

Air Pollution Source Detection System

1. Core Concept

This system uses YOLOv5 object detection to identify potential pollution sources in images.

2. Technical Implementation

2.1 Object Detection Pipeline

```
# Initialize YOLOv5
model = torch.hub.load("ultralytics/yolov5", "yolov5s", pretrained=True)

# Process image
results = model(image)
detections_df = results.pandas().xyxy[0]

# Filter pollution sources
filtered_df = filter_pollution_sources(detections_df)
```

2.2 Detected Source Categories

The system detects these objects:

- Cars
- Trucks
- Buses
- Motorcycles
- Trains
- Boats
- Airplanes

2.3 Visualization Features

- Red overlay for detected areas (30% transparency)
- Bounding boxes with white borders
- Confidence scores
- Class labels

3. Core Functions

3.1 Image Loading

```
def load_image(image_path):  
    # Loads image and converts BGR to RGB  
    return img_rgb
```

3.2 Pollution Source Detection

```
def filter_pollution_sources(detections_df):  
    pollution_sources = [  
        "car", "truck", "bus", "motorcycle",  
        "train", "boat", "airplane"  
    ]  
    return filtered_df
```

3.3 Visualization

```
def create_pollution_mask(img, detections):  
    # Creates red overlay mask  
    return mask  
  
def display_detections(img, detections, class_names):  
    # Draws bounding boxes and overlays  
    # Saves output as PNG
```

4. Output Specifications

- Resolution: 300 DPI
- Format: PNG
- Figure size: 20x12 inches
- Transparent overlays (30% opacity)

5. Technical Requirements

- Python with PyTorch
- YOLOv5
- OpenCV

- Matplotlib