The University of North Carolina at Charlotte

School of Data Science - Spring 2024

DSBA 6190

Optimizing Parking Efficiency: Easy LPR1's Automated License Plate Recognition System Development and Deployment Strategy

Names of Team Members:

Bedilu Jebena

Hillary Cheruiyot

Dezmon Patten

Spring 2024

Table of Content:

1.Introduction

With the increasing challenges surrounding parking service efficiency and security, Easy LPR1 company seeks to implement a cutting-edge License Plate Recognition (LPR) system using advanced computer vision techniques. The main objective of this project is to develop a real-time LPR system capable of seamlessly and accurately identifying license plates from images and videos, in doing so revolutionize the traditional gate attendance process by eliminating the need for manual gate attendance.

The company's current practice of stationing personnel at the gate for parking access control poses some challenges. The company is experiencing increased operational costs, potential errors in manual identification, and limitations in data collection for customer analytics. By transitioning to an automated system, the company can mitigate these issues and uncover several key benefits.

Firstly, the implementation of an LPR system will reduce staffing costs associated with gate attendance, allowing the Easy LPR1 company to allocate resources to other areas of operation. Additionally, the automation of the gate process promises to increase the overall customer experience by facilitating quicker entry and exit times, enabling better data collection for analysis, and finally increasing security through accurate identification.

While this is an exciting upgrade for the company, the successful development of the system does not come without its challenges. Technical complexities, issues with integrating with the existing infrastructure, privacy considerations, and potential resistance due to financial costs are many factors that must be addressed throughout the project's development.

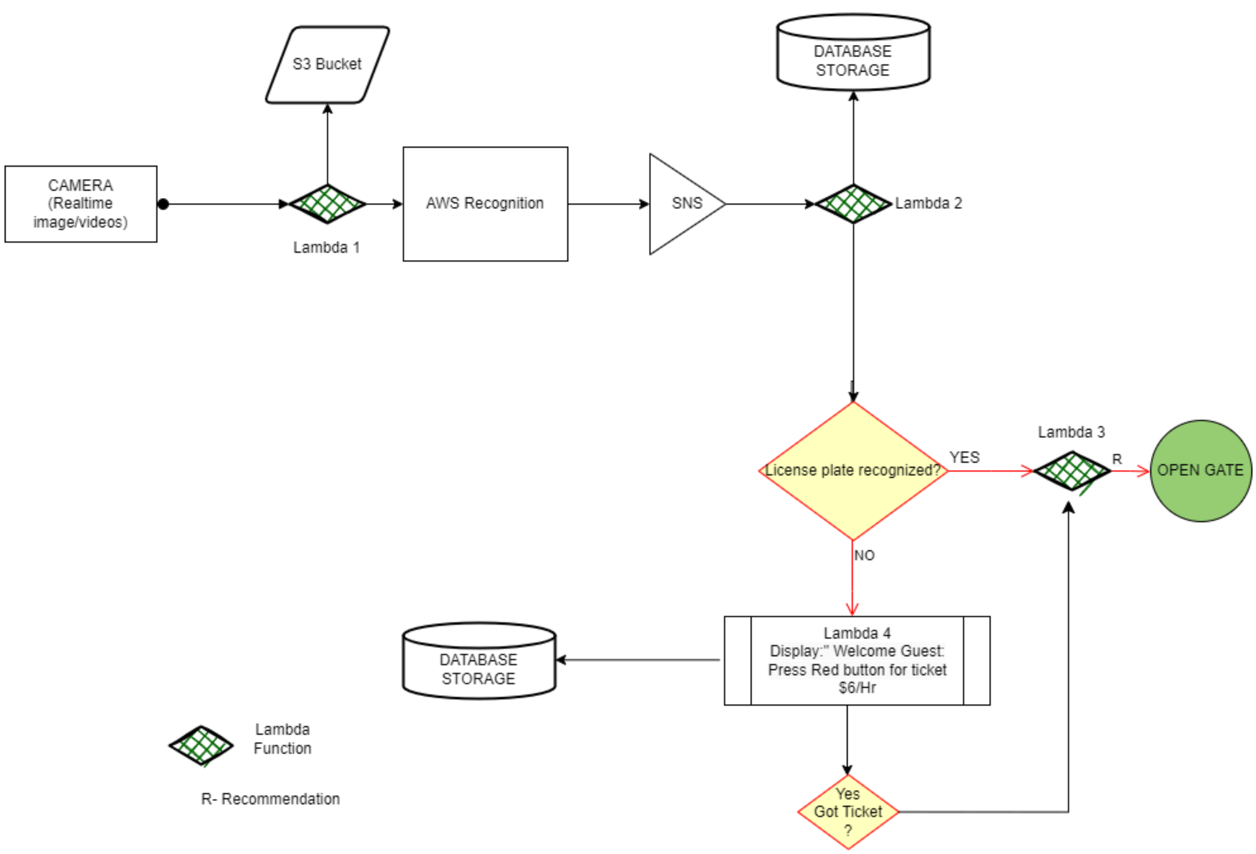
This report will provide a comprehensive overview of the development and deployment of the LPR system for the Easy LPR1 company. It includes details on the methodology employed, the overall architecture of the system along with its implementation strategies. This report will highlight the initial proposed system along with a detailed plan on how this service will be improved over time. Moreover, we will go over the potential cost estimations of implementing this cloud based system. By doing this we aim to demonstrate the effectiveness of using the cloud-based LPR system to optimize parking service operations and security.

2.Planned Multiphase Approach

2.1. Phase 1 Approach

The license plate recognition system is structured around AWS Rekognition, an image and video analysis service. Initially, images or video streams are captured from strategically positioned cameras, acting as primary data sources. These raw inputs undergo preprocessing steps, including resizing to standard dimensions, noise reduction, and other enhancements to optimize recognition accuracy.

Figure 1: Easy LPR 1 Parking Validation Computer vision Flow Chart



Once preprocessing is complete, the processed data is fed into AWS Rekognition for license plate detection. Using its built-in capabilities, AWS Rekognition's algorithms automatically identify regions of interest within the images or videos, specifically targeting license plates. This step is crucial as it isolates the pertinent information needed for further analysis, eliminating the need for separate computer vision algorithms during phase 1 of our project.

In cases where license plate characters are tightly clustered or overlapping, additional character segmentation may be necessary. This segmentation process separates individual characters within the license plate, preparing them for subsequent recognition.

Following detection and segmentation, the segmented characters undergo recognition using AWS Rekognition, an image and video analysis service. AWS Rekognition's built-in capabilities automatically analyze the segmented characters and decipher the alphanumeric content present on the license plates.

Upon image capture, a Lambda function (Lambda 1) is triggered, which performs two tasks. Firstly, it stores the captured image or video data in an S3 bucket for archival and further processing. Secondly, the Lambda function triggers AWS Rekognition for license plate detection, leveraging its capabilities to identify regions of interest within the images or videos, specifically targeting license plates. With the use of Amazon CloudWatch, the system's performance is continuously monitored for accurate analysis of image data, further enhancing the system's recognition capabilities. CloudWatch is leveraged for logging and monitoring system performance, ensuring that the system operates smoothly and efficiently. Finally, IAM manages access to AWS services, ensuring that only authorized users or services can interact with the system components, safeguarding its integrity and security.

After our initial image is processed by AWS Rekognition, the system is configured to send a notification to Amazon SNS upon detecting license plate information. Then a second Lambda function (Lambda 2) is triggered by the SNS notification. Lambda 2 has two primary functions: storing the output data in the database storage and identifying if the license plate is recognized or not in the database. DynamoDB, a NoSQL database service, is utilized as our database storage system for its flexibility and scalability in handling such metadata.If the license plate is recognized, Lambda 3 function will be triggered to open the parking lot gate.

However, if the license plate is not recognized, Lambda 4 will be triggered which will advise the customer to take a ticket as a guest. Additionally, Lambda 4 records the guest's decision to take a ticket in the database, facilitating payment processing upon exit. Once the guest takes the parking ticket, Lambda 3 automatically opens the parking lot gate. This orchestrated flow ensures seamless management of parking access based on license plate recognition outcomes.

2.1.1. Trade-offs between using AWS Rekognition and developing a custom computer vision model

The decision between using AWS Rekognition and developing a custom computer vision model depends on various factors such as time constraints, accuracy requirements, flexibility needs, scalability considerations, and budget constraints.

Firstly, regarding time to market and development effort, AWS Rekognition offers a turnkey solution with pre-trained models, reducing deployment time and requiring minimal development effort. In contrast, developing a custom model demands considerable time and resources for tasks like data collection, annotation, and model training, potentially delaying system deployment.

Secondly, in terms of accuracy and performance, AWS Rekognition provides robust performance with high accuracy levels, continuously updated and improved by AWS. Custom models may achieve higher accuracy through meticulous training and optimization but require significant expertise and resources to match the performance of AWS Rekognition.

Evaluating these factors carefully, we determined that we will do a two phase approach where we continue to develop our own custom made computer vision model for license plate recognition in phase 2.

2.1.2. Low confidence predictions

Our approach to low-confidence predictions is proactive. When confidence falls below a threshold, human experts validate predictions, ensuring critical decisions are based on high accuracy. Additionally, a feedback loop continuously collects and labels low-confidence instances. This iterative process refines the model, enhancing its predictive capabilities over time.

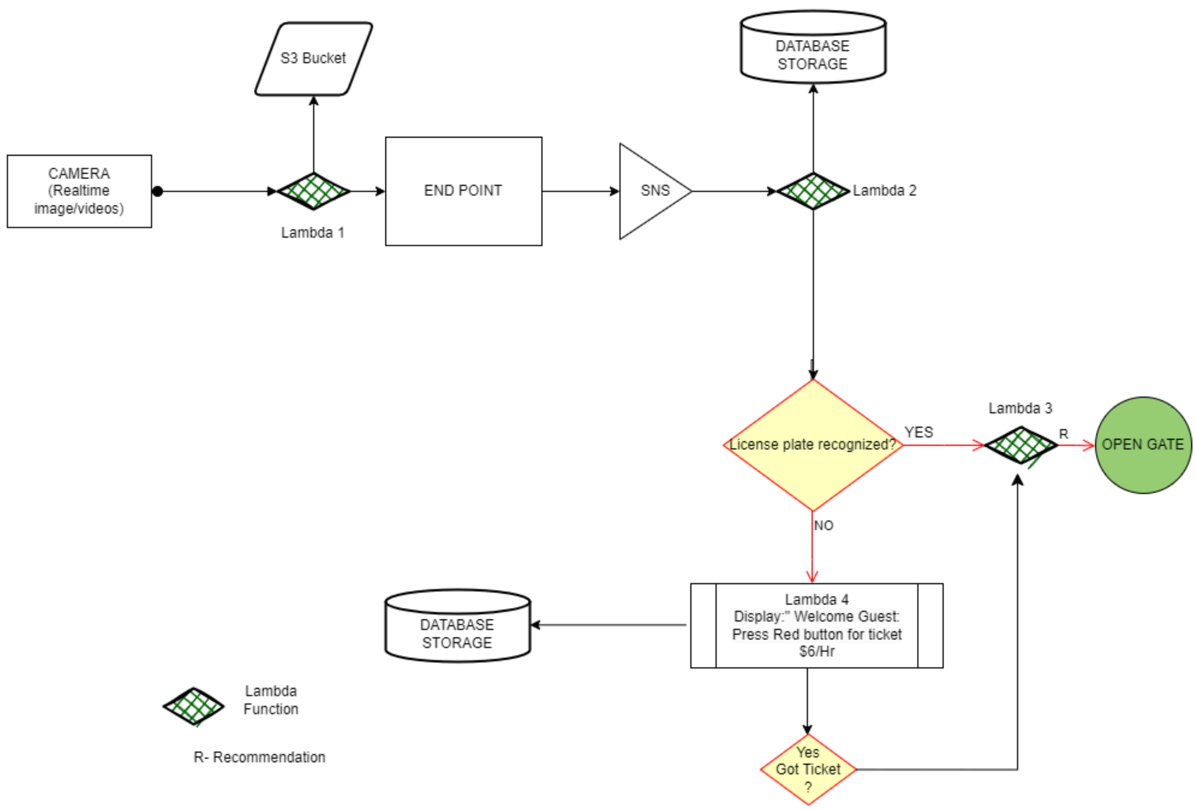
Looking ahead, we plan to integrate AWS Rekognition with custom models (phase 2) to leverage Rekognition's robustness while tailoring functionalities to our needs. The feedback loop remains crucial, providing continuous optimization of prediction accuracy.

2.2. Phase 2 Approach

In phase 2 of our project, we are making a big transition by moving away from relying on AWS Rekognition and instead developing our own custom license plate recognition system. The decision to do so stems from several considerations taken by the group. Firstly, while AWS Rekognition provides great functionality we discovered that developing our own specific use system offers greater control over the algorithms.

Moreover, by developing our own system, we plan to mitigate the cost escalations associated with using third party systems along with ensuring data privacy compliance.

Figure 2: Easy LPR 1 Parking Validation custom made Computer vision Flow Chart Phase 2



One of the biggest advantages of transitioning to a custom system is that we have the ability to leverage the data accumulated during Phase 1. By storing not only the license plate images captured by the camera but also the output from AWS Rekognition over our first running phase of the system, we have accumulated a valuable new dataset for our model development. This dataset will serve as the foundation of training and refinement of our custom system, in which we can precisely improve accuracy over time.

A major goal of our team is the continuous improvement of our system beyond the results attained during Phase 1. With a team of analysts and scientists, we will actively monitor the performance of our system. By tracking metrics such as recognition accuracy, processing speed, and our systems reliability, we will identify areas for improvement. This interactive approach allows us to ensure that our system evolves over time.

Furthermore, by enhancing the delivery speed and accuracy of predictions, we can improve the overall experience for users, which enables smoother entry and exit processes and overall security of the garage.

3. Systems improvements Over Time

In addition to accessing and analyzing the data stored in our S3 bucket and database, our team of analysts and scientists will use advanced techniques to extract valuable insight from data accumulated in Phase 1. By leveraging metadata generated by the AWS system, such as timestamps and location information, we will be able to identify patterns related to recognizing license plates. This analysis will embody a wide range of methods, including statistical analysis, machine learning algorithms, and data visualization allowing us to gain a deeper understanding of the patterns and how our Phase 1 system is working.

Furthermore, data scientists will utilize these insights to drive improvements in accuracy and performance of our license plate recognition system. Through the use of iterative refinement they will fine-tune parameters, train new models, and deploy more sophisticated algorithms. As the system accumulates more data, it will learn from past recognition results, which in turn will lead to continuous improvement within our metrics.

Another important aspect component of our improvement strategy is the integration of human review by Amazon Augment AI in Phase 1. This feature allows our team to validate and refine the predictions made by the Recognition system. By combining automated recognition with human review we establish a robust feedback loop that will drive ongoing refinement and improvement of the system. This will be crucial during the Phase 1 of our system as we will need the system to be performing at its best as we accumulate more data and seek to use that data to develop our own system.

4.Cost Estimations

Understanding AWS services and their associated costs is crucial for effective budgeting and resource allocation. AWS provides a variety of services for compute, storage, and data transfer, each with its pricing model and usage options.

Compute services like EC2 instances and Lambda functions are priced based on factors like instance type and usage duration, while storage options such as Amazon S3 and EBS volumes have pricing models based on storage capacity and data transfer.

It's essential to estimate costs accurately by considering factors like resource requirements and usage patterns. AWS offers flexible usage options, allowing users to pay for what they use without long-term commitments. This pay-as-you-go model is beneficial for optimizing costs while utilizing AWS resources effectively.

By understanding the services offered, estimating costs, and leveraging usage options like paying for what you use, AWS users can manage expenses efficiently while meeting their application's needs. In addition, we will be using SageMaker Training, SageMakerReal-time Inference and SageMaker Transform as described below:

SageMaker Training: This option allows us to train machine learning models on SageMaker using built-in algorithms, custom algorithms, or your own custom code. We would use this option to train our custom license plate recognition model using your dataset.

SageMaker Real-Time Inference: Once our model is trained, we can deploy it as a real-time inference endpoint using SageMaker Real-Time Inference. This allows you to make predictions in real-time by sending inference requests to the deployed endpoint.

SageMaker Batch Transform: In addition to real-time inference, we may also need batch inference capabilities for processing large batches of images. SageMaker Batch Transform allows you to run batch inference jobs on your trained model using input data stored in Amazon S3.

Besides the cloud computing costs, other costs that will be incurred will include hardware such as camera and gate equipment; and human resources such as a data scientist and parking attendant for backup.

Assuming that we have 100 images per day and the cost of estimation is for a month.

AWS Service	Monthly Price(US Dollars)	Description
S3	$0.0036	S3 Intelligent - Tiering
SageMaker Real-Time Inference	$40.32	1 model deployed/month, 1 model per endpoint, 1 instance per endpoint, 24 hr per day, 30 days per month, ml.t2.medium instance type
SageMaker Model Monitor	$2	30 model monitor jobs per month, 1 model monitor per month, 1 hr model monitor per job, ml.t.medium instance type
SageMaker Training	$0.12	1 train job per month, 1 instance per month, 1 hour training/month, ml.m5.large instance type
SageMaker Batch Transform	$3.45	30 batch transform jobs per month, 1 instance per job, 1hr per instance per job, ml.m5.large instance type
Amazon Rekognition	$3.00	This is with roughly 3000 images being processed through Rekognition a month
DynamoDB	$22.95	Monthly write cost (Monthly): 18.95 and Monthly read cost (Monthly): 3.75
CloudWatch	$6.05	With getting around 20 metrics per month
Lambda	$0.0000000021	128MB
SNS	$0.12	This is with publishing and delivering around 1.5GB of message data per month
Total	**$77.69**	

Note this pricing includes SageMaker. During phase 1 of our plan we can expect the price to be about $32.12 per month.

5.Conclusion

The integration of AWS Rekognition into our license plate recognition system offers significant advantages in efficiency and accuracy. By leveraging Rekognition's powerful image and video analysis capabilities, we streamline the process from image capture to license plate detection and recognition. This not only reduces development time but also ensures high accuracy in identifying license plates, even in challenging scenarios such as clustered or overlapping characters.

Furthermore, the use of AWS services like Lambda, S3, and DynamoDB enhances the scalability, reliability, and performance of our system. Lambda functions automate key processes such as image storage and recognition triggering, while S3 provides robust storage for captured data. DynamoDB efficiently manages metadata associated with recognized license plates, enabling seamless retrieval and analysis.

The incorporation of human review through Amazon Augment AI adds an extra layer of validation and refinement to our recognition process, ensuring accurate results and providing a robust feedback loop for continuous improvement.

Overall, the combination of AWS Rekognition and associated services with our custom system architecture sets the foundation for a robust, scalable, and accurate license plate recognition system that meets our requirements for efficiency, reliability, and security. Furthermore phase 2 of our projects showcases a vital look into the future of what the future may hold with the integration of our custom license plate recognition system.

The Dataset

To test the functionality and performance of our system before deployment and implementation into the company's garage system we shall be working with a Chinese City Parking Dataset (CCPD) of about 12GB with over 300k images with refined annotations. Each image will contain only a single license plate (LP). Detectors are allowed to predict only one bounding box for each image. A LP is correct if and only if all characters in the LP number are correctly recognized. This dataset will be a vital resource for evaluating our systems capabilities and gaining insights into its capabilities.

References

- (*Image Recognition Software, ML Image & Video Analysis - Amazon Rekognition - AWS*, n.d.)
- (*Machine Learning Service - Amazon SageMaker - AWS*, n.d.)
- (*Papers With Code - Towards End-To-End License Plate Detection and Recognition: A Large Dataset and Baseline*, n.d.)
- This dataset is based on a research paper: Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Yin, C., & Huang, L. (Year). "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline."