Requirements Document
Demo: A Tool for Assessing Adversarial Defense Capabilities in Vision-Language Models

1. Introduction

This document specifies the technical requirements and detailed execution plan for the demo of our Visual Question Answering (VQA) adversarial-sample detection tool, to be presented at ECAI. The organizers will endeavor to satisfy the requirements below but cannot guarantee full compliance in all respects.

2. Technical Requirements

Component	**Specification**
Hardware	NVIDIA GeForce RTX 4060 Ti (16 GB)
CUDA	12.4
Operating System	Ubuntu 22.04.5 LTS
CPU	≥4 cores, 2.5 GHz or higher
Memory	≥16 GB RAM
Storage	≥20 GB free disk space
Network	Internet access (or offline mirror)

Software Dependencies

- **Python**: version 3.12
- **Python Packages**: install via

```
pip install -r requirements.txt
```

3. Demo Execution Plan

3.1 Repository Setup

1. Clone the repository:

```
git clone https://github.com/peilin1011/VLmode_adv_detection.git
cd VLmode_adv_detection
```

2. Create and activate a virtual environment:

```
python3 -m venv env
source env/bin/activate
```

3. Install dependencies:

```
pip install -r requirements.txt
```

4. Launch the demo:

```
python3 app.py
```

   Access the web interface at `http://127.0.0.1:5000/`.

3.2 Scenario 1: Test Mode (Labeled Data)

1. **Detection Settings**:
   - Image Detection: `feature_squeezing_2`
   - Text Detection: `MaskPure`
   - Joint Detection: `JointDetection`
   - VQA Model: `BLIP`
2. Select mode **Test (Data with Labels)**.
3. Upload folder `examples/test_examples`.
4. Click **Submit** to run detection.
5. View logs and click **View Image** for sample inspection.
6. Download results via **Download Results as CSV**.
7. Return to start with **Back to Upload Page**.

Scenario 2: Inference Mode (Unlabeled Data)

1. **Detection Settings** (same as Test Mode).
2. Select mode **Inference**.
3. Upload image `examples/inference_examples/262197003_adversarial.png` and enter the question:

```
Is there a person riding a bike?
```

4. Click **Submit** to run detection.
5. Inspect results and images as above.
6. Download results via **Download Results as CSV**.
7. Return to start with **Back to Upload Page**.

4. Additional Notes

Each upload directory should include an `info.json` file mapping image filenames to text descriptions. See `examples/test_examples/info.json` and `examples/inference_examples/info.json` for reference.