

Ryan Raschkovsky

04/27/2025

OSINT Pattern Recognition: Slip (Street Level Image Processing)

Extract Bollard Features:

1. Get raw data to show object boundaries
 - Possibly enhance to better recognize color or differences in arrangements or color scheme - (adaptive histogram equalization)
 - Use Gaussian blur to reduce noise
2. Detect bollards using object detection (Faster R-CNN)
 - Apply edge detection (Canny or Sobel)
 - Use image processing techniques that help you find and separate upright objects (Morphological Operations that can be used to isolate vertical structures)
 - Further filtering for precise image recognition
 - Also for additional contrast for image detection: Contour detection
3. Work on extracting pattern features
 - Measure bollard height-to-width ratios
 - Record bollard top shapes (flat, rounded, decorative) **Distinctive across regions**
 - Document material appearance (concrete, metal, plastic)
 - Detect color patterns or reflective elements
 - Calculate alignment patterns (straight line, curved, staggered)
4. Return Vector layout of bollard arrangement for later analysis

To Analyze Central Asian Telephone Pole:

1. Detect telephone poles
 - Use vertical line detection for poles
 - Apply template matching with common Central Asian pole pattern
2. Extract distinctive regional features (examples given)
 - Kazakhstan: Measure height-to-width ratio (typically taller)
 - Uzbekistan: Count number and arrangement of crossbeams (often X-pattern)
 - Tajikistan: Detect specific insulator configurations (ceramic clusters)
 - Kyrgyzstan: Identify mountain-specific reinforcement patterns
 - Turkmenistan: Detect distinctive pole tops and desert-specific modifications
3. Analyze surrounding power line configurations
 - Calculate angles between lines
 - Count number of wires
 - Measure height of lowest wire from ground

4. Return feature vector of pole characteristics

Further Machine Learning algorithms to Classify Country From Infrastructure:

1. bollardFeatures = Extraction from Bollard features given (*image*)
2. poleFeatures = Analysis from Central Asian Telephone Poles given (*image*)
3. Load pre-trained country-specific models
 - bollardModel = loadBollardClassificationModel()
 - poleModel = loadPoleClassificationModel()
4. Generate prediction scores
 - bollardScores = bollardModel.predict(bollardFeatures)
 - poleScores = poleModel.predict(poleFeatures)
5. Apply weighted ensemble method
 - If image is urban street with bollards: weight bollardScores higher
 - If image is roadside/rural with poles: weight poleScores higher
 - Otherwise use balanced weighting
6. Calculate confidence intervals
 - Implement bootstrap resampling on feature vectors
 - (Estimates the sample distribution through repeated sampling of a specific population; for a range of predictions)
7. Calculate confidence bounds and return top 3 country predictions with confidence scores

// Key bollard patterns by country (Source: <https://plonkit.net>)

```
bollardPatterns = {  
  
    Netherlands: {"spacing": "wide", "shape": "cylindrical", "color": "red/white alternating"},  
  
    UK: {"spacing": "medium", "shape": "round-topped", "material": "cast iron"},  
  
    Japan: {"spacing": "tight", "shape": "square/rectangular", "color": "yellow/black"},  
  
    France: {"spacing": "medium", "shape": "conical", "material": "concrete"},  
  
    // Put other countries here during implementation...  
  
}  
  
// Central Asian telephone pole characteristics  
  
centralAsianPoles = {  
  
    "Kazakhstan": {
```

```
"height": "tall",  
  
"crossbeam": "perpendicular double",  
  
"insulators": "white ceramic clusters"  
},  
  
"Uzbekistan": {  
  
  "height": "medium",  
  
  "crossbeam": "X-pattern",  
  
  "insulators": "spaced evenly in rows"  
},  
  
"Tajikistan": {  
  
  "height": "variable",  
  
  "crossbeam": "single T-top",  
  
  "insulators": "mixed ceramic/glass"  
},  
  
"Kyrgyzstan": {  
  
  "height": "reinforced base",  
  
  "crossbeam": "angled braces",  
  
  "insulators": "compact arrangement"  
},  
  
"Turkmenistan": {  
  
  "height": "shorter",  
  
  "crossbeam": "triangular pattern",  
  
  "insulators": "desert-specific spacing"
```

