

Sistema de IA para Detecção de Placas Veiculares: Automação do Controle de Acesso com Visão Computacional

AI System for License Plate Detection: Automation of Access Control with Computer Vision

Rafael Funchal

Faculdade de Tecnologia da Baixada Santista | Fatec Rubens Lara
rafael.funchal@fatec.sp.gov.br

Gabrielle Vitória da Silva

Faculdade de Tecnologia da Baixada Santista | Fatec Rubens Lara
gabrielle.silva24@fatec.sp.gov.br

Pedro da Silva Santos

Faculdade de Tecnologia da Baixada Santista | Fatec Rubens Lara
pedro.santos241@fatec.sp.gov.br

Luciana Maria Guimaraes

Faculdade de Tecnologia da Baixada Santista | Fatec Rubens Lara
luciana.guimaraes4@fatec.sp.gov.br

Resumo

O controle de acesso veicular em terminais portuários enfrenta desafios devido à dependência de processos manuais, que permitem a entrada de veículos não autorizados, como carretas sem agendamento ou fora do horário agendado, causando congestionamentos e perda de eficiência no gate de entrada. Este estudo propõe um sistema automatizado baseado em visão computacional e inteligência artificial, utilizando o modelo YOLOv11n para detecção de veículos e placas, integrado ao modelo PaddleOCR para reconhecimento de caracteres e a um banco de dados para validação em tempo real. A metodologia incluiu treinamento do algoritmo de reconhecimento de placas veiculares com um extenso conjunto de imagens e testes em ambiente simulado, demonstrando alta confiabilidade na leitura de placas e capacidade de barrar veículos irregulares antes da entrada no terminal. Os resultados indicam que a solução pode eliminar transações desnecessárias, liberando tempo operacional equivalente ao processamento de dezenas a centenas de carretas adicionais por dia. Além disso, o sistema permite o registro de tentativas de acesso irregular, viabilizando a aplicação de medidas educativas, como restrições ou penalidades a motoristas e transportadoras reincidentes, promovendo maior conformidade com as normas de agendamento. Concluiu-se que a proposta é uma alternativa eficiente, de baixo custo e escalável, podendo ser utilizada para mitigar outros motivos de retorno no gate, sendo uma ferramenta capaz de aumentar a produtividade, reduzir gargalos operacionais e fortalecer o controle logístico em terminais portuários.

Abstract

The control of vehicle access in port terminals faces challenges due to the reliance on manual processes, which allow the entry of unauthorized vehicles, such as trucks without prior scheduling or outside their scheduled

Palavras-chave

Controle de Acesso
Visão Computacional
Inteligência Artificial
Terminais Portuários
Automação Logística

Keywords

Control
Computer Vision
Artificial Intelligence

time slots, causing congestion and loss of efficiency at the entry gate. This study proposes an automated system based on computer vision and artificial intelligence, using the YOLOv11n model for vehicle and license plate detection, integrated with PaddleOCR model for character recognition and a database for real-time validation. The methodology included training the license plate recognition algorithm with an extensive set of images and testing it in a simulated environment, demonstrating high reliability in reading license plates and the ability to block irregular vehicles before they enter the terminal. The results indicate that the solution can eliminate unnecessary transactions, freeing up operational time equivalent to processing dozens to hundreds of additional trucks per day. Furthermore, the system records irregular access attempts, enabling the implementation of corrective measures, such as restrictions or penalties for repeat-offender drivers and carriers, thereby promoting greater compliance with scheduling rules. It was concluded that the proposal is an efficient, low-cost, and scalable alternative, which can also be used to mitigate other reasons for gate returns, serving as a tool capable of increasing productivity, reducing operational bottlenecks, and strengthening logistical control in port terminals.

Port Terminals
Logistics Automation

1. Introdução

O controle de acesso efetivo de veículos às instalações portuárias representa um importante desafio nos terminais portuários. Em muitos cenários, esse controle ainda depende em grande parte de processos totalmente manuais - ou sequer existe - viabilizando o acesso indevido de veículos não autorizados às instalações. Um exemplo prático são as carretas que chegam ao terminal sem agendamento ou fora do horário agendado. Essas ocorrências criam um gargalo na fluidez operacional, provocando congestionamentos no gate de entrada originam manobras de retorno (U-Turn) no gate, que ocupam tempo e espaço que poderia ser utilizado por veículos regulares, reduzindo a capacidade operacional do terminal.

Sob essa ótica, durante o segundo semestre de 2024, uma operação portuária de grande porte foi analisada e registrou uma média diária de 47 carretas que acessaram o terminal sem agendamento ou fora do horário previamente estipulado, totalizando 8.638 ocorrências nesse período. Ainda nesse contexto, a presença de veículos fora dos padrões de agendamento também contraria diretrizes estabelecidas por legislações como a Resolução DP nº 136/2013 da Companhia Docas do Estado de São Paulo (CODESP), que prevê a organização do fluxo de cargas e veículos no transporte rodoviário, além de regulamentações internas de terminais que exigem agendamento prévio para entrada. O não cumprimento dessas normas pode resultar em penalidades, riscos operacionais e perda de controle logístico tanto para a transportadora e motorista, quanto para o terminal.

Dessa forma, a aplicação de tecnologias baseadas em Inteligência Artificial (IA), especialmente a visão computacional, surge como uma alternativa promissora para mitigar esses problemas, sendo uma área focada na interpretação e análise automatizada de imagens e vídeos. Essa ferramenta pode identificar placas veiculares com um elevado grau de precisão e oferece suporte em contextos como, por exemplo, o controle logístico em ambientes portuários.

Este trabalho propõe o desenvolvimento de um sistema inteligente e de alta confiabilidade para detecção automática de placas veiculares e validação de acesso, utilizando visão computacional e inteligência artificial para otimizar o controle de acesso na portaria de entrada dos terminais portuários. O sistema proposto é capaz de identificar, em tempo real, veículos que estejam irregulares, como carretas fora do horário ou sem agendamento prévio, realizando a leitura de sua placa antes que adentre as instalações do terminal e barrando seu acesso na portaria do terminal, sem a necessidade de sensores específicos, sistema de radares ou consulta manual. A solução é baseada exclusivamente em imagens capturadas por câmeras de alta resolução, o que a torna uma alternativa acessível e de fácil implementação.

Neste estudo aplicado, foi adotada uma metodologia quantitativa e descritiva, de tipo experimental. O estudo de caso realizado em ambiente controlado visou validar um algoritmo de visão computacional, fundamentado em inteligência artificial, que tem por objetivo detectar e reconhecer placas veiculares em tempo real, integrando-as ao sistema de controle de acesso, para a identificação dos veículos que não foram agendados ou que estão fora do horário. A validação do sistema foi realizada em ambiente simulado, buscando-se propor uma solução de pequeno custo e alta escalabilidade para os terminais logísticos.

2. Fundamentação Teórica

Portos contemporâneos vêm incorporando sistemas sofisticados de monitoramento baseados em tecnologias como Internet das Coisas (IoT), sensores, câmeras e análise de dados em tempo real. A emergência dos chamados "Portos Inteligentes" ou "Portos 4.0" impulsionou uma integração mais profunda de recursos tecnológicos nas atividades operacionais. Esses progressos vão além da simples modernização da infraestrutura, promovendo a adoção de soluções inovadoras que ampliam tanto a segurança quanto a eficiência em diferentes setores do ambiente portuário (Prosertek, 2023).

Nesse contexto, dentro da Inteligência Artificial, as Redes Neurais Convolucionais (CNNs – *Convolutional Neural Networks*) são uma classe de algoritmos especialmente eficazes no processamento de dados com estrutura espacial, como imagens. Inspiradas na forma como o cérebro humano processa estímulos visuais, as CNNs utilizam camadas de filtros (convoluções) para identificar automaticamente características visuais relevantes, como bordas, formas e texturas, reduzindo a necessidade de pré-processamento manual. Elas são amplamente usadas em tarefas como reconhecimento de objetos, detecção de rostos, leitura de placas veiculares e classificação de imagens (Scapicchio e Holdsworth, 2024).

O treinamento de uma rede neural consiste no ajuste de seus pesos internos para que ela possa realizar tarefas como classificação ou previsão. Esse processo é realizado por meio de iterações repetidas chamadas *epochs*, onde cada *epoch* representa uma passagem completa por todo o conjunto de dados de treinamento. Durante cada *epoch*, os dados são divididos em pequenos grupos chamados *batches*. O *batch size* define quantos exemplos são processados antes que a rede atualize seus pesos. Por exemplo, se um conjunto de dados contém 1.000 exemplos e o *batch size* é 100, serão necessárias 10 iterações (*batches*) para completar uma *epoch*. A escolha do número de *epochs* e do *batch size* influencia diretamente a eficácia e a eficiência do treinamento da rede neural (Brownlee, 2022).

A arquitetura YOLO (You Only Look Once), desenvolvida pela empresa *Ultralytics*, destaca-se entre os modelos de CNN mais utilizados para detecção em tempo real, amplamente adotado em aplicações de visão computacional, devido sua alta velocidade de processamento, utilizado para identificação de objetos como veículos, pessoas, bicicletas, móveis, entre outros, e aplicado em projetos como sistemas de veículos autônomos ou sistemas de vigilância, onde velocidade é essencial (Campos, 2024 apud SÁ et al., 2023). Os algoritmos da família YOLO são modelos do tipo single-shot, ou seja, analisam a imagem inteira em uma única varredura em busca de objetos, retornando as *bounding boxes* (coordenadas das caixas delimitadoras dos objetos encontrados na imagem), e suas respectivas classes (Campos, 2024 apud HUANG; PEDOEEM; CHEN, 2018). Essa abordagem é essencial para aplicações em tempo real, pois apresenta baixo tempo de inferência (tempo de processamento por quadro), e que permite que decisões sejam tomadas de forma quase instantânea. (Karagiannakos, 2019).

Ainda nesse contexto, destaca-se o *PaddleOCR*, ferramenta desenvolvida pela empresa chinesa Baidu para Reconhecimento Óptico de Caracteres (OCR) com base em redes neurais profundas. Trata-se de uma linha de modelos leves, eficientes e compatíveis com diversos idiomas e permite extrair texto de imagens com diferentes níveis de complexidade e tem sido amplamente aplicado em sistemas de LPR (*License*

Plate Recognition), automatizando a leitura de placas veiculares em contextos como controle de acesso, fiscalização e logística (PaddlePaddle, 2025).

3. Desenvolvimento da Temática

Esta seção discute o processo de desenvolvimento do sistema proposto com o objetivo de esclarecer como funciona um modelo de IA para detectar objetos como placas veiculares e demonstrar como pode ser implementado de forma eficiente.

3.1 Estrutura lógica do algoritmo desenvolvido.

O algoritmo desenvolvido tem como objetivo principal a leitura confiável dos caracteres das placas de carretas e carros em movimento, utilizando uma câmera de alta qualidade posicionada de forma a captar frontalmente tanto o cavalo mecânico quanto sua placa de licença. A solução é baseada exclusivamente em imagens capturadas, dispensando o uso de sensores adicionais, radares ou verificações manuais constantes. Ademais, o sistema é capaz de realizar a identificação e leitura de qualquer formato e layout de placa, inclusive placas estrangeiras.

Para garantir a acurácia da leitura, o sistema adota uma estratégia dinâmica de acompanhamento do veículo ao longo de toda sua permanência no campo de visão da câmera. Ao ser detectado, o veículo recebe um número de identificação único (ID) que permite rastreá-lo em tempo real. A cada novo frame, o algoritmo localiza o veículo, recorta a região onde se encontra a placa, realiza a leitura dos caracteres e armazena os resultados em uma matriz. Esse histórico de leituras sucessivas é combinado com um pós-processamento estatístico que avalia a leitura mais confiável (detalhado na seção 3.7), sendo essencial para mitigar falhas causadas por distorções momentâneas da imagem, variações de iluminação ou ângulos desfavoráveis.

Cada placa lida é validada em tempo real contra o banco de dados do terminal, verificando, por exemplo, a existência de agendamentos ativos. A leitura final e informações associadas, como horário e status, são registradas em um banco de dados (detalhado na seção 3.7). Esse tipo de abordagem integrada da detecção, rastreamento e validação, combinada com a redundância de múltiplas leituras ao longo do trajeto, assegura uma alta taxa de acerto mesmo em condições adversas de operação. Assim, o algoritmo não apenas maximiza a confiabilidade da identificação veicular, como também reduz a dependência de condições ideais de captura, promovendo maior robustez e escalabilidade para sua aplicação prática em ambientes logísticos.

3.2 Modelo de visão computacional para reconhecimento do veículo e placa

Para o desenvolvimento do algoritmo de inteligência *artificial*, foi escolhida a arquitetura YOLO11. Essa escolha se deu pelo fato de ser a versão mais recente e otimizada da família YOLO, trazendo melhorias em relação à precisão e velocidade se comparado a versões anteriores como o YOLOv5, YOLOv7 ou YOLOv8. Além disso, a *Ultralytics*, desenvolvedora do modelo, disponibiliza versões pré-treinadas com imagens do *dataset COCO (Common Objects On Context)*, um enorme banco de imagens de

objetos do cotidiano (Ultralytics, 2024). Nesse contexto, segundo a Ultralytics (2025), além do modelo já estar treinado para reconhecer uma grande parte dos objetos do dia a dia, incluindo carros e caminhões, ele pode passar por um processo de *fine-tuning*, que em outras palavras significa um retreinamento com novos dados, ampliando os tipos de objetos detectáveis, mas aproveitando sua estrutura otimizada.

Entre as variantes disponíveis do YOLOv11, como as versões *small* (s), *medium* (m), *large* (l) e *extra large* (x), optou-se pela versão nano (YOLOv11n), uma vez que essa é a mais leve da série, com o menor número de parâmetros (complexidade da rede neural) e o tempo de inferência (processamento por imagem) mais baixo, o que a torna ideal para aplicações em tempo real, como é o caso deste projeto. Embora as versões maiores ofereçam melhor desempenho na detecção em tarefas de alta complexidade ou ambientes com grande variedade de objetos, a versão Nano se mostrou mais que suficiente para os objetivos do projeto, que envolvem detecção rápida de veículos e placas em tempo real (Ultralytics, 2025).

No entanto, apesar da eficiência dos modelos YOLO em objetos maiores e bem definidos, de acordo com Kundu (2023), esse tipo de modelo, como muitos algoritmos de detecção, pode apresentar perda de precisão na detecção de objetos pequenos - como as placas veiculares - especialmente quando inseridas em imagens detalhadas ou quando os objetos estão longe da câmera.

Para contornar essa limitação, o processo de detecção da placa foi dividido em duas etapas principais: (1) detecção do veículo e (2) detecção da placa dentro da região do veículo. Essa abordagem em cascata permite reduzir a área de busca, concentrando a detecção de placas somente nas regiões onde veículos foram previamente identificados, aumentando significativamente a acuracidade do sistema, como demonstrado na figura 1.



Fonte: Os autores (2025).

3.3 Modelo para reconhecimento dos caracteres

Para a extração dos caracteres da imagem da placa foi escolhido especificamente o modelo versão PP-OCRv3 do *PaddleOCR*, o qual é um modelo extremamente leve e veloz, ideal para este tipo de aplicação em tempo real. (PaddlePaddle, 2024).

Figura 2 - Extração dos caracteres da placa



Fonte: Os autores (2025).

3.4 Processo de detecção do veículo

Para a detecção da região na imagem da câmera onde o veículo está localizado, não foi necessário realizar nenhum tipo de treinamento, uma vez que o modelo YOLO11n já é treinado para identificar carros e caminhões em imagens, desta forma, como o modelo foi treinado em um imenso *dataset* com variadas imagens, sua acurácia é muito alta, reduzindo a necessidade de *fine-tuning*. Neste contexto, é fornecido a imagem da câmera ao modelo, o qual irá retornar a *bounding box* do veículo, ou seja, a região na imagem o qual o veículo está localizado (Telega, 2024).

Como o objetivo é acompanhar o veículo ao longo de sua passagem na câmera, foi utilizado uma funcionalidade de tracking, desenvolvida pela *Ultralytics* e integrada ao YOLO, que atribui um ID de identificação a cada objeto detectado (Gosthipaty e Raha, 2024). Desta forma, é possível acompanhar o veículo durante sua passagem na câmera e saber se o veículo ainda está presente no frame da câmera ou não, podendo ser atribuído um gatilho para o algoritmo retornar a leitura mais confiável ao veículo sair do frame.

3.5 Processo de detecção da placa

Com o recorte da imagem onde está localizado o veículo, é necessário localizar a placa. Para isso, também foi utilizado o modelo YOLO11n, porém com uma particularidade importante: este modelo não é originalmente treinado para reconhecer placas. Por isso, foi necessário realizar um processo de *fine-tuning*, ou seja, ensinar ao modelo o que é uma placa, treinando-o com diversas imagens de placas de veículos e adaptando os pesos pré-treinados para que o modelo agora aprenda a identificar placas veiculares com precisão.

Essa técnica de adaptar os pesos pré-treinados, segundo Brownlee (2020), se chama *transfer learning* e permite aproveitar o conhecimento prévio do treinamento anterior do modelo (como detecção de formas, bordas, padrões de textura etc.), sem a necessidade de treinar uma rede do zero, o que exigiria muito mais tempo, e estaria propenso a erros de generalização do modelo, que ocorrem quando um modelo de IA não tem bons resultados com dados novos por conta de *overfitting* (o modelo "decora" os dados de treino, mas não entende padrões gerais) ou *underfitting* (o modelo aprende pouco, ficando simples demais para captar padrões importantes).

3.5.1 Treinamento do modelo de detecção de placas

Para o treinamento do modelo para detecção de placas, foi utilizado o *dataset* de imagens de domínio público "*Large-License-Plate-Detection-Dataset*" disponibilizado na plataforma *Kaggle*, que contém uma variedade de imagens de treinamento, validação e teste, contendo placas veiculares em condições adversas de ângulo, distorção e iluminação, junto de suas respectivas *labels* – arquivos de texto, contendo as coordenadas das placas nos veículos na imagem. Os volumes de imagens de treino, validação e teste utilizados no treinamento estão especificados no Quadro 1.

Quadro 1 - Relação dos dados de treino

Objetivo	Quantidade
Treino (para treino do modelo)	25500
Validação (para ajuste do modelo durante treinamento)	1000
Teste (para avaliação final após o treinamento do modelo)	400

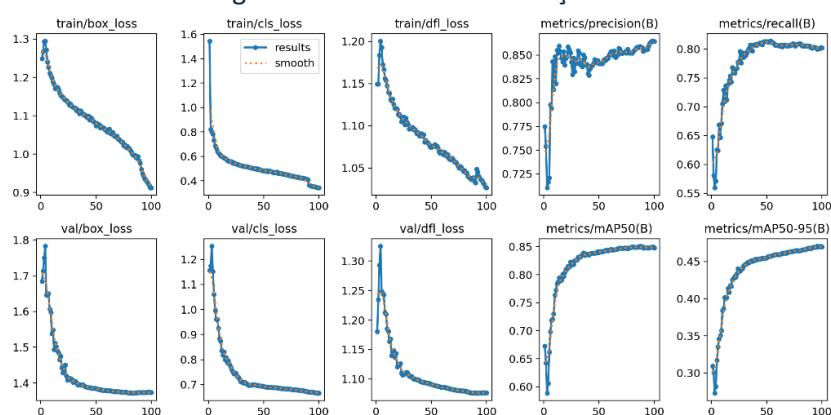
Fonte: Kaggle (2024).

Para realizar o treinamento de uma rede neural com esse volume de imagens, conforme evidenciado por Scapicchio e Holdsworth (2024) é necessário um alto poder computacional, geralmente sendo utilizada uma placa gráfica (GPU) potente, caso contrário, o treinamento do modelo pode durar dias até atingir um resultado satisfatório. Entretanto, há soluções para esse viés, sem ter necessariamente o hardware necessário, utilizando poder computacional em plataformas em nuvem.

Nesse contexto, para o treinamento do modelo, foi utilizado a plataforma Kaggle, que fornece de forma gratuita poder de processamento de GPUs via streaming. Dessa forma, foi configurado o ambiente de treinamento em Python 3, utilizando aceleração com GPUs T4, com 100 *epochs* e um *batch size* de 16.

Após 8 horas de treinamento, o modelo obtido no último *epoch* teve resultados incrivelmente satisfatórios, conforme os gráficos de evolução do modelo na Figura 3.

Figura 3 - Estatística de evolução do modelo



Fonte: Os autores (2025).

O modelo apresentou uma evolução consistente ao longo das 100 *epochs* de treinamento, com todas as perdas de treino e validação (*box*, *cls* e *dfl*) diminuindo progressivamente, o que indica um aprendizado estável. As métricas de desempenho também melhoraram continuamente, com a precisão atingindo cerca de 87%, o recall estabilizando em torno de 81%, e o mAP50 chegando a 85%, demonstrando ótima capacidade de detecção. Além disso, o mAP50-95, uma métrica mais rigorosa, alcançou 48%, reforçando que o modelo não só detecta bem, mas também localiza os objetos com boa precisão. Esses resultados são decorrentes do grande volume de imagens de treino e mostram que o modelo está bem treinado e preparado para aplicação prática.

3.6 Processo de reconhecimento dos caracteres

Após ter o modelo para reconhecimento do veículo e o modelo treinado para reconhecimento da placa, a última etapa é a extração dos caracteres da placa. Para isso,

processou-se a imagem com o modelo PP-OCrv3 do *PaddleOCR*, que retornará a cadeia de caracteres extraída da placa.

Técnicas de pré-processamento da imagem geralmente são aplicadas para melhorar a legibilidade da imagem, com o fito de auxiliar o modelo a reconhecer os caracteres com maior assertividade (Tavares, 2024, p. 5). Neste caso, o *PaddleOCR* teve uma performance excelente na extração de caracteres apenas com ajustes de normalização de brilho e contraste na imagem recortada da placa, conforme a Figura 4. Esse tipo de pré-processamento é relativo a diferentes cenários e os valores foram ajustados com base na tentativa e erro até obter leituras corretas e consistentes em diferentes cenários de iluminação.

Figura 4 - Antes e depois do pré-processamento



Fonte: Os autores (2025).

3.7 Otimização geral do algoritmo

Conforme comentado anteriormente, o algoritmo deve garantir leituras precisas de placas veiculares em condições desafiadoras, como iluminação variável ou distorções, e para isso, processa múltiplas leituras de caracteres extraídos pelo *PaddleOCR* enquanto o veículo está no campo de visão da câmera. Esse pós-processamento funciona da seguinte forma: a cada frame, o sistema gera propostas confiáveis usando uma média, que considera quantas vezes cada leitura aparece. O código desenvolvido faz isso somando as leituras, calculando uma média baseada em sua frequência, e retornando uma proposta se ela ultrapassar um limite mínimo de confiabilidade, definido experimentalmente. Essa proposta é enviada ao sistema do terminal, que verifica se a placa corresponde a um agendamento ativo. Se confirmada, o processamento para e a leitura é dada como correta.

Se o sistema do terminal não validar nenhuma proposta durante a passagem do veículo, o algoritmo seleciona a leitura mais provável ao final, usando outro código, que compara todas as leituras, dando pontos para placas com sequências de caracteres mais frequentes e verificando se os caracteres aparecem nas mesmas posições (ex.: "ABC123" aparece mais que "ABD123"). A leitura com maior pontuação é escolhida. A leitura final, junto com a foto do veículo, horário e resultado da validação, é salva em um banco de dados. Essa abordagem tem o objetivo de assegurar alta precisão e robustez nos resultados das leituras do algoritmo.

4. Resultados e Discussão

O sistema desenvolvido foi pensado para ser flexível a qualquer tipo de uso, sendo responsável pela leitura assertiva das placas, podendo ser integrado de acordo com a necessidade de uso, como, por exemplo, para validar agendamentos. Nesta seção será discutido os requisitos para implantação do sistema, além da simulação da aplicação real do sistema nesse cenário.

4.1 Requisitos de implantação do sistema.

A operação eficiente do sistema proposto requer uma infraestrutura adequada de hardware. Nesse contexto, para a captura de imagens em tempo real, é essencial uma câmera de alta resolução (mínimo de 1080p) com taxa de quadros elevada (30 FPS ou superior), posicionada estrategicamente para garantir o enquadramento frontal das placas veiculares e veículo, mesmo em condições de movimento. O processamento das imagens demanda um computador com capacidade robusta, incluindo uma GPU (placa gráfica) dedicada (ex.: NVIDIA T4, RTX 3060 ou superior) para acelerar a inferência dos modelos YOLOv11n e *PaddleOCR*, além de um processador *multicore* (ex.: Intel i7 ou AMD Ryzen 7) e memória RAM mínima de 16 GB para evitar gargalos.

4.2 Simulação

Para a simulação da aplicação do algoritmo no caso de validação de agendamento, foi consultado um terminal da baixada santista, que disponibilizou dados do volume de carretas que chegaram ao Gate de entrada sem agendamento ou fora do horário de agendamento no período do segundo semestre de 2024, sendo a média diária de 47 carretas, totalizando 8638 ocorrências no período.

Com base nessas informações, observou-se a oportunidade de implementar o sistema desenvolvido, com o objetivo de barrar essas carretas na portaria anterior ao Gate, direcionando-as ao estacionamento para correção do agendamento ou aguardo do horário agendado, ou até mesmo orientar que o motorista faça o retorno e saia das dependências do terminal, dessa forma, desfogando as operações do Gate, mitigando o congestionamento por conta dessas carretas. Nesse contexto, cabe analisar como esse controle seria realizado: de forma humana, ou automático.

No caso de uma integração totalmente automática, seria necessário a adoção de um sistema de cancelas automáticas e indicações visuais, podendo aumentar a eficiência e confiabilidade, pois elimina a dependência de fatores humanos e reduzindo erros no direcionamento. No entanto, a implementação desse sistema exige maiores investimentos em infraestrutura, como instalação de equipamentos e integração com sistemas existentes, o que pode acarretar custos elevados e prazos mais longos para implantação.

Em contrapartida, a validação manual é de implantação mais rápida e com menor custo, entretanto, com maior chance de falha, pois depende a correta atuação do inspetor, que demanda treinamento e reforço contínuo nas instruções. Neste caso, será alocado um inspetor na portaria o qual portará um tablet ou notebook com um sistema visual, que mostra as carretas conforme chegam, mostrando informações como foto da carreta, placa, tipo de agendamento, status (verde - agendado, vermelho - não agendado, amarelo - antecipado), Gate agendado, nome do motorista, entre outras, conforme necessidade. Pensando nesse cenário, foi desenvolvido uma proposta de interface para ser utilizada por um inspetor na portaria, a qual mostra as informações referentes ao veículo e ao seu agendamento, conforme a Figura 5.

Figura 5 - Screenshot do sistema proposto para validação manual



Fonte: Os autores (2025).

Com esse sistema, quando a carreta chegar na portaria, o inspetor verificará a placa no sistema e validará suas informações, pôr fim direcionando-a conforme protocolo alinhado.

4.2.1 Resultados obtidos.

Considerando o cenário em que o problema de carretas sem agendamento e fora do horário fosse resolvido por completo, foi possível simular o ganho em produtividade de Gate comparado ao cenário anterior onde não havia o controle de entrada.

Segundo dados fornecidos pelo terminal, uma média de 47 carretas diárias chega ao Gate sem agendamento ou fora do horário previsto, e os tempos de transação, calculados como a diferença entre a abertura da cancela e a chegada da carreta ao Gate, são apresentados no Quadro 2, incluindo durações para retornos e operações normais.

Quadro 2 - Relação de tempos de transação

Operação	Duração
Tempo de transação de retorno	3 minutos
Tempo de transação normal – Carreta cheia	2,5 minutos
Tempo de transação normal – Carreta vazia	40 segundos (aprox. 0,66 minutos)

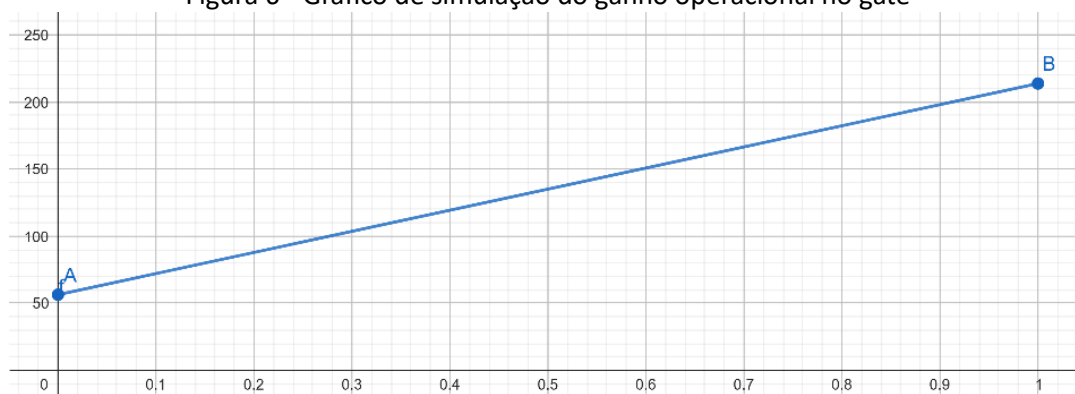
Fonte: Os autores (2025).

Considerando um ganho diário de 141 minutos operacionais no Gate de entrada, devido à eliminação de transações de retorno, pôde-se simular o quanto esse tempo extra permite aumentar o volume de transações efetivas, dependendo da proporção de carretas vazias e cheias.

Chama-se de x a porcentagem de carretas vazias em relação ao volume total médio de carretas processadas. A porcentagem de carretas cheias é então $(100 - x)$, ou seja, é o valor complementar.

A função $f(x)$ representa o ganho em volume de carretas processadas, com base na porcentagem x de carretas vazias que compõem o tempo total de 141 minutos. Portanto, por exemplo, se 60% dos 141 minutos extras diários for preenchido pelo recebimento de carretas vazias, os outros 40% serão carretas cheias. Dessa forma, a soma do volume de carretas vazias e cheias que compõem esse tempo de 141 minutos, considerando os tempos médios de transação.

Figura 6 - Gráfico de simulação do ganho operacional no gate



Fonte: Os autores (2025).

$$f(x) = x \times \frac{141}{0,66} + (1 - x) \times \frac{141}{2,5}, x \in \{0, 100\} \quad (B)$$

$$A_{\min} \text{ (apenas carretas cheias)} \approx 56 \text{ carretas} \quad (C)$$

$$B_{\max} \text{ (apenas carretas vazias)} \approx 213 \text{ carretas} \quad (D)$$

Conclui-se, nesse cenário, que o terminal obterá um ganho médio entre 56 e 213 carretas diárias com transações válidas no *gate* ao implantar o sistema de controle de agendamento na portaria, aumentando suas operações. O Quadro 3 mostra o efeito dos resultados obtidos com base na visão mensal e anual.

Quadro 3 - Relação de tempos de transação

Visão	Ganho
Mensal	De 1680 a 6390 carretas
Anual	De 20440 a 77745 carretas

Fonte: Os Autores (2025).

Sob essa análise, verifica-se um ganho operacional significativo, utilizando o tempo desperdiçado por carretas fora das normas para realizar operações normais.

5. Considerações Finais

A aplicação de visão computacional e inteligência artificial no controle de acesso veicular em terminais portuários demonstra grande potencial para transformar o controle logístico. Nesse sentido, o sistema desenvolvido permite a identificação imediata de veículos sem autorização válida antes mesmo de acessarem as instalações. Essa automação, integrada ao correto barramento desses veículos, diminui e reduz o número de manobras de retorno no *gate* e aumenta significativamente a produtividade e fluidez operacional.

Além da verificação de regularidade com o agendamento, o sistema pode ser facilmente expandido para tratar outros motivos comuns de bloqueio, como motoristas sem cadastro biométrico ou motoristas sem integração válida no sistema. Assim, essas informações, também disponíveis nas bases do terminal, podem ser incorporadas à lógica de verificação automática, ampliando o controle e reduzindo ainda mais os retornos no *gate*. Ademais, o sistema de reconhecimento de placas pode ser utilizado para outras funcionalidades em outras áreas do terminal, uma vez que pode ser totalmente customizado, não se limitando apenas para controle de acesso na portaria.

Outro ponto relevante é o uso estratégico dos dados coletados. Ao registrar cada tentativa de entrada negada, o sistema gera uma base sólida, que pode ser utilizada para aplicação de medidas corretivas, como advertências, restrições ou penalidades a transportadoras, motoristas ou veículos reincidentes. Isso contribui para educar os envolvidos sobre a importância do cumprimento das normas de agendamento, promovendo uma mudança de comportamento e fortalecendo a disciplina operacional.

Com base nos dados supracitados, a solução proposta não apenas representa uma inovação no processo de controle de acesso, como também oferece ferramentas para gestão estratégica e preventiva, com baixo custo e alta escalabilidade, sendo aplicável em diversos tipos de terminais logísticos.

Referências

BROWNLEE, Jason. *Difference Between a Batch and an Epoch in a Neural Network*.

Machine Learning Mastery, 2022. Disponível em:

<https://www.machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.

Acesso em: 4 maio 2025.

BROWNLEE, Jason. How to use transfer learning when developing convolutional neural network models. **Machine Learning Mastery**, 18 ago. 2020. Disponível em:

<https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>. Acesso em: 4 mai. 2025.

CAMPOS NETO, Manoel Freitas. *Desenvolvimento de ferramentas de IA para o monitoramento da dinâmica de colmeias de abelhas canudo (*Scaptotrigona spp*)*. 2024.

89 f. **Dissertação (Mestrado em Engenharia Elétrica)** – Instituto de Tecnologia, Universidade Federal do Pará, Belém, 2024. Disponível em:

https://repositorio.ufpa.br/jspui/bitstream/2011/16911/1/Dissertacao_DesenvolvimentoFerramentasIA.pdf. Acesso em: 4 mai. 2025.

CODESP. Resolução DP nº 136, de 30 de dezembro de 2013. **Estabelece o regimento para o acesso terrestre ao Porto de Santos.** Disponível em:

<https://www.gov.br/transportes/pt-br/centrais-de-conteudo/res-136-2013-pdf>. Acesso em: 4 mai. 2025.

ELMENSHAWII, Fares. **Large License Plate Dataset: [dataset]**. Kaggle, [s.d.]. Disponível em: <https://www.kaggle.com/datasets/fareselmenshawii/large-license-plate-dataset>.

Acesso em: 4 mai. 2025.

GOSTHIPATY, Aritra Roy; RAHA, Ritwik. Object Tracking with YOLOv8 and Python.

PylmageSearch, 17 jun. 2024. Disponível em:

<https://pyimagesearch.com/2024/06/17/object-tracking-with-yolov8-and-python/>. Acesso em: 4 mai. 2025.

HOLDSWORTH, Jim; SCAPICCHIO, Marcos. **What is deep learning?** 17 jun. 2024.

Disponível em: <https://www.ibm.com/think/topics/deep-learning>. Acesso em: 4 maio 2025.

KARAGIANNAKOS, Sergios. **YOLO: you only look once (single shot detectors).** *AI*

Summer, 13 maio 2019. Disponível em: <https://theaisummer.com/YOLO/>. Acesso em: 4 mai. 2025.

KUNDU, Rohit. YOLO: algorithm for object detection explained [+examples]. **V7 Labs**, 17

jan. 2023. Disponível em: <https://www.v7labs.com/blog/yolo-object-detection>. Acesso em: 4 mai. 2025.

PADDLEPADDLE. Reconhecimento de placas de veículos com PaddleOCR Light.

PaddleOCR Documentation, [s.d.]. Disponível em:

<https://paddlepaddle.github.io/PaddleOCR/main/en/applications/%E8%BD%BB%E9%87%8F%E7%BA%A7%E8%BD%A6%E7%89%8C%E8%AF%86%E5%88%AB.html#1>. Acesso em: 4 mai. 2025.

PROSERTTEK. **Innovation in port risk management.** *Prosertek*, 25 ago. 2023. Disponível

em: <https://prosertek.com/blog/innovation-in-port-risk-management/>. Acesso em: 4 mai. 2025.

TAVARES, Renato Augusto. Comparison of Image Preprocessing Techniques for Vehicle

License Plate Recognition Using OCR: Performance and Accuracy Evaluation. [S. l.]: **arXiv**, 15 out. 2024. Disponível em: <https://arxiv.org/pdf/2410.13622>. Acesso em: 4 maio 2025.

TELEGA, Slawomir. **YOLO8 basics: plotting bboxes.** *Medium*, 11 abr. 2024. Disponível

em: <https://medium.com/@telega.slawomir.ai/yolo8-basics-plotting-bboxes-e50a8f3e5227>. Acesso em: 4 mai. 2025.

ULTRALYTICS. **Conjunto de dados COCO – Ultralytics YOLO Docs**, 2024. Disponível em:

<https://docs.ultralytics.com/pt/datasets/detect/coco/>. Acesso em: 4 maio 2025.

ULTRALYTICS. Perspectivas sobre a avaliação e o aperfeiçoamento de modelos.

Ultralytics YOLO Docs, [s.d.]. Disponível em:

<https://docs.ultralytics.com/pt/guides/model-evaluation-insights/>. Acesso em: 4 mai. 2025.

"Os conteúdos expressos no trabalho, bem como sua revisão ortográfica e adequação às normas ABNT são de inteira responsabilidade dos autores."

«Declaração de IA generativa e tecnologias assistidas por IA no processo de redação»

“Declara-se pelos autores que durante a preparação deste trabalho foi utilizado ChatGPT para revisão ortográfica. Após utilizar esta ferramenta/serviço, os autores editaram e revisaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo da publicação.”