

Deployment Instructions

A. Local Deployment

Clone Repository:

```
git clone <repository-url>
cd <repository-directory>
```

Set Up Virtual Environment:

```
python3 -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate
```

Install Requirements:

```
pip install torch transformers huggingface_hub opencv-python numpy
ipywidgets scikit-learn
```

Run Jupyter Notebook:

```
jupyter notebook
```

Upload and Run the Notebook:

Open the notebook in Jupyter, upload a video, and run each cell sequentially.

B. Deployment on AWS

For cloud deployment, AWS EC2 with a GPU instance is recommended.

1. Launch EC2 Instance:

- **Instance Type:** `g4dn.xlarge` or similar with GPU support.
- **AMI:** Deep Learning AMI (Ubuntu).
- **Storage:** Minimum 20 GB.
- **Security Group:** Allow ports 8888 (Jupyter Notebook) and 22 (SSH).

SSH into Instance:

```
ssh -i <your-key.pem> ubuntu@<instance-public-dns>
```

Set Up Python Environment:

```
sudo apt update
sudo apt install python3.8 python3.8-venv
python3.8 -m venv venv
source venv/bin/activate
pip install torch transformers huggingface_hub opencv-python numpy
ipywidgets scikit-learn
```

Run Jupyter Notebook:

```
pip install jupyter
jupyter notebook --no-browser --port=8888
```

Access Jupyter Notebook:

Set up port forwarding:

```
ssh -i <your-key.pem> -L 8888:localhost:8888
ubuntu@<instance-public-dns>
```

- Open `localhost:8888` in a browser and enter the token to access Jupyter Notebook.

Upload and Run the Notebook.

Optional: Docker Containerization

For a more portable deployment, can containerize the application using Docker.

Dockerfile:

```
FROM pytorch/pytorch:latest

WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt

COPY . .
EXPOSE 8888
```

```
CMD ["jupyter", "notebook", "--ip=0.0.0.0", "--port=8888",  
"--no-browser", "--allow-root"]
```

Build and Run the Docker Container:

```
docker build -t video-anomaly-detection .
```

```
docker run -p 8888:8888 video-anomaly-detection
```