Crime Scene Analysis System

Graph-Based Forensic Tool

Overview

This Java application implements a graph-based system for analyzing crime scene evidence, combining:

1. **Automated Object Classification** using Region Adjacency Graphs (RAGs)
2. **Optimal Path Reconstruction** via grid-based A* pathfinding

Built for Computer Science 3A Mini Project, the system addresses South Africa's forensic analysis challenges by automating evidence processing.

Features

1. Object Classification Module

- SLIC superpixel segmentation
- Region Adjacency Graph construction
- k-NN classification of:
    - Weapons (guns, knives)
    - Tools (crowbars, hammers)
    - Blood stains
- Bounding box visualization

2. Pathfinding Module

- Floor plan grid conversion
- Walkability analysis
- A* pathfinding with Euclidean heuristic
- 2D path visualization

Technical Specifications

Core Data Structures:

- Custom Graph ADT implementation
- SuperPixel nodes with texture/color features

- Weighted edges for similarity/path costs

Algorithms:

- SLIC superpixel segmentation
- k-Nearest Neighbors classification
- A* search algorithm

GUI:

- JavaFX interface
- Image input/output handling
- Interactive visualization

System Requirements

- Java 17 or higher
- JavaFX SDK
- Minimum 4GB RAM (8GB recommended for large images)

Installation

1. From source:

javac --module-path /path/to/javafx-sdk/lib --add-modules javafx.controls src/* java --module-path /path/to/javafx-sdk/lib --add-modules javafx.controls crimescene.CrimeSceneAnalysisSystem

---

Usage

To Run the Project:

Open the docs folder, then click unme.bat.

Object Classification

1. Click "Load Image" to select crime scene photo
2. Adjust superpixel parameters (optional)
3. Click "Classify Objects"
4. View results with color-coded bounding boxes

Path Finding

1. Click "Load Floor Plan"
2. Set start/end points
3. Click "Find Path"
4. View optimal path overlay

Performance Metrics

Operation	Average Time (1024x768 image)
Superpixel generation	1.2s
RAG construction	0.8s
Object classification	0.5s
Pathfinding (20x20 grid)	0.1s

Tested on Intel i7-11800H, 16GB RAM

Sample Inputs

Example images available in /data:

- crime_scene_1.jpg Weapon classification demo
- loor_plan.png Pathfinding demo

Documentation

- Full Javadoc available in /docs

Credits

Developed by:

- Kagiso Maja (221176553)
- Bonginkosi Mdakane (221113865)
- Sifiso Kau (221100291)