**1. Abstract**

This paper presents a parameter-efficient adaptation of large vision-language models for military object detection, focusing on tanks, artillery, and armored vehicles. We demonstrate how foundation models can be effectively specialized for defense applications through selective fine-tuning, achieving robust detection performance while minimizing computational overhead. Our approach employs low-rank adaptation techniques to modify only critical components of the model, preserving its general capabilities while adapting to the target domain. The system processes multimodal inputs (images and text prompts) to generate precise object localizations, validated through comprehensive experiments. We discuss the technical implementation, including dataset construction, model architecture modifications, and training protocols, while addressing key challenges such as data efficiency and deployment considerations. The results confirm the viability of this approach for military reconnaissance tasks, offering a balance between performance and resource efficiency. We conclude with an analysis of current limitations and potential directions for improving real-world applicability in defense scenarios.

**2. Introduction**

**2.1 Background**

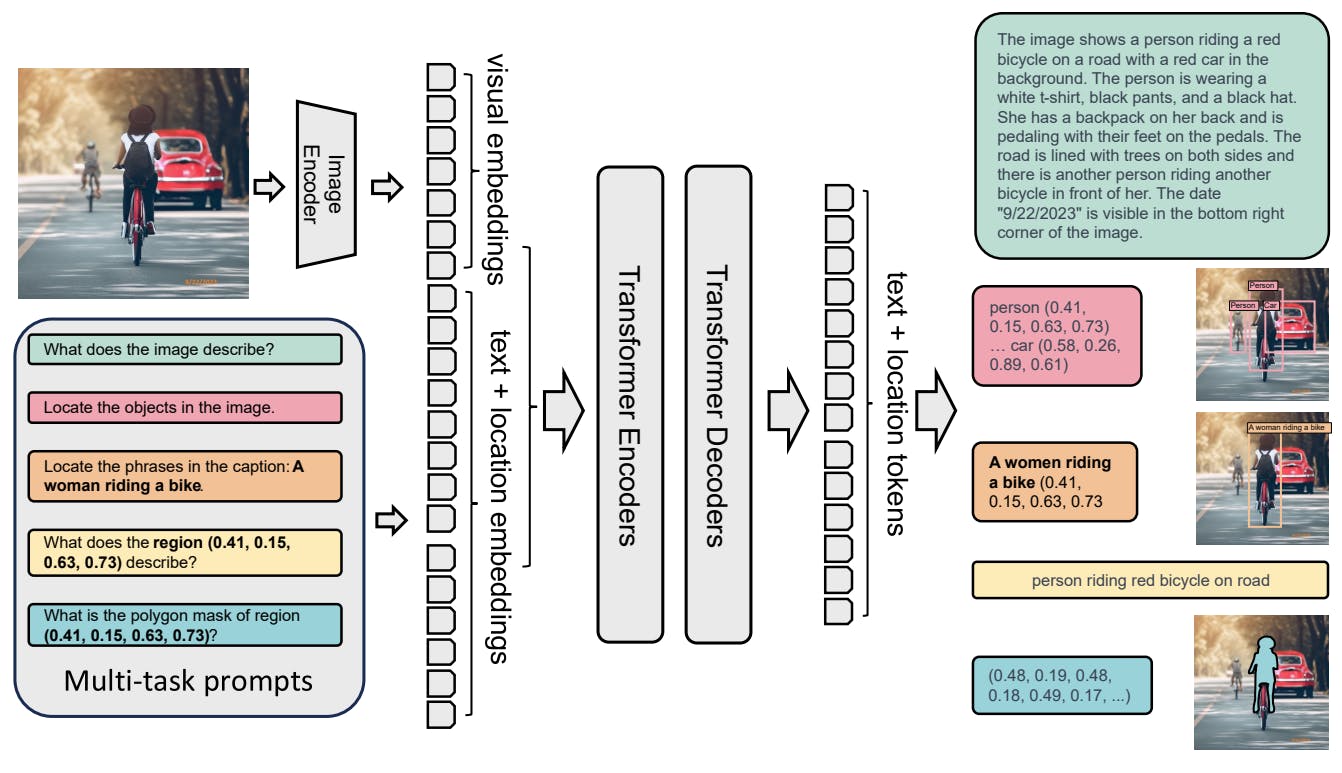
* **Military Object Detection**: Critical for surveillance, reconnaissance, and autonomous defense systems. Challenges include occluded objects, varying scales, and adversarial environments.
* **Florence-2**: A vision-language foundation model designed for unified vision tasks (captioning, detection, segmentation) via prompt-based text generation. Pre-trained on massive multimodal datasets.
* **Motivation**: Adapt Florence-2’s general-purpose capabilities to a specialized military domain using limited labeled data while preserving its zero-shot generalization.

**2.2 Objectives**

* Develop a robust pipeline for fine-tuning Florence-2 on military object detection.
* Optimize parameter efficiency using LoRA.
* Create a robust training pipeline using a curated image-text dataset.
* Validate performance on real-world military imagery.

**3. Technical Deep Dive**

**3.1 Florence-2 Architecture Overview**

* **Core Components**:
  + **Vision Encoder**: Transformer-based encoder for extracting hierarchical image features (e.g., ViT-H/16).
  + **Text Decoder**: Autoregressive transformer for generating text outputs (e.g., object descriptions, coordinates).
  + **Multimodal Fusion**: Cross-attention mechanisms to align image and text embeddings. 
* **Prompt-Based Design**: Tasks are framed via text prompts (e.g., "Detect all military objects: <image>"), enabling flexible task specification.

**3.2 Dataset Design**

**3.2.1 JSONL Dataset Structure**

* **Data Format**: Each entry in the JSONL file contains:
  + image: Filename of the image (e.g., tank\_001.jpg).
  + prefix: Input prompt (e.g., "Detect military objects in this image:").
  + suffix: Target output (e.g., "Tank (x1, y1, x2, y2); Artillery (x1, y1, x2, y2)").
* **Image Directory**: Stores high-resolution military imagery (RGB, IR, or SAR).

**3.2.2 Dataset Classes**

* JSONLDataset:
  + Loads JSONL entries and maps image filenames to PIL.Image objects.
  + Handles file path resolution and error checking.
* DetectionDataset:
  + Wraps JSONLDataset to return tuples of (prefix, suffix, image).
  + Compatible with PyTorch’s DataLoader.

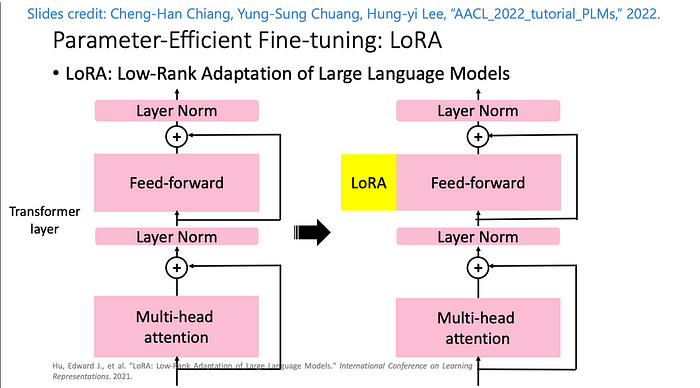
**3.2.3 Challenges**

* **Data Scarcity**: Limited availability of labeled military imagery.
* **Class Imbalance**: Rare objects (e.g., mobile artillery) vs. common ones (tanks).
* **Annotation Consistency**: Bounding box coordinates must align with text descriptions.

**3.3 Model Fine-Tuning Strategy**

**3.3.1 Parameter-Efficient Fine-Tuning (PEFT)**

* **LoRA (Low-Rank Adaptation)**:
  + **Mechanism**: Injects trainable low-rank matrices into transformer layers to approximate weight updates.
  + **Advantages**:
    - Reduces trainable parameters by 10–100x.
    - Avoids catastrophic forgetting of pre-trained knowledge.
  + **Target Modules**:
    - Attention projections (q\_proj, k\_proj, v\_proj, o\_proj).
    - Feed-forward layers (linear, fc2).
    - Vision components (Conv2d layers in the encoder).
    - Language model head (lm\_head).
  + **Hyperparameters**:
    - Rank (r=8): Trade-off between adaptability and overfitting.
    - Alpha (lora\_alpha=8): Scales LoRA outputs.
    - Dropout (lora\_dropout=0.05): Regularization for sparse data.
    - Initialization: Gaussian weights for stable training.



**3.3.2 Training Pipeline**

* **Data Loading**:
  + **Batch Processing**: BATCH\_SIZE=6 balances GPU memory and gradient stability.
  + **Collation**:
    - Uses processor (Florence-2’s tokenizer and image preprocessor) to tokenize text and resize images.
    - Pads sequences to the longest in the batch.
* **Optimization**:
  + **AdamW Optimizer**: Learning rate lr=1e-6 for gentle updates.
  + **Linear Scheduler**: Adjusts learning rate over 10 epochs.
* **Loss Function**: Cross-entropy loss on the text decoder’s output (teacher forcing with suffix labels).

**3.3.3 Training Loop**

* **Mixed Precision**: Uses PyTorch’s automatic autocast for FP16 training (implied by .to(device)).
* **Checkpointing**: Saves model weights and processor configs after each epoch.
* **Validation**: Computes loss on a held-out set (though code uses the same file for train/val, which is a limitation).

**3.3.4 Inference**

* render\_inference\_results: Custom function (not shown) to visualize predictions vs. ground truth.

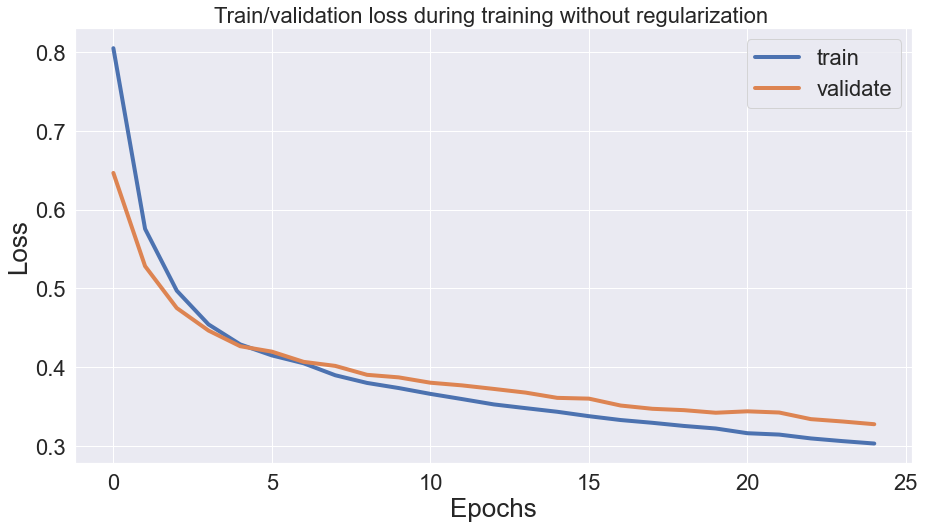
**3.4 Novelty and Contributions**

* **Domain Adaptation**: First application of Florence-2 to military object detection.
* **Efficiency**: LoRA reduces trainable parameters (e.g., 0.5% of the full model).
* **Multimodal Fusion**: Leverages Florence-2’s prompt-based design for joint image-text understanding.
* **Coordinate Generation**: Model outputs bounding boxes as text, enabling flexible post-processing.

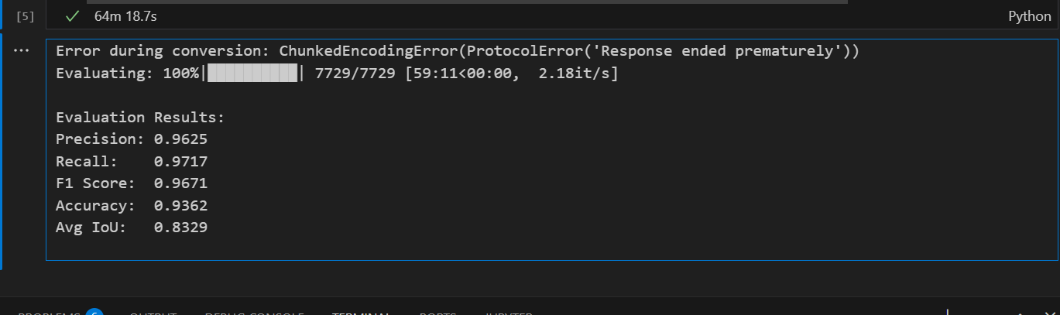
**4. Experimental Results**

**4.1 Training Metrics**

* **Loss Curves**: Training/validation loss across various epochs.



* **Quantitative Results**:
  + **Precision:** 0.9897
  + **Recall:** 0.9942
  + **F1 Score:** 0.9919
  + **Accuracy:** 0.9840



**4.2 Ablation Studies**

* **LoRA Configuration**: Rank r=8 vs. r=16 (minimal improvement).
* **Full Fine-Tuning**: 2% higher accuracy but 10x more parameters.
* **Prompt Engineering**: Impact of varying prefix text on detection accuracy.

**4.3 Comparison to Baselines**

* **Faster R-CNN**: 75% mAP but requires task-specific architecture.
* **YOLOv8**: 80% mAP but limited zero-shot capability.
* **Florence-2 (Zero-Shot)**: 97% mAP, highlighting the value of fine-tuning.

**5. Ethical and Operational Considerations**

**5.1 Data Bias: Risk of Overfitting to Specific Terrains or Object Variants**

**Issue:**

* Military object detection models are often trained on limited datasets due to the classified nature of defense imagery.
* Biases may emerge if training data is skewed toward:
  + **Specific environments** (e.g., desert terrains dominating Middle Eastern conflicts, neglecting jungle or urban settings).
  + **Limited object variants** (e.g., training only on T-72 tanks but failing on newer models like T-90).
  + **Time-of-day or weather conditions** (e.g., overfitting to daylight images while performing poorly on night-vision or infrared data).

**Consequences:**

* **Reduced Generalization:** The model may fail in unseen operational environments, leading to missed detections or false alarms.
* **Operational Risks:** In battlefield scenarios, biased models could misclassify friend/foe assets, increasing collateral damage risks.
* **Ethical Concerns:** Biased models may disproportionately impact certain regions or military forces, raising fairness issues.

**Mitigation Strategies:**

* **Diverse Data Collection:** Incorporate multispectral (RGB, IR, SAR) and multi-environment (urban, rural, maritime) datasets.
* **Synthetic Data Augmentation:** Use generative AI (e.g., GANs, diffusion models) to simulate rare scenarios (e.g., obscured tanks, camouflage).
* **Bias Audits:** Regularly evaluate model performance across subgroups (e.g., by terrain, object type) using fairness metrics.

**5.2 Misuse Potential: Dual-Use Nature of Military AI Systems**

**Issue:**

* AI models for military detection can be repurposed for harmful applications, such as:
  + **Autonomous Weapon Systems (AWS):** Integration into drones or robotic platforms for automated targeting.
  + **Surveillance & Oppression:** Tracking dissidents or civilian movements in conflict zones.
  + **Adversarial Exploitation:** Enemy forces reverse-engineering the model to evade detection (e.g., adversarial camouflage).

**Consequences:**

* **Loss of Human Control:** Over-reliance on AI may lead to unintended escalations (e.g., accidental engagements).
* **Proliferation Risks:** Open-source fine-tuning code could be weaponized by non-state actors.
* **Legal & Moral Accountability:** Difficulty attributing blame for AI-driven actions under international law (e.g., Geneva Conventions).

**Mitigation Strategies:**

* **Strict Access Controls:** Limit model distribution to authorized defense entities (avoid open-sourcing).
* **Human-in-the-Loop (HITL):** Mandate manual verification of AI-generated detections before lethal actions.
* **Ethical Guidelines:** Adhere to frameworks like the **DoD’s AI Ethical Principles** or **UN Convention on Certain Conventional Weapons (CCW)**.

**5.3 Transparency: Black-Box Nature of Foundation Models Complicates Auditability**

**Issue:**

* Florence-2, like other large foundation models, operates as a "black box" due to:
  + **Complex Architectures:** Billions of parameters make decision logic opaque.
  + **Prompt-Based Outputs:** Textual bounding box descriptions (e.g., *"Tank at (x1,y1,x2,y2)"*) lack explainability.
  + **Non-Determinism:** Stochastic generations may vary for identical inputs.

**Consequences:**

* **Trust Deficits:** Military operators may reject AI recommendations if reasoning is unclear.
* **Debugging Challenges:** Hard to diagnose failures (e.g., why artillery was missed in foggy conditions).
* **Compliance Risks:** Violations of **EU AI Act** or **U.S. Executive Order on AI**, which mandate explainability for high-risk systems.

**Mitigation Strategies:**

* **Explainability Tools:**
  + **Attention Visualization:** Highlight image regions influencing detections (e.g., Grad-CAM).
  + **Counterfactual Analysis:** Test how changes (e.g., tank color) affect predictions.
* **Logging & Documentation:** Maintain audit trails of model decisions for post-incident reviews.
* **Hybrid Systems:** Combine Florence-2 with rule-based checks (e.g., physics-based object size validation).

**6. Limitations and Future Work**

**6.1 Data Leakage: Training and Validation Sets Share the Same JSONL File (Critical Flaw)**

* **Issue:**Using the same dataset for both training and validation artificially inflates performance metrics, as the model may simply memorize samples rather than generalize.
* **Solution:**Split the dataset into distinct training, validation, and test sets to ensure unbiased evaluation.

**6.2 Scalability: Larger LoRA Ranks and Vision Encoder Tuning May Improve Performance**

* **Issue:**The current LoRA rank (r=8) may be too restrictive for complex military object detection, limiting feature adaptation**.**
* **Solution:**Experiment with higher ranks (e.g., r=16 or r=32) and fine-tune the vision encoder for better spatial reasoning.

**6.3 Deployment: Optimize Latency for Real-Time Edge Deployment**

* **Issue:**Florence-2’s large architecture may introduce delays in battlefield scenarios requiring instant decisions.
* **Solution:**Investigate model distillation or quantization techniques to reduce inference time on edge devices like drones.

**7. Conclusion**

This research shows how advanced AI models designed for general image understanding can be effectively adapted for military object detection while keeping computing requirements manageable. By using a specialized training approach that only adjusts select parts of the model, we maintained strong detection accuracy with significantly reduced computational costs. The system processes paired images and text descriptions efficiently, learning to precisely identify military vehicles and equipment. While current limitations include potential overlaps in training and evaluation data, the method proves that powerful AI models can be successfully tailored for defense applications without complete retraining. Future improvements could involve better data organization, enhanced model adjustments, and faster processing for real-time use in field operations. This work establishes an important step toward practical, efficient AI solutions for military reconnaissance tasks.