# Autonomous Vehicle Surveillance Through Fuzzy C-Means Segmentation and DeepSORT on Aerial Images

## Authors

* Asifa Mehmood Qureshi (Air University, Pakistan)
* Moneerah Alotaibi (Shaqra University, Saudi Arabia)
* Sultan Refa Alotaibi (Shaqra University, Saudi Arabia)
* Dina Abdulaziz AlHammadi (Princess Nourah bint Abdulrahman University, Saudi Arabia)
* Muhammad Asif Jamal (Chosun University, Korea)
* Ahmad Jalal (Air University, Pakistan)
* Bumshik Lee (Chosun University, Korea)

**Corresponding Author:** Bumshik Lee  
**Contact:** bslee@chosun.ac.kr

## Overview

This project presents a system for detecting and tracking vehicles in aerial imagery. It combines Fuzzy C-Means (FCM) clustering for image segmentation with YOLOv4 for vehicle detection and DeepSORT for multi-vehicle tracking. Designed for aerial surveillance applications, it has been tested on UAVDT and KIT-AIS datasets, achieving high precision and recall in vehicle tracking.

### Key Features

1. **Hybrid Detection and Tracking**: Leverages FCM and YOLOv4 for effective vehicle detection.
2. **Advanced Multi-Vehicle Tracking**: Uses DeepSORT with Kalman filters and SURF for ID assignment and recovery, providing robust tracking across frames.
3. **Benchmark Testing**: Validated on the UAVDT and KIT-AIS datasets, demonstrating competitive precision and recall.

## System Components

The system operates in three main phases:

1. **Image Segmentation**: Uses FCM clustering to isolate vehicles from background elements.
2. **Vehicle Detection**: YOLOv4 identifies vehicles within segmented areas, focusing on small objects in complex scenes.
3. **Vehicle Tracking**: DeepSORT tracks vehicles across frames with the help of Kalman filters and appearance descriptors, while SURF is used for ID assignment and recovery.

### Datasets

* **UAVDT Dataset**: Contains over 10 hours of aerial video footage from urban environments.
* **KIT-AIS Dataset**: Includes aerial images with five vehicle classes.

## Prerequisites

To run this code, ensure you have:

* **Python 3.7** or later.
* Necessary Python libraries (see below for installation).

Required libraries:

* + tensorflow
  + opencv-python
  + scikit-image
  + scikit-learn
  + numpy
  + matplotlib

### Installation of Dependencies

1. Install the dependencies listed above by running:

## Implementation

Follow these steps to implement the vehicle detection and tracking system:

### 1. Clone the Repository

Download or clone the repository containing **model.ipynb** and place it in your preferred working directory:

### 2. Dataset Preparation

* Download the **UAVDT** and **KIT-AIS** datasets. Ensure you have them in a structured format accessible to the notebook:
* Unzip and place the datasets in the data/ directory within the project folder.
* Inside **model.ipynb**, modify the dataset paths as necessary.

### 3. Run the Notebook

Open **model.ipynb** in Jupyter Notebook or any compatible environment, and execute each cell step-by-step:

* **Segmentation Phase**: The first section applies FCM clustering to segment vehicles from the background. Run these cells to preprocess the images and obtain segmented results.
* **Detection Phase**: Next, the notebook utilizes YOLOv4 to detect vehicles in the segmented images. Ensure the YOLOv4 weights are correctly downloaded and configured in the notebook.
* **Tracking Phase**: Finally, the DeepSORT tracking cells track vehicles across frames. Run these cells to track multiple vehicles and maintain IDs across frames.
* **Detailed Execution**

For each of the main phases, the notebook provides code comments to guide you through parameter adjustments:

* **FCM Clustering**: Set the clustering parameters to refine segmentation accuracy based on the background complexity.
* **YOLOv4**: Adjust detection confidence thresholds if needed. Paths to model weights are specified here.
* **DeepSORT Tracking**: Customize parameters such as Kalman filter tuning to enhance tracking accuracy.

### 4. View and Analyze Results

* **Visualization**: The notebook generates visualizations at each phase, allowing you to observe the segmentation, detection, and tracking processes.
* **Performance Metrics**: Precision and recall metrics are calculated, providing insight into the model’s accuracy on the UAVDT and KIT-AIS datasets.

## Computing Environment

The experiments were conducted on a system with the following specifications:

* OS: Windows 10
* Processor: Intel Core i5-7200U (2.7 GHz)
* RAM: 8 GB

## Results and Evaluation

The code in **model.ipynb** demonstrates the performance of the proposed system on UAVDT and KIT-AIS datasets, showcasing robust vehicle tracking capabilities in aerial imagery with high precision and recall.

## Code and Data Availability

The code and details on obtaining the datasets are provided with this README file. Dataset download links and paths for data preparation are also included within **model.ipynb** for easy access.