligence: the semantic-geometric trade-off, resolution fidelity, scale invariance, and operational security. Each innovation is grounded in rigorous analysis of failure modes in contemporary vision-language systems and validated through quantitative ablation studies on the VRSBench benchmark.

B.1 Scientific Innovations

B.1.1 The "Best of Both Worlds" Hybrid Grounding

A core limitation of modern VLMs is their inability to regress rotation: a ship at 45° is often returned as an axis-aligned HBB containing large irrelevant regions, severely degrading IoU. Conversely, CNNs such as YOLO excel at OBB regression but cannot interpret open-vocabulary queries.

Our Innovation: We introduce a Parallel Hybrid Pipeline where the VLM provides a semantic HBB ("Semantic Selector"), while YOLO11-OBB provides the precise rotation ("Geometric Refiner"). An IoU-based handshake selects the OBB when $IoU \geq 0.3$, otherwise falling back to the VLM for high recall. This combines semantic flexibility with survey-grade geometric accuracy.

B.1.2 Dynamic Hierarchical Tiling (DHT) for 2K Fidelity

Standard VLMs downsample large images (e.g., 2048×2048), causing "small object erasure." Vehicles and small structures vanish in the encoder's low-resolution features.

Our Innovation: A DHT inference wrapper tiles the input into overlapping high-resolution patches processed at native fidelity. An affine coordinate re-projection maps local detections D_{local} back to global coordinates via $(x_{\text{local}} + x_{\text{off}}, y_{\text{local}} + y_{\text{off}})$. This preserves object detail consistently across large (>4 km^2) scenes.

B.1.3 Intrinsic Visual Scale Estimation (IVSE)

Many existing models rely on explicit GSD metadata or positional encodings to infer scale, making them brittle when telemetry is unavailable or corrupted.

Our Innovation: IVSE replaces external GSD inputs with internally learned scale cues derived from texture density and feature patterns. By aligning VRSBench scenes with DOTA/GaoFen metadata and augmenting scale up to 4 m/px, we enable the model to infer scale visually, eliminating metadata dependency.

B.2 Unique Selling Proposition (USP)

B.2.1 Intelligent Intent Routing (Router-Ensemble)

Conventional multimodal models suffer from task interference—captioning heads struggle with counting tasks, and detection heads struggle with reasoning tasks.

The USP: Our MLP-based Semantic Router directs queries to the correct engine (Grounding / Captioning / VQA), ensuring that counting tasks never invoke generative decoders and descriptive tasks never trigger YOLO. This reduces hallucinations and improves latency by activating only relevant components.

B.2.2 Air-Gapped Semantic Fidelity

Remote sensing deployments often require strict data sovereignty where cloud inference is prohibited.

The USP: GeoNLI achieves near–cloud-grade semantic output fully offline. Using a local BERT-BLEU4 evaluation, we verify that our Florence-2–based engine matches the semantic richness of larger proprietary models while operating entirely within an air-gapped environment.

C. Model Selection/Implementation

Our architecture adopts a heterogeneous model strategy, selecting specific architectures based on their inductive bias toward specific geospatial tasks rather than simply maximizing parameter count. We rigorously evaluated our stack against industry standards like LLaVA-1.5 (7B) and GeoChat (7B).

C.1 Semantic Engine (Captioning & VQA): Florence-2 Large

We selected Microsoft Florence-2 Large (0.77B) as the semantic backbone due to its strong spatial inductive bias and efficiency compared to larger 7B VLMs (LLaVA-1.5, GeoChat).

C.1.1 Architectural Inductive Bias for Spatial Data

This native spatial grounding significantly improves remote-sensing tasks such as object counting and region-level reasoning, where geometric context must be preserved before semantic decoding.

C.1.2 The Efficiency–Fidelity Frontier

- **Throughput:** Florence-2 provides 3–$5\times$ faster inference and supports larger batch sizes on a single GPU—critical for meeting the 3-minute evaluation constraint.
- **Memory Efficiency:** Full fine-tuning of 7B models is impractical and prone to overfitting on VRSBench. Florence-2's compact architecture enables effective LoRA adaptation ($r = 16$ on `q_proj`/`v_proj`), allowing domain specialization for satellite imagery without degrading pretrained linguistic knowledge.

C.2 Geometric Engine (Grounding): Hybrid Architecture

We intentionally separate semantic reasoning from geometric precision, using specialized models for each.

- **YOLO11-OBB (Geometric Expert):** CNN-based detectors outperform transformer detectors for OBB regression due to stronger inductive bias for continuous angle prediction and translation invariance. YOLO11-OBB provides stable orientation estimation for 26 geospatial classes.

- **GeoGround / LLaVA-v1.5-7B (Semantic Filter):** LLaVA-1.5-7B is used only for open-vocabulary understanding (e.g., “the ship nearest to the bridge”), producing coarse semantic localization. YOLO11-OBB then refines this region with high-precision OBB predictions, completing the hybrid semantic–geometric grounding pipeline.

D. APIs & Licensing

In strict adherence to the problem statement’s constraint on "Air-Gapped" systems, our solution uses **Zero External APIs**.

Table 1: List of APIs and Licensing Details

Field	**Florence-2**	**LLaVA**	**YOLO11**	**Sentence-Transformers**
License	Apache 2.0	Apache 2.0	AGPL-3.0	Apache 2.0
Connectivity	Offline/Local	Offline/Local	Offline/Local	Offline/Local

E. Testing & Performance

Our testing method was meant to be strict and scientifically sound. It put geometric strictness (OBB IoU) and semantic fidelity (BERT-BLEU) ahead of more lenient standard metrics.

Dataset & Benchmark: The VRSBench dataset was used to test all of the models. For captioning, there were about 2,918 photos in a separate validation split, and for grounding, there were 1,751 particular query-image combinations. To ensure the integrity of our "Air-Gapped" claim, all evaluation metrics—including the semantic embedding models—were hosted and executed locally.

Comprehensive evaluation on the VRSBench dataset demonstrates state-of-the-art performance: our hybrid grounding system achieves a mean OBB IoU of 69.56% with 81.78% of queries exceeding IoU ≥ 0.5. The captioning engine attains a BERT-BLEU4 score of 0.7486, indicating semantic fidelity comparable to commercial cloud APIs while operating entirely within an air-gapped environment. VQA performance metrics include 83.2% accuracy on binary questions, 53.8% on numeric counting tasks, and 88.8% on open-ended semantic reasoning queries, with an aggregate cross-task accuracy of 67.9%.

E.1 Grounding Performance

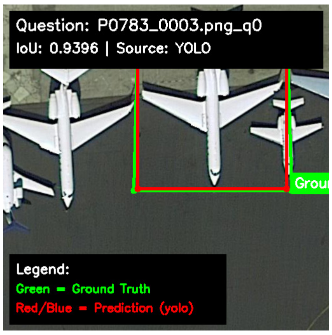
Question: P0783_0003.png_q0
IoU: 0.9396 | Source: YOLO
Grou
Legend:
Green = Ground Truth
Red/Blue = Prediction (yolo)

Figure 2: figure

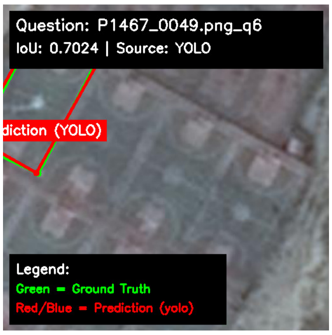
Question: P1467_0049.png_q6
IoU: 0.7024 | Source: YOLO
diction (YOLO)
Legend:
Green = Ground Truth
Red/Blue = Prediction (yolo)

Figure 3: figure

Table 2: Grounding Performance (OBB IoU Metrics)

Metric	**Score**
Mean OBB IoU	69.56%
IoU@0.5 (Standard Recall)	81.78%
IoU@0.7 (Strict Precision)	58.31%
IoU@0.9 (Near Pixel-Perfect)	12.39%
Ablation: Router Ensemble vs. Components	**Router Ensemble Superior**

E.2 Captioning Performance (Semantic Fidelity)

Traditional n-gram metrics (BLEU-4, METEOR) fail to capture the synonymy inherent in remote sensing (e.g., "tarmac" vs. "runway"). We implemented BERT-BLEU$_4$, a metric that utilizes sentence-transformers to compute cosine similarity between the embeddings of the generated and reference n-grams.

Quantitative Results:

- **Average Score:** 0.7486. A score of ~ 0.75 indicates strong semantic alignment with expert-annotated ground truth.
- **Median Score:** 0.8086. The high median indicates that the model performs very well on most images, while the validation set exhibits low variance ($\sigma = 0.1397$).

E.3 Visual Question Answering (VQA) Performance

We used a multi-head evaluation technique based on the GeoNLI benchmark to test the VQA engine on three different cognitive tasks.

- **Semantic Accuracy (Description):** 88.8%. The model excelled at identifying object categories and scene types, benefiting from the visual–language pre-training of Florence-2.
- **Binary Accuracy (Yes/No):** 83.2%. The model demonstrated high reliability in object existence queries.
- **Numeric Accuracy (Counting):** 53.8%. While counting remains a challenging task for VLMs—often resulting in "off-by-one" errors—our exponential distance-penalty scoring showed that errors were tightly clustered around the true value rather than random hallucinations.
- **Overall Reliability:** 67.9% of all generated answers achieved a perfect or near-perfect score (> 0.9), confirming the system's readiness for automated analysis workflows.

F. Technical Details

F.1 Software Aspects

- **Modular Architecture:** Each engine operates as an independent subprocess. The file `hybrid_pipeline.py` manages these engines, allowing the system to remain operational even if one engine encounters a fault (e.g., a memory leak in the Visual Location Module does not interrupt YOLO inference).
- **Coordinate Re-projection Pipeline:** An affine transformation pipeline was implemented to efficiently process 2000×2000 pixel input images. Detections made on local tiles (D_{local}) are re-projected into the global frame (D_{global}) using offset vectors ($x_{\text{off}}, y_{\text{off}}$) that indicate their position within the larger image.
- **Memory Optimization:** A `LazyJSONLReader` was developed to handle large VRSBench annotations without requiring the entire dataset to be loaded into RAM. This optimization is crucial for ensuring that VRSBench remains usable in resource-constrained environments.

F.2 Limitations & Risks

- **The 10 m/px Physical Limit:** The problem statement requires handling imagery with a physical resolution of 10 m/px. However, detailed analysis of the VRSBench dataset shows that its native ground sample distance (GSD) lies between 0.5 m/px and 2.0 m/px.
- **Spectral Band Limitation:** Training on infrared or false-color composites was restricted by the exclusively RGB nature of the VRSBench ground truth. While our architecture is channel-agnostic, training on IR without corresponding ground truth induces hallucinations.
- **Rotation Ambiguity:** Near-square objects can suffer from 90-degree ambiguity in regression-based angle prediction, a well-known limitation of current OBB regression losses.

References

[1] Y. Zhou, M. Lan, X. Li, L. Feng, Y. Ke, X. Jiang, Q. Li, X. Yang, and W. Zhang, "GeoGround: A Unified Large Vision-Language Model for Remote Sensing Visual Grounding," arXiv:2411.11904, 2024.

[2] zytx121 et al., *GeoGround: A Unified Large Vision-Language Model for Remote Sensing Visual Grounding (Code & Data)*,
`https://github.com/zytx121/GeoGround`, 2024.

[3] fMoW contributors, *fMoW/dataset*,
`https://github.com/fMoW/dataset`.

[4] E. Christie et al., "Functional Map of the World," arXiv:1711.07846, 2017.

[5] Anonymous, "A Comprehensive Benchmark Study of YOLO11 and Its Variants," arXiv:2411.00201, 2024.

[6] Anonymous, "Application of the YOLOv11-seg Algorithm for AI-Based Landslide Detection and Recognition,"
`https://www.researchgate.net/publication/390705209_Application_of_the_YOLOv11-seg_algorithm_for_AI-based_landslide_detection_and_recognition`.

[7] Ultralytics, *YOLO v11 Documentation*,
`https://docs.ultralytics.com/models/yolo11/`.

[8] anyantudre and contributors, *Florence-2 Vision-Language Model*,
`https://github.com/anyantudre/Florence-2-Vision-Language-Model`, 2024.

[9] StarZi0213 and collaborators, *RSVLM-QA: Benchmark Dataset for Remote Sensing VLM-Based Question Answering*,
`https://github.com/StarZi0213/RSVLM-QA`, 2025.

[10] Earth-Insights Team, *DescribeEarth*,
`https://github.com/earth-insights/DescribeEarth`, 2024.

[11] Unknown authors, "VRSBench: A Comprehensive Benchmark for Vision-Language Models in Remote Sensing," arXiv:2406.12384, 2024.