# 🤖 GeoAI Assistant Pro

Enterprise-Grade AI-Powered Geospatial Assistant for QGIS

**Version 2.0.0**

QGIS 3.0+ | Python 3.9+ | MIT License

Technical Documentation Report

Generated: December 9, 2025

# Table of Contents

# 🌟 Overview

**GeoAI Assistant Pro** is a revolutionary QGIS plugin that brings the power of Artificial Intelligence directly into your geospatial workflows. Whether you're a GIS analyst, data scientist, or developer, this plugin transforms how you interact with spatial data.

### 🎯 Key Capabilities

- **Multi-Provider AI:**

  Choose from 6+ LLM providers with 50+ models
- **Visual Intelligence:**

  Convert QGIS Model Builder screenshots to executable code
- **Natural Language SQL:**

  Describe what you want in plain English, get SQL code instantly
- **Smart Error Fixing:**

  AI automatically detects and fixes SQL errors
- **Intelligent Suggestions:**

  Get AI-powered recommendations for your projects
- **Enterprise Features:**

  Query history, batch processing, templates, analytics

# What Makes It Special?

GeoAI Assistant Pro stands out with its comprehensive AI integration, supporting multiple providers and offering seamless integration with QGIS workflows. The plugin combines the power of modern LLMs with geospatial expertise, making complex spatial operations accessible through natural language.

# ✨ Features

## 🤖 LLM Handler - Multi-Provider AI Support

Support for **6 major AI providers** with **50+ models**:

| Provider | Models | Use Case |
|---|---|---|
| **OpenAI** | GPT-4, GPT-4o, GPT-3.5-turbo | Best quality, general purpose |
| **Anthropic** | Claude 3.5 Sonnet, Claude 3 Opus | Long context, detailed analysis |
| **Google** | Gemini 2.5 Pro, Gemini 2.5 Flash | Fast, cost-effective |
| **Ollama** | phi3, mistral, llama, codellama | Local, private, free |
| **OpenRouter** | 100+ models | Access to all major models |
| **HuggingFace** | Open-source models | Community models |

✅ **Key Features:**

- Automatic retry with exponential backoff
- Quota error handling
- Safety settings management
- Model validation
- Fallback mechanisms

# 🖼️ Model Converter - Image-to-Code Magic

Convert QGIS Model Builder screenshots to executable code using Azure Computer Vision and LLM analysis.

### 📸 Screenshot Analysis

Upload screenshots of your Model Builder workflow for AI analysis.

### 🤖 AI Processing

Azure Computer Vision + LLM analyze the image and generate code.

### 💻 Code Generation

Get clean SQL or Python code ready for execution.

### ✅ Direct Execution

Execute generated code directly in QGIS.

# 🔍 SQL Generator - Natural Language to SQL

Transform your thoughts into SQL queries with context-aware generation.

**Example Query:**

```
"Find all buildings within 500 meters of parks"

Generated SQL:
SELECT b.*
FROM buildings b, parks p
WHERE ST_DWithin(b.geom::geography, p.geom::geography, 500)
```

# 🔧 Error Fixer - AI-Powered SQL Correction

Never struggle with SQL errors again. The AI automatically detects and fixes:

- Syntax errors (`FORM` → `FROM`)
- Missing keywords
- Wrong table/column names
- Type mismatches
- Geometry column issues (`geometry` → `geom`)
- Quote problems

- Logical errors

# 🚀 Installation

## Prerequisites

- ✅ **QGIS 3.0 or higher**
- ✅ **Python 3.9+** (included with QGIS)
- ✅ **Internet connection** (for cloud-based LLM providers)
- ✅ **Azure account** (optional, for Model Converter)

## Step 1: Download Plugin

1 **Navigate to QGIS plugins directory:**

`~/Library/Application Support/QGIS/QGIS3/profiles/default/python/plugins/`

2 **Clone or download** the plugin repository

# Step 2: Enable Plugin

1 **Restart QGIS**

2 Go to `Plugins → Manage and Install Plugins`

3 Search for **"GeoAI Assistant Pro"**

4 ✅ Check the box to enable

# Step 3: Install Dependencies

```
# Azure Computer Vision
/path/to/qgis/python3 -m pip install azure-cognitiveservices-visio

# LLM Providers (optional)
pip install openai anthropic google-generativeai
```

⚠️ **Important:**

Copy `.env.example` to `.env` and fill in your API keys before using the plugin.

# ⚡ Quick Start Guide

## Example 1: Generate SQL from Natural Language

1. Open Plugin → Click GeoAI Assistant Pro icon
2. Go to **SQL Generator** tab
3. Type: `"Show me all buildings with area greater than 1000"`
4. Select Model from dropdown
5. Click **Generate SQL**
6. Click **Execute** → See results

# Example 2: Convert Image to Code

1. Go to **Model Converter** tab
2. Click **Browse** → Select screenshot
3. Select Type: SQL or Python
4. Click **Convert to Code**
5. Review and use generated code

# ⚙️ Configuration

## Environment Variables (.env file)

```
# Azure Computer Vision
AZURE_VISION_ENDPOINT=https://your-endpoint.cognitiveservices.azur
AZURE_VISION_SUBSCRIPTION_KEY=your-subscription-key-here

# LLM Provider API Keys
OPENAI_API_KEY=sk-your-key-here
ANTHROPIC_API_KEY=sk-ant-your-key-here
GOOGLE_API_KEY=your-google-api-key-here

# Database Configuration (Optional)
DB_HOST=localhost
DB_PORT=5432
DB_NAME=postgis_db
DB_USER=postgres
DB_PASSWORD=your-password

# Ollama Configuration
OLLAMA_BASE_URL=http://localhost:11434
```

💡 **Tip:**

Copy `.env.example` to `.env` and fill in your values. Never commit `.env` to version control!

# 📚 Usage Guide

## SQL Generator

**Workflow:**

1. Describe your query in natural language
2. Select LLM model
3. Click "Generate SQL"
4. Review and execute
5. View results

## Model Converter

**Best Practices:**

- Use high-quality screenshots
- Ensure Model Builder is clearly visible
- Include all workflow steps
- Use PNG format for best results

# Error Fixer

### When to Use:

- SQL syntax errors
- Column name issues
- Type mismatches
- Geometry column problems

# 🏗️ Architecture

## Project Structure

```
GeoAI_Assistant_Pro/
├── modules/          # Core modules
│   ├── llm_handler.py
│   ├── sql_executor.py
│   ├── image_processor.py
│   └── error_fixer.py
├── core/            # Domain layer
├── services/         # Business logic
├── infrastructure/   # Infrastructure
└── ui/              # User interface
```

## Design Patterns

- **Service-Oriented Architecture:** Separated business logic
- **Component-Based UI:** Modular, reusable components
- **Provider Pattern:** Pluggable LLM providers
- **Factory Pattern:** Dynamic initialization

- **Observer Pattern:** Event-driven updates

# 🔧 Troubleshooting

## Common Issues

### ❌ Plugin Won't Load

**Solutions:**

- Check Python syntax errors in Log Messages
- Verify dependencies are installed
- Clear `__pycache__` folders
- Restart QGIS completely

### ❌ Azure Computer Vision Not Working

**Solutions:**

- Verify credentials in Settings
- Check `.env` file exists
- Verify endpoint is accessible
- Plugin will auto-fallback to LLM

## ❌ SQL Generation Fails

### Solutions:

- Check LLM provider is configured
- Verify API keys are correct
- Check internet connection
- Try different model

**GeoAI Assistant Pro** - Technical Documentation Report

Version 2.0.0 | MIT License | Generated: December 9, 2025

For more information, visit the project repository or contact support.