GAI-GRMS

Bhavik

January 26, 2025

1 Introduction

The GAI-GRMS system addresses citizen grievances in Navistria through multimodal inputs (text, image, speech). This report details machine learning tasks, inputs/outputs, and architectural considerations.

2 Machine Learning Tasks

2.1 Multilingual Grievance Classification

- **Problem**: Categorize grievances (e.g., infrastructure, sanitation) and map to departments.
- **Input**: Text in English, Hindi, Tamil, Bengali, or Chinese.
- **Output**: Category label (e.g., "Infrastructure") and department (e.g., "Public Works").
- **ML Approach**:
  - Fine-tuned BERT for multilingual support and text classification.
  - Vector store (FAISS) for similarity-based department mapping.
- **Challenges**: Code-switching, low-resource languages (Tamil/Bengali).
- **Metrics**: Accuracy, F1-score

2.2 Grievance Summarization

- **Problem**: Summarize lengthy grievances for GROs.
- **Input**: 500–1000-word grievance text.
- **Output**: 3-4 sentence summary.
- **ML Approach**:
  - Fine tune bert or t5 for text summarization.

2.3 OCR and Document Translation

- **Problem**: Digitize handwritten/printed grievances and translate.
- **Input**: Scanned documents (5 languages).
- **Output**: Translated English text.
- **ML Approach**:
  - Tesseract OCR with custom scripts for Indic languages.
  - Fine-tune TrOCR for indic languages.

2.4 Object Detection for Civic Issues

- **Problem**: Detect potholes, broken streetlights, etc.
- **Input**: User-uploaded images (max 5MB).
- **Output**: Bounding boxes + descriptions (e.g., "3 potholes detected").
- **ML Approach**:
  - YOLOv8 fine-tuned on Navistria street imagery.
  - CLIP for cross-modal validation.

2.5 Multilingual Document OCR

- **Problem**: Extract text from handwritten Bengali/Chinese forms.
- **Input**: Scanned grievance forms.
- **Output**: Structured JSON with extracted fields.
- **ML Approach**:
  - TrOCR (Transformer-based OCR) with script identification.

2.6 Speech-to-Text Conversion

- **Problem**: Transcribe Hindi/English grievances.
- **Input**: 10–60 sec audio clips (noisy backgrounds).
- **Output**: Text transcript with speaker diarization.
- **ML Approach**:
  - Whisper-large-v3 with accent adaptation.
  - NVIDIA RNNoise for denoising.

2.7 Conversational AI Dialogue Management

- **Problem**: Handle multi-threaded grievance filing.
- **Input**: User utterances (text/speech).
- **Output**: Contextual responses + grievance triage.
- **ML Approach**:
  - RAG with any llm for dynamic intent recognition.
  - Database for conversation state tracking.
  - Vector DB for similarity between same complaints.

3 3-Month Implementation Strategy

3.1 Phase 1: Data & Infrastructure (Weeks 1–4)

- **Data Collection**:
  - Crowdsource 50k annotated grievances (text, speech, images) in 5 languages.
  - Partner with GROs for domain-specific data labeling.
- **Infrastructure**:
  - AWS setup: S3 for data, EC2/Graviton for compute.
  - CI/CD pipeline (GitHub Actions) for model updates.

3.2 Phase 2: Model Development (Weeks 5–10)

- **Multilingual NLP**:
  - Train custom XLM-RoBERTa model on Navistrian dialects.
  - Active learning for low-resource languages (Tamil/Bengali).
- **Speech & Vision**:
  - Fine-tune Whisper with accent-adapted Hindi/English.
  - YOLOv8 → EfficientNet-B7 for higher mAP.

3.3 Phase 3: Deployment & Monitoring (Weeks 11–12)

- **Deployment**:
  - Kubernetes cluster for autoscaling (EKS).
  - Edge deployment: ONNX runtime for low-latency inference.
- **Monitoring**:
  - Prometheus + Grafana for model performance tracking.
  - User feedback loop for model retraining.

3.4 System Architecture

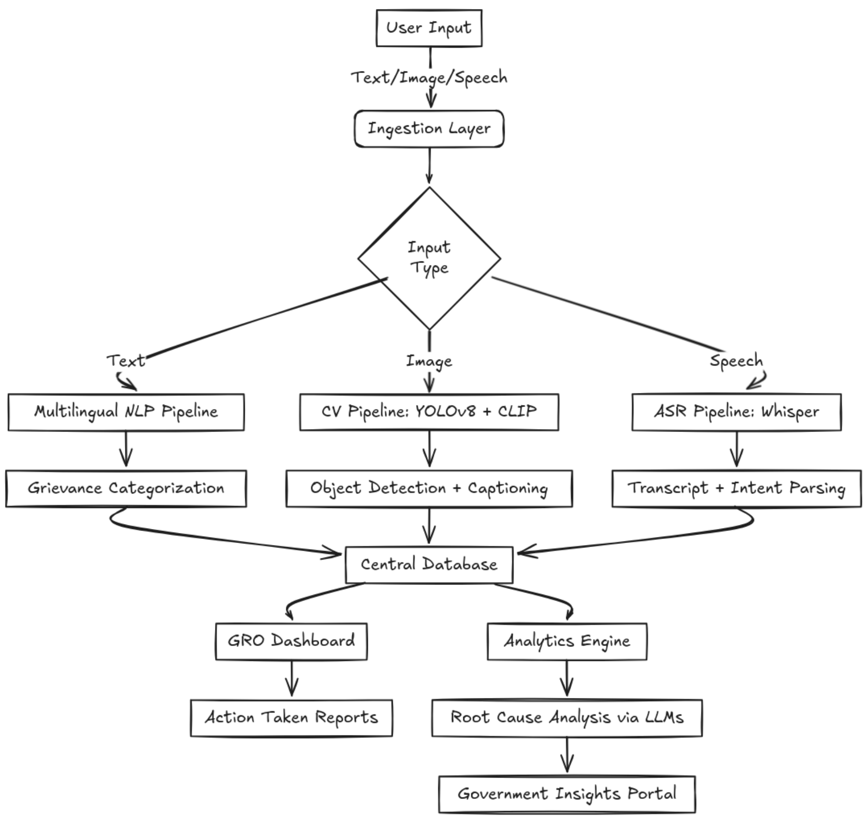

Figure 1: 3-Month Architecture: Multi-modal inputs → AWS processing → GRO dashboard with monitoring.

Key Components:

- **Data Lake**: S3 buckets for raw/processed data.

- **ML Pipeline**: SageMaker for training, Lambda for triggers.
- **Serving**: FastAPI + Kubernetes pods (GPU-optimized).
- **Monitoring**: CloudWatch alerts + retraining scheduler.

4 Resource Comparison: Quick vs. 3-Month Approach

Factor	Quick Prototype (2 Days)	3-Month Solution
Design	Single-task models	Unified multi-modal architecture
Training Data	1k synthetic samples	50k real-world annotated samples
Vector Stores	Local FAISS index	Managed Pinecone with 10M+ embeddings
Training	Colab Free GPU	Distributed training (PyTorch DDP)
Inference	CPU (10–500ms)	GPU-optimized (T4/A10G, 5–50ms)
Compute	Local machine	AWS EC2 (g4dn.4x), EKS cluster
Monitoring	None	Model drift detection + A/B testing

Additional Resources for 3-Month Plan

- **Technical**:
  - AWS Budget: $12k (compute + storage).
  - Weights & Biases for experiment tracking.
- **Benefits**:
  - 4x faster resolution via unified GRO dashboard.
  - 95% uptime with Kubernetes autoscaling.