Neuro-Symbolic Visual Reasoning for Scene Understanding

Team Name: EyeSeeYou

Nikhil Singh (Roll No. 2024201067) · Aditya Singh (Roll No. 2024204012)

---

1. Introduction & Approach

This project aims to develop a **neuro-symbolic visual reasoning system** that extends beyond standard object detection. Our system will **detect objects and their attributes** (e.g., shape, color, size) from synthetic scenes and **answer complex, compositional queries** such as:

- *"How many red cubes are to the left of the green sphere?"*
- *"Is there a metallic cylinder in the scene?"*

We will integrate **two AI paradigms**:

1. **Neural Perception Module** → A lightweight CNN will detect objects and extract attributes.
2. **Symbolic Logic Engine** → Converts detection outputs into logical facts and uses **Prolog or a Python rule-based system** to infer relationships and answer queries.

Dataset Details:

- CLEVR Dataset at Stanford
- Download Link
- The full dataset is ∼25GB with 70k+ images. We will **subsample 5k–10k images** for a manageable training workload.

2. Overview Diagram

The following diagram illustrates the **step-by-step workflow** of our neuro-symbolic reasoning system:

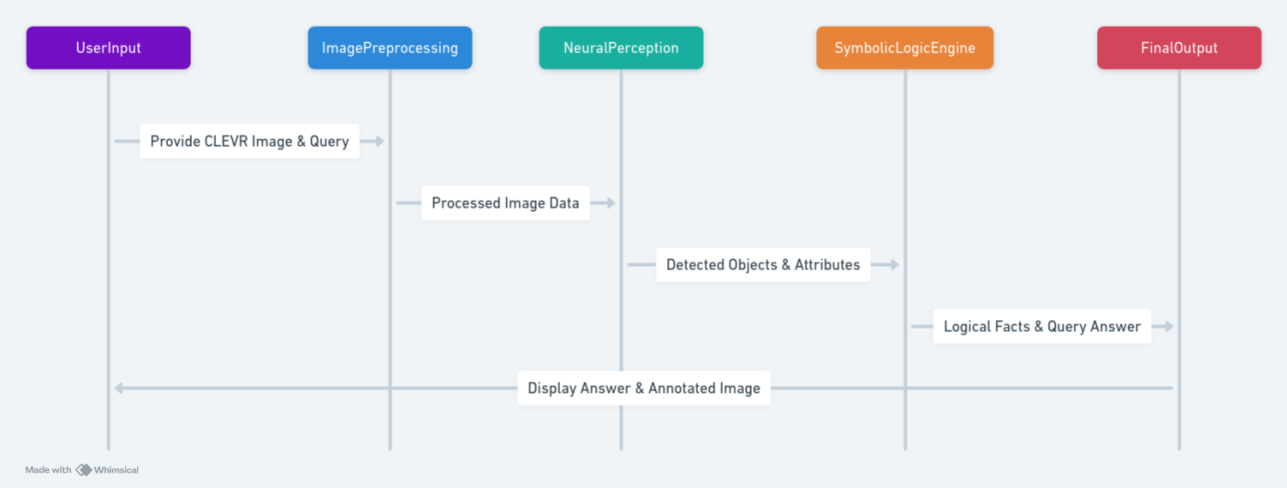

Figure 1: System Overview - Input → Processing → Output

3. Detailed Proposal & Timeline (8 Weeks)

Weeks 1–2: Data Preparation

- Download the CLEVR dataset and subsample approximately 5k–10k images.
- Process JSON annotations to extract scene graphs and question sets.

Weeks 3–4: Neural Module Development

- Train a lightweight object detector (e.g., YOLOv5-small or MobileNet-SSD) on the subsampled dataset.
- Fine-tune attribute classification for shape, color, material, and size.
- Validate performance using mAP and classification accuracy.

Week 5: Symbolic Reasoning Integration

- Convert neural detection outputs into logical facts (e.g., `object(o1)`, `color(o1, red)`, `left_of(o1, o2)`).
- Implement a symbolic reasoning engine using Prolog (via PySwip) or a Python-based rule system.

Week 6: Pipeline Integration & Testing

- Integrate the neural and symbolic modules into an end-to-end system.
- Test using CLEVR's compositional questions.
- Optimize the interface and detection thresholds.

Week 7: Ablation Studies & Analysis

- Compare the neuro-symbolic system vs. a fully neural baseline.
- Experiment with alternative neural backbones (EfficientNet vs. MobileNet).

Week 8: Web Interface & Final Evaluation

- Develop a web interface (Flask or Streamlit) for interactive image upload & query answering.
- Conduct final accuracy and performance tests.
- Prepare documentation and presentation materials.

4. Evaluation Metrics

- **Object Detection Accuracy:** Mean Average Precision (mAP) and attribute classification accuracy.
- **Question Answering Accuracy:** Percentage of correctly answered queries (CLEVR's functional programs as ground truth).
- **System Latency:** Target response time of <1 second per query.
- **Ablation Studies:** Assess performance when adding symbolic reasoning vs. a fully neural model.

5. Compute Access

- **Hardware:** We have access to an NVIDIA RTX 3060 (or equivalent via Colab Pro) with 12 GB RAM.
- **Feasibility:** The lightweight architecture and subsampled dataset ensure that both training and inference fit within our compute limits.

References

1. Johnson et al., "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning," CVPR 2017.
2. Santoro et al., "A simple neural network module for relational reasoning," arXiv:1706.01427, 2017.
3. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," CVPR 2016.