**A purple circle with dots and lines

Description automatically generated**

**NAME:**

**Basit Abbas**

**ROLL NUMBER:**

**SU92-BSDSM-F23- 013**

**PROGRAM:**

**BS DATA SCIENCE**

**SUMBITTED TO:**

**SIR RASIKH**

***TASK : 06***

**Herd Detection System Report**

**Overview**

The Herd Detection System is a machine learning-based application utilizing YOLOv8 for real-time cattle herd detection. It incorporates geolocation tagging, an alert system, and a FastAPI-based web interface to provide an end-to-end solution for monitoring and tracking herds. The system is designed to process both images and video footage to detect herds and send alerts with GPS coordinates.

**Components**

**1. Dataset Acquisition**

* The script downloads a dataset from Roboflow using an API key.
* The dataset is structured for training YOLOv8.
* The download\_roboflow\_dataset function ensures that the dataset is available before model training.

**2. YOLOv8 Model Training**

* A pretrained YOLOv8 model (yolov8n.pt) is used.
* The model is trained on the downloaded dataset for 100 epochs with a batch size of 16.
* The trained model is saved for inference.

**3. Herd Detection in Video**

* The detect\_herds\_in\_video function processes a video frame-by-frame.
* Every 10th frame is analyzed for object detection.
* If a herd is detected, GPS coordinates are extracted and alerts are sent.

**4. Geolocation Tagging**

* The get\_gps function extracts GPS coordinates from image metadata.
* This helps in mapping detected herds accurately.

**5. Alert System**

* The send\_alert function sends an email notification when a herd is detected.
* SMTP is used to send emails, requiring a sender email and app password.

**6. Map Visualization**

* The plot\_on\_map function generates a map with the herd's detected location.
* A marker is added to the map, and it is saved as herd\_map.html.

**7. FastAPI Integration**

* Provides an API for uploading images and detecting herds.
* /detect endpoint processes an uploaded image and returns detections and GPS data.
* /map endpoint generates and serves a map with detected herd locations.

**8. Deployment**

* The system can be deployed using Uvicorn (uvicorn.run(app, host="0.0.0.0", port=8000)).
* FastAPI serves the endpoints for real-time detection and visualization.

**Issues and Recommendations**

1. **Hardcoded API Key Dependency**
   * Issue: API key is expected in an environment variable but is not handled dynamically.
   * Fix: Provide a fallback mechanism or prompt the user to input the key if missing.
2. **Lack of GPS Extraction from Video**
   * Issue: The GPS extraction function works for images but not video frames.
   * Fix: Implement a method to extract metadata from video frames or integrate with a drone telemetry system.
3. **Security Risks in Email Alert System**
   * Issue: Storing plaintext email credentials is a security risk.
   * Fix: Use environment variables or a secure credential manager.
4. **Processing Every 10th Frame**
   * Issue: This may miss important detections.
   * Fix: Optimize frame selection based on motion or object presence rather than fixed intervals.
5. **No Logging Mechanism**
   * Issue: The system does not log errors or detections.
   * Fix: Implement logging to track detections, errors, and API requests for debugging.

**Conclusion**

The Herd Detection System is a well-structured application combining deep learning, geolocation tracking, and web-based deployment. By addressing the suggested improvements, the system can be enhanced for better accuracy, security, and real-time performance.