Case Study: Pomo Robotics

AWS AI for Fully Autonomous Trucks

Introduction

This case study details the development of an Autonomous Vehicle Stack for commercial vehicles. PomoRobotics aims to achieve Level 5 autonomy in logistics trucks by 2030. Utilising AWS high-powered GPU instances to run their ML algorithms not only reduced training time but also provided scalability needed. Training the model on AWS instances allowed them to process their dataset and train the model in just 2 days, compared to the week it previously required. Additionally, the ability to scale up resources when needed was a significant advantage over on-premises solutions.

Our Partnership

This was the customer's first engagement with transACT and their first experience working with any cloud provider. Although the customer had a team of highly skilled developers and data scientists, they lacked AWS expertise. transACT ensured that this lack of cloud experience did not impede their business objectives. We provided full support throughout the project lifecycle, assisting with design, architecture, deployment, and ongoing support, as well as addressing any ad hoc requests that arose during the project.

Business Challenges

Before transitioning to AWS, the team was working with public datasets but had to limit the number of training images and videos due to the extensive time required. Scalability was a significant challenge, especially since they needed to train the model under various weather and road conditions. Due to resource limitations, they had to reduce their dataset. However, with AWS, they could scale up or down as needed, resulting in continuous improvements in model accuracy. This progress aligns with their vision of achieving Level 5 autonomy by 2030. Pilot customers are pleased with the advancements and have requested trials with multiple trucks. While autonomous vehicles are more common in the U.S., Europe presents unique challenges due to narrower roads and unpredictable weather.

Challenges

PomoRobotics had been running ML workloads from their on-premises demo runs. However, they faced significant challenges:

1. Limited on-premises compute power for ML model training.
2. Need for advanced object detection and recognition in varying weather conditions.
3. Requirement for quick iteration and model fine-tuning.
4. Desire for easy-to-use ML platforms to accelerate development.

Implementation Process

1. Migration to AWS:

- a. Models were pretrained on COCO dataset. However, custom dataset form their trial run was included
- b. Utilized Amazon S3 for data storage and Amazon EC2 for compute resources

2. Model Selection and Development:

- a. Leveraged Amazon SageMaker JumpStart to explore and compare pre-trained models
- b. Selected the YOLOv3 model for object detection tasks

3. Model Fine-tuning and Training:

- a. Used SageMaker's built-in algorithms and custom frameworks for model adaptation.
- b. Implemented distributed training to accelerate the process

Model Comparison and Selection

PomoRobotics conducted an initial comparison of various models available on SageMaker JumpStart:

1. YOLOv3:

- a. Pros: Fast inference speed, good accuracy.
- b. Cons: Slightly lower accuracy on small objects.

2. Faster R-CNN:

- a. Pros: High accuracy, especially on small objects.
- b. Cons: Slower inference speed, more complex architecture.

3. ResNet (as a backbone for object detection):

- a. Pros: Deep architecture, good feature extraction.
- b. Cons: Requires additional detection head, not a complete object detection solution on its own.

4. SSD (Single Shot Detector):

- a. Pros: Fast, good for real-time applications.
- b. Cons: Lower accuracy compared to YOLOv3, especially on small objects.

After careful consideration, PomoRobotics chose YOLOv3 for the following reasons:

1. Balance of speed and accuracy: Crucial for real-time decision making in autonomous driving
2. Availability on SageMaker JumpStart: Pre-trained YOLOv3 models were readily available
3. Simplicity: Single-stage architecture made it easier to deploy and fine-tune

4. Performance in varying conditions: Showed robust performance in different weather and lighting conditions common in the UK.

5. Community support: Strong community backing ensured ongoing improvements and support.

Hyperparameters

PomoRobotics used the following approach for fine-tuning the YOLOv3 model:

1. Transfer Learning: Started with a pre-trained YOLOv3 model from SageMaker JumpStart and fine-tuned it on their custom dataset of UK roads and dataset from their trail runs.

2. Dataset: Used a curated dataset of 100,000 images, annotated with region-specific vehicles, road signs, and weather conditions.

3. Data Augmentation: Implemented various augmentation techniques including random scaling, rotation, and colour jittering, brightness.

4. Hyperparameters:

- a. Learning rate: 0.001 with step decay (decay factor 0.1 every 30 epochs)

- b. Batch size: 64

- c. Number of epochs: 100

- d. Input image size: 416x416 pixels

Evaluation Metrics

PomoRobotics used the following key metrics apart from several other to evaluate their fine-tuned YOLOv3 model:

- 1. Mean Average Precision (mAP): Achieved mAP of 0.88 at IoU threshold of 0.5

- 2. Precision and Recall: Average precision across all classes: 0.87, Average recall: 0.86

- 3. F1 Score: Average F1 score: 0.88

- 4. Inference Time: Achieved average inference time of 22ms per frame on their target hardware.

Results

By leveraging AWS SageMaker JumpStart and carefully selecting and fine-tuning the YOLOv3 model, PomoRobotics significantly improved their object detection capabilities for autonomous trucks operating between Scotland and the rest of the UK. The ease of use of SageMaker JumpStart allowed them to quickly iterate and optimize their model, overcoming the limitations they faced with on-premises computing resources.

The YOLOv3 model provided an excellent balance of speed and accuracy, crucial for real-time decision making in autonomous driving. While it didn't achieve the highest accuracy compared to two-stage detectors like Faster R-CNN, its speed and simplicity made it the best choice for PomoRobotics' specific use case.

The company achieved substantial improvements in detection accuracy, inference speed, and robustness to varying weather conditions. The flexibility t allowed them to easily experiment with different models before settling on YOLOv3, significantly accelerating their development process and positioning them as a leader in autonomous trucking technology in the UK.

This success bolstered the confidence of their pilot customers and the Scottish government, leading to government funding for the next phases of their project, keeping them on track with their commitments.

The solution involved the following AWS services:

1. Amazon S3 (Simple Storage Service)

- Store raw sensor data, pre-processed data, and trained models.

- S3 offers scalable and durable storage for large datasets, including LiDAR, images, and videos.

2. AWS IoT Core {To be used in Phase 2}

- Manage truck sensors and ingest real-time data.

- Enables secure communication between truck sensors and the AWS cloud.

3. AWS Lambda

- Trigger data processing workflows immediately after new data is uploaded to an S3 bucket.

4. Amazon SageMaker Jump start

- Preprocess data, train models, and deploy and manage ML models.

- Core service for building, training, and deploying ML models at scale.

5. Amazon ECR

- Store and manage Docker images for custom ML models like YOLO/ResNet .
- Integrates seamlessly with SageMaker for deploying custom models.

6. AWS Glue

- Automate the ETL (Extract, Transform, Load) process to clean and transform raw data into a structured format suitable for model training.

7. Amazon CloudWatch / SNS / IAM

- CloudWatch for monitoring, SNS for notifications, and IAM for secure authentication and access control.

8. Amazon SageMaker Notebook Instances

- For interactive development, hyperparameter tuning, and model experimentation.

Implementation

A) Data Collection and Storage

Data Sources:

- Data from various sensors, including LiDAR, cameras, radar, and GPS installed on the vehicle, were collected by the truck's edge computing device and transferred to AWS S3 on a weekly basis.

Event Automation:

- AWS Lambda functions were triggered by S3 upload events to initiate the AWS Glue pipeline.

B) Data Processing and Preparation:

- AWS Glue was used to automate the ETL (Extract, Transform, Load) process, cleaning and transforming raw data into a structured format suitable for model training.

C) Data Annotation:

- The YOLOv3 model was pre-trained on the COCO dataset, and manual labeling was performed for evaluation purposes. Although using SageMaker GroundTruth is planned, it has not yet been implemented.

D) Monitoring and Security

Monitoring:

- Amazon CloudWatch: Monitored AWS resource performance, set up alarms, and automated responses to specific conditions.
- AWS X-Ray: Traced and analysed requests across services to identify bottlenecks and performance issues.

Security:

- **AWS IAM:** Managed secure access to AWS resources by setting up roles and policies to ensure that only authorized personnel and services could access sensitive data and critical systems.
- Access to S3 was restricted to authorized users only.

E) Compliance and Data Governance:

Security:

- **AWS Config:** Used to monitor and audit AWS resource configurations, ensuring compliance with industry standards.

Products and Services

Sagemaker JumpStart, Sagemaker Notebook, S3, ECR, Lambda, IAM, Config.

Outcomes

- **80% Reduction in Model Training Time:** Achieved a significant decrease in the time required to train a model.
- **Scalable Resource Management:** Enabled flexible scaling of resources according to demand, with the ability to reduce resources when not in use.
- **Enhanced Focus on Core Business:** Delegated management, network, and administrative tasks to AWS, allowing developers and data scientists to concentrate on core business challenges.
- **Increased Prediction Accuracy:** Improved the accuracy of image predictions to 87%.