Paper Title:
An Intelligent Automatic Number Plate Recognition System Based on Computer Vision and Edge Computing

Paper Link:
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9945787

1 Summary:

1.1 Introduction

The research paper describes a traffic monitoring system that analyses copious volumes of data from still and moving camera images. The information is used to track traffic collisions, identify number plates, and manage traffic flows. Usually, data centres receive this data for processing and analysis, which might impact operator decision-making speed and use less network capacity.

The edge computing paradigm is the way the researchers suggest solving this issue. This entails positioning edge nodes close to the cameras and using these edge devices to preprocess data. Computer vision techniques are employed to preprocess data at the edge device, which in this case is the Raspberry Pi.

1.2 Related Works

The related works section of the research paper discusses the main stages of number plate recognition: Number Plate Localization (NPL), Character Segmentation (CS), and Optical Character Recognition (OCR). The first stage involves image binarization or grayscale under adverse weather conditions or low light. The number plate is then localized and segmented through Connected Component Analysis (CCA) and ratio analysis. Finally, character recognition is achieved through various methods such as Support Vector Classifier (linear, polygonal, rbf), k Nearest Neighbor (KNN), Extra Tree Classifier, Logistic Regression (LR), Random Forest (RF) and Support Vector Classifier (SVC) plus KNN.

1.3 Methodology

The methodology section of the research paper discusses the use of a Raspberry Pi 4 Model B with an OmniVision OV5647 camera as an edge device for number plate recognition based on computer vision algorithms. The MQTT protocol, based on Node Red and MQTTBox, is used to organize messages between the publisher (Raspberry Pi) and subscribers (TMC operators).

The researchers installed Node Red on the Raspberry Pi to organize the data flow and visual programming for IoT devices, and also configured the MQTTBox message broker on the side of the client. The recommended operating system for the Raspberry Pi is Raspberry Pi OS (Raspbian), but for the server, any compatible operating system with a message broker based

on the Linux core or Windows can be used. In this case, Windows and MQTTBox message brokers have been chosen.

1.4 Conclusion

This approach by the researchers to the given problem helps to optimize network traffic and offload the computing power of the data center. The use of Raspberry Pi and data transmission as text reduces the cost of the technical solution compared to field-programmable gate arrays (FPGAs) and personal computers. The method also allows engineers to increase the number of connected IoT devices to the network due to the minimal impact on network traffic and keep most of the data confidential.