

AI-Powered Saudi License Plate Recognition System

A Multi-Stage Pipeline using YOLO and Logic-Based Post-Processing

Hassan Algamdi

Omar Talib

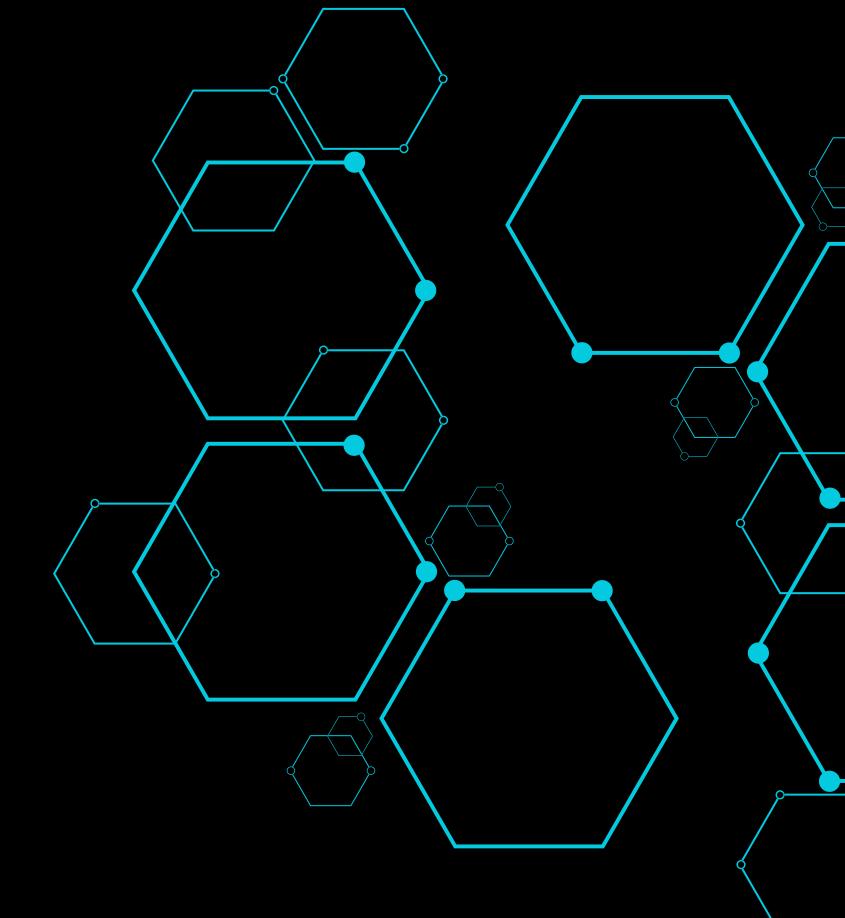
Hoyon Balkhi

Sultan Alasami



The Problem

- **Context:** Automated License Plate Recognition (ALPR) is critical for traffic management, security, and parking systems in Saudi Arabia.
- **The Challenge:** Saudi plates present unique difficulties compared to standard Western plates:
 1. **Bilingual Layout:** Contains both Arabic and English text.
 2. **Complex Layout:** Includes emblems (swords & palm) and vertical "KSA" strips that confuse standard models.
 3. **Environmental Factors:** Dust, harsh sunlight/shadows, and varying camera angles (tilt).



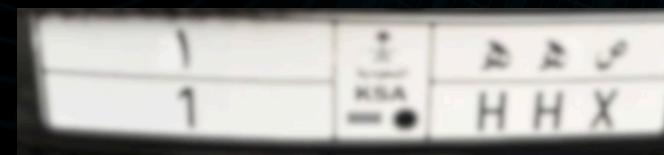
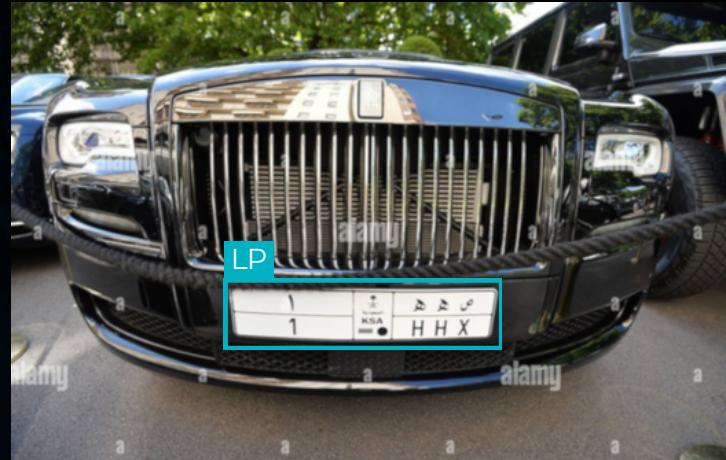
Evolution of Our Approach

Initial Strategy: Standard Object Detection (find plate)
→ Image Cropping → Image Upscaling → Image Segmentation -> Optical Character Recognition (OCR) to "read" the text.

Why We Abandoned It:

- **OCR Limitations:** Generic OCR models (EasyOCR/Tesseract) struggled with the specific Saudi font and calligraphy.
- **Segmentation Failures:** Slight tilts caused crop lines to slice letters (e.g., turning a "Z" into an "L").
- **Noise:** OCR frequently misread the "KSA" logo and emblem as random characters.

First Approach Pipeline



Optical Character
Recognition

Arabic Part: ١٠٠
English Part: 1 HHX

Final System Architecture

Stage 1

- Plate Detection: A YOLO model finds the license plate on the vehicle.

Stage 2

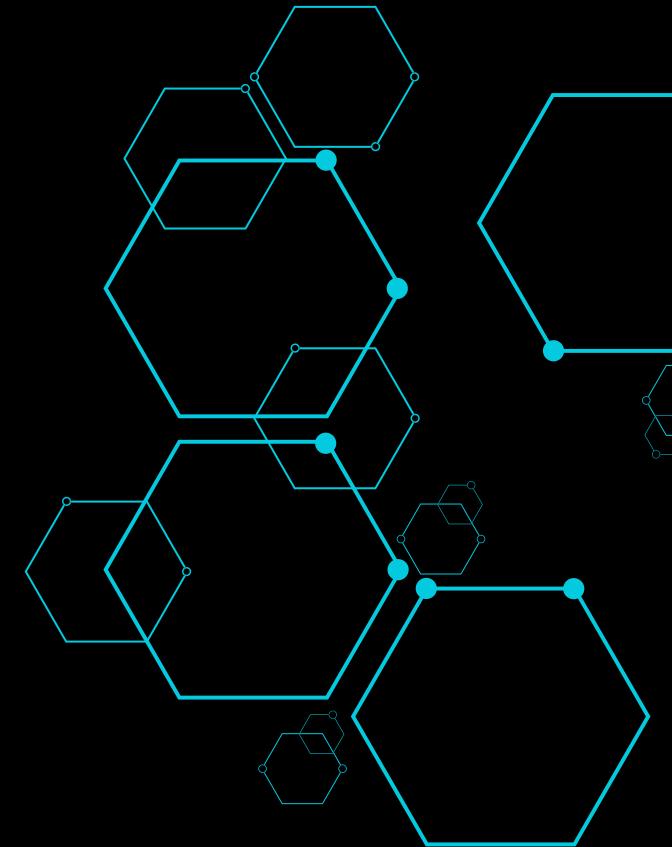
- Crop: Extract the plate image.

Stage 3

- Character Detection: A second specialized YOLO model detects individual characters (A, 9, K) inside the plate.

Stage 4

- Filter out noise (logos/Arabic script) using coordinate geometry (y-axis filtering).
- Sort detected characters from left to right.



Second Approach Pipeline

Standard Object
Detection



Image Cropping



Charachters
Detection



Arabic Part: ١٠٠
English Part: 1 HHX

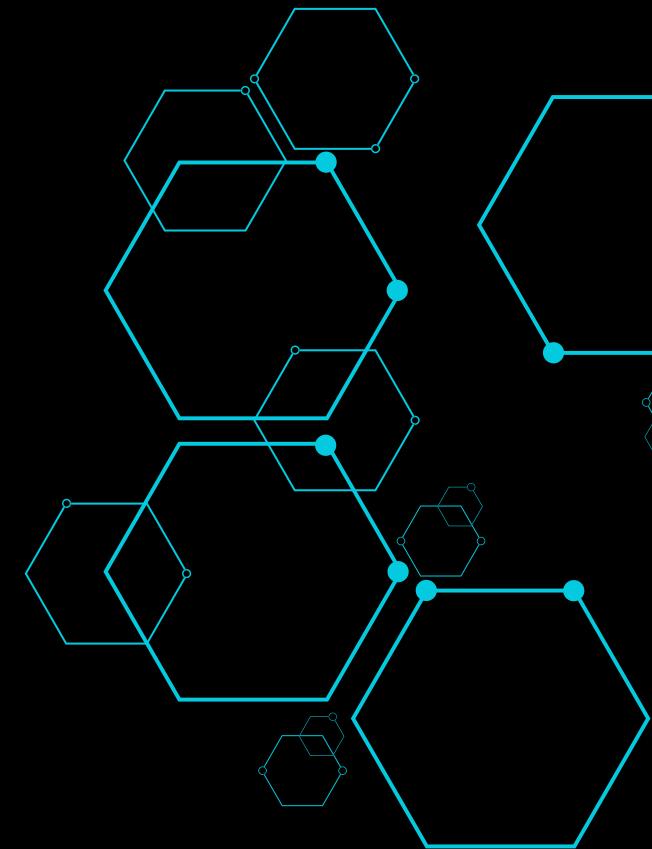
Data & Training Strategy

Dataset 1: Plate Detection

- Size: ~4,500 images with annotated plates.
- Purpose: Trained YOLO to locate license plates in different environments.

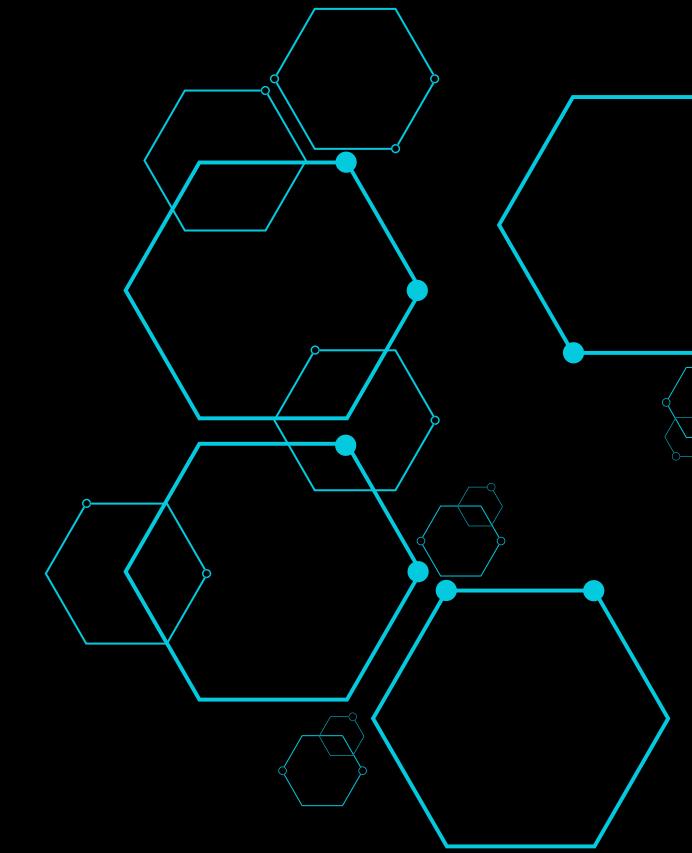
Dataset 2: Character Recognition

- Size: ~1,300 images with annotated characters.
- Purpose: Trained another YOLO to detect and read each character in the plate



Future Work

- **Real-Time Video:** Optimize the pipeline to run at 30 FPS on video streams.
- **Deployment:** Deploy the model to a Raspberry Pi or Jetson Nano for use in actual parking gates.
- **Mobile App:** Wrap the Python backend in a Flutter/React Native app for traffic officers.



Thank you for listening



Q&A