# Anti-personnel Landmine Detection

## Table of Contents

# 1. Introduction

The development of the APL (Anti-personnel Landmines) detection system involved the following steps:

1. **Image Selection and ROI Segmentation:**
   - 136 images with the best thermal contrast were selected from a set of 198 images acquired at a height of 1 m.
   - From each of these images, 8 Regions of Interest (ROI) with dimensions of 16 × 16 pixels were manually segmented: 4 corresponding to regions with buried APL and 4 to clean areas, totalling 1088 ROIs.
2. **Feature Extraction and Normalization:**
   - For each ROI, 22 characteristics were extracted, including statistical moments (mean, standard deviation, kurtosis, asymmetry), maximum and minimum intensities, and texture characteristics (energy, contrast, correlation, and homogeneity) of co-occurrence matrices at 0◦, 45◦, 90◦, and 135◦.
   - The extracted features were normalized in the range from 0 to 1.
3. **Feature Selection:**
   - The Lilliefors test was applied to evaluate each feature individually by class to determine if they have Gaussian probability density functions.
   - All characteristics were found to be Gaussian, so the Fisher Discriminant Ratio (FDR) criterion and scalar selection technique were used to select the most discriminant features: mean of intensities, minimum and maximum values, and co-occurrence matrix energy at 90◦.
4. **Classifier Training:**
   - An MLP (Multi-Layer Perceptron) network with 15 neurons in the hidden layer and a sigmoidal activation function was selected as the classification method.
   - The training dataset consisted of 200 ROIs of the "mine" class and 101 ROIs of the "non-mine" class, normalized to the range [0, 1].
   - The training achieved a detection success percentage of 93.33%.
5. **Classifier Operation:**
   - The detection.py script implements a sliding window approach to perform a complete scan on the input image to be classified.
   - The script loads the pre-trained MLP (Multi-Layer Perceptron) classifier from the **mlp_classifier.pkl** file.
   - Functions are defined to extract features from regions of interest (ROIs) in the thermal image.
   - These features include statistical moments, histogram features, and texture features extracted from GLCM (Gray-Level Co-occurrence Matrix).
   - The script utilizes a sliding window technique to scan the input image with a specified step size and window size.
   - Each ROI extracted from the image is classified using the trained classifier to determine the presence of a landmine.
   - After classification, a binary image is generated based on the classification results.
   - The script accumulates classification results from multiple ROIs to generate the binary image.

- Thresholding criteria are applied to the binary image to identify potential landmine locations.
- The script utilizes OpenCV to visualize the detection results.
- Contours are extracted from the binary image to outline the detected landmine areas.
- These contours are then drawn on the original thermal image to visualize the detected areas.
- The script provides a file dialog for the user to select the input thermal image.
- Once an image is selected, the detection process is initiated, and the results are displayed in a new window.

This process resulted in the development of an effective APL detection system, achieving high detection accuracy in both training and operational stages.

## 2. Requirements

- Python 3.x
- OpenCV (cv2)
- NumPy
- scikit-learn
- tkinter (for file dialogs)
- joblib

## 3. Execution

**Execution Steps:**

1. Ensure all required Python packages are installed. You can install all the packaged with the following command on the command line interface with the directory open - run - ```pip install -r requirements.txt```
2. Download the provided code files and ensure they are saved in the same directory.
3. Execute the **detection.py** script. This script contains the main code for landmine detection.
   run - ```python detection.py```
4. Follow the prompts or instructions provided by the script.
5. When prompted, select an image containing the area to be scanned for landmines.
6. Wait for the script to process the image and display the detection results.
7. Analyze the detection results displayed in the window. Landmine locations will be highlighted in green contours on the original image.
8. Close the image window to exit the script.

## 4. Code Overview

he provided code consists of several components:

1. **Feature Extraction**: Functions to extract features from regions of interest (ROIs) in images.
2. **Data Preparation**: Loading images, extracting features, creating labels, and splitting the data into training and testing sets.
3. **Model Training**: Training a Multi-Layer Perceptron (MLP) classifier using the extracted features and labels.
4. **Model Evaluation**: Evaluating the trained classifier's accuracy on the test set.
5. **Model Saving**: Saving the trained classifier to a file using joblib.
6. **Detection Process**: Loading the trained classifier and using sliding windows to detect landmines in images.

## 5. Feature Extraction

The feature extraction process involves several steps:

1. **Statistical Moments**: Calculate mean, standard deviation, kurtosis, and skewness of the ROI using NumPy.
2. **Histogram**: Calculate the histogram of the ROI using OpenCV's `cv2.calcHist` function and normalize it.
3. **GLCM Features**: Calculate Gray-Level Co-occurrence Matrix (GLCM) features such as contrast, energy, and homogeneity from the histogram.

These features are then concatenated to form a feature vector for each ROI.