

RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES COM APRENDIZADO PROFUNDO PARA A SEGURANÇA EM CONDOMÍNIOS

Luan Costa Margotti¹, Luiz Ricardo Fiera²

Resumo: A segurança em condomínios residenciais de pequeno e médio porte representa um desafio crescente no Brasil, onde os métodos tradicionais de controle de acesso veicular, como registro manual por porteiros e sistemas de identificação por radiofrequência, apresentam limitações relevantes, incluindo suscetibilidade a falhas humanas e compartilhamento indevido de credenciais. Este trabalho propõe e avalia um sistema de Reconhecimento Automático de Placas Veiculares voltado ao controle de acesso condominial, com foco específico no padrão Mercosul vigente desde 2018. O sistema foi desenvolvido em Python com a arquitetura YOLOv8n, treinado sobre 6.536 imagens de placas veiculares reais em carros, e integra cinco estágios sequenciais: detecção da placa, pré-processamento, reconhecimento de caracteres, pós-processamento por expressão regular e controle de acesso com registro em log. Os experimentos foram conduzidos em estação de trabalho com GPU NVIDIA GeForce RTX 3060. Sobre o conjunto de 1.862 imagens de validação, o sistema alcançou acurácia de 92,37% no reconhecimento completo das placas e velocidade de processamento de 25,85 quadros por segundo, demonstrando viabilidade operacional para controle de acesso em tempo real. Complementarmente, em três vídeos capturados por smartphone em cenários condominiais reais, o sistema identificou corretamente as placas e emitiu as decisões esperadas nos três cenários avaliados: morador com acesso ativo, morador com cadastro inativo e visitante não cadastrado. Os resultados indicam a viabilidade técnica do sistema proposto como alternativa baseada em infraestrutura convencional aos sistemas comerciais de controle de acesso veicular.

Palavras-chave: reconhecimento de placas veiculares; aprendizado profundo; YOLOv8; padrão Mercosul; controle de acesso.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. luancm56@unesc.net

²Orientador, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. ricardo.fiera@gmail.com

ABSTRACT: Security in small and medium-sized residential condominiums represents a growing challenge in Brazil, where traditional vehicle access control methods, such as manual registration by doormen and radio-frequency identification systems, present relevant limitations, including susceptibility to human error and improper credential sharing. This work proposes and evaluates an Automatic License Plate Recognition system aimed at condominium access control, with specific focus on the Mercosul standard in force since 2018. The system was developed in Python using the YOLOv8n architecture, trained on 6,536 images of real vehicle license plates in cars, and integrates five sequential stages: plate detection, preprocessing, character recognition, regular expression post-processing, and access control with log registration. Experiments were conducted on a workstation with an NVIDIA GeForce RTX 3060 GPU. On the 1,862-image validation set, the system achieved 92.37% accuracy in complete plate recognition and a processing speed of 25.85 frames per second, demonstrating operational viability for real-time access control applications. Additionally, in three smartphone-captured videos in real condominium scenarios, the system correctly identified plates and issued the expected decisions in all three evaluated scenarios: active resident, inactive resident, and unregistered visitor. The results indicate the technical feasibility of the proposed system as an alternative based on conventional infrastructure to commercial vehicle access control systems.

Keywords: license plate recognition; deep learning; YOLOv8; Mercosul standard; access control.

1 INTRODUÇÃO

A segurança em condomínios residenciais de pequeno e médio porte consolidou-se como uma preocupação crescente no Brasil. Nesses ambientes, frequentemente operados com orçamentos restritos e infraestrutura tecnológica limitada, os processos de urbanização e o aumento da criminalidade exigem sistemas de controle de acesso eficazes e economicamente viáveis. Segundo a Associação Brasileira das Administradoras de Imóveis (ABADI), existem mais de 68 mil condomínios apenas no estado de São Paulo, evidenciando a necessidade de soluções escaláveis para esse nicho (Associação Brasileira das Administradoras de Imóveis (ABADI), 2022). Em cenários práticos de implantação, os custos elevados das soluções comerciais disponíveis tornam sua adoção inviável para grande parte

desse segmento.

Entre os mecanismos de controle mais utilizados em condomínios, destacam-se o registro manual por porteiros, os sistemas de identificação por radiofrequência (TAGs) e os cartões magnéticos. Apesar de difundidos, esses métodos apresentam limitações relevantes: suscetibilidade a falhas humanas, clonagem ou compartilhamento indevido de credenciais e incapacidade de resposta ágil em situações emergenciais (Izidio, Ferreira e Barros, 2018).

O Reconhecimento Automático de Placas Veiculares (do inglês *Automatic License Plate Recognition*, ALPR) representa uma solução tecnológica promissora para esse cenário. Em sua fundamentação teórica moderna, esse sistema baseia-se em Visão Computacional e Redes Neurais Convolucionais (CNNs) para realizar inferências sobre imagens, técnicas consolidadas pela literatura clássica de aprendizado profundo (LeCun, Bengio e Hinton, 2015), aliando inteligência artificial à praticidade e à segurança, com potencial de substituir ou complementar os métodos tradicionais de identificação (Du et al., 2013).

Laroca et al. (2018) foram pioneiros na aplicação sistemática do detector YOLO como solução central para um *pipeline* ALPR completo, propondo uma abordagem de dois estágios na qual CNNs independentes foram treinadas e ajustadas para cada etapa do reconhecimento, conferindo robustez a variações de câmera, iluminação e plano de fundo. Para o reconhecimento de caracteres, os autores introduziram duas técnicas distintas de *data augmentation*: o espelhamento horizontal de caracteres isolados, que gera variações sintéticas de cada glifo, e a inversão da imagem completa da placa (rotação de 180°), que simula instalações invertidas ou câmeras em ângulos incomuns. Embora ambas ampliem artificialmente a diversidade do treinamento, atuam em escalas diferentes caractere individual e placa como um todo, respectivamente contribuindo juntas para a robustez do modelo a variações de câmera, iluminação e plano de fundo. O sistema foi avaliado sobre o *dataset* SSIG, composto por 2.000 quadros extraídos de 101 vídeos de veículos, e sobre o *dataset* UFPR-ALPR, proposto pelos próprios autores, contendo 4.500 quadros capturados em condições dinâmicas com câmera e veículos em movimento, abrangendo automóveis, motocicletas, ônibus e caminhões. No *dataset* SSIG, o sistema superou sistemas comerciais como Sighthound (89,80%) e OpenALPR (93,03%), alcançando taxa de reconhecimento de 93,53% e desempenho de 47 FPS; no UFPR-ALPR, obteve 78,33% de reconhecimento a 35 FPS, cenário em

que as soluções comerciais não superaram 70%.

Em trabalho posterior, Laroca et al. (2021) aprimoraram a arquitetura tornando-a independente do *layout* da placa, sem assumir restrições sobre a disposição dos caracteres ou o formato visual da placa. O sistema foi avaliado sobre oito *datasets* públicos de cinco regiões distintas, alcançando taxa média de reconhecimento de 96,9% de ponta a ponta, superando tanto os trabalhos anteriores quanto os sistemas comerciais avaliados, e consolidando o YOLO como referência para o estado da arte em ALPR em múltiplos contextos geográficos. Em estudo complementar, Laroca et al. (2023) investigaram a fusão de múltiplos modelos para reconhecimento de placas, obtendo ganhos de desempenho em relação às abordagens individuais e reforçando a consolidação de arquiteturas baseadas em YOLO como referência para sistemas ALPR modernos.

No contexto brasileiro, com ênfase em restrições severas de infraestrutura, Izidio, Ferreira e Barros (2018) propuseram uma solução embarcada implementada em Raspberry Pi 3 com módulo de câmera Pi NoIR v2, escolhidos pela viabilidade econômica em cenários sem acesso a GPUs de uso geral. A detecção das placas foi realizada pela arquitetura Tiny YOLOv3 e o reconhecimento de caracteres por uma segunda CNN treinada sobre imagens sintéticas e posteriormente ajustada com imagens reais, estratégia que permitiu contornar a escassez de dados rotulados. O sistema foi validado sobre imagens reais de placas capturadas em diferentes condições ambientais, demonstrando capacidade de operação a variações de ângulo, iluminação e ruído, e obtendo 99,37% de taxa de detecção e 97,00% de reconhecimento, com tempo médio de processamento de 2,70 segundos por imagem de resolução 1024×768 pixels em plataforma embarcada de recursos reduzidos.

Para o desafio específico do padrão Mercosul, adotado em 2018 pelo Brasil e pelos demais países membros do bloco, Ribeiro et al. (2019) abordaram o problema da escassez crítica de imagens reais rotuladas do novo padrão por meio do uso exclusivo de dados sintéticos para treinamento. As placas Mercosul foram reproduzidas digitalmente com fidelidade e técnicas de processamento de imagem foram empregadas para reduzir a diferença de domínio (*domain gap*) entre os dados sintéticos e as imagens reais, enquanto manipulações básicas de imagem foram utilizadas para inserir as placas artificiais em contextos visuais realistas. O modelo treinado foi validado sobre imagens reais capturadas em estacionamento e em vídeo público de monitoramento de tráfego, alcançando 95% de precisão

na localização das placas e tempo médio de execução de 40 milissegundos. Contudo, o escopo do trabalho restringiu-se à etapa de detecção, sem avançar para o reconhecimento completo dos caracteres nem para a integração com lógica de controle de acesso. No âmbito do reconhecimento de caracteres por aprendizado profundo, Selmi et al. (2020) propuseram um sistema integrado de detecção e reconhecimento de placas baseado em *pipeline* de múltiplos estágios, demonstrando que a combinação de etapas especializadas contribui para ganhos de desempenho em cenários variados. No contexto brasileiro, Almeida, Oliveira et al. (2018) contribuíram com o *dataset* RodoSol-ALPR, composto por imagens de veículos brasileiros e mercosulinos capturadas em praças de pedágio, reforçando a relevância de conjuntos de dados nacionais para o treinamento e a validação de sistemas ALPR em cenários de captura reais.

A análise desses trabalhos revela dimensões complementares e, até o momento, não unificadas: Laroca et al. (2018) e Laroca et al. (2021) estabelecem o desempenho de referência para ALPR em estações de trabalho convencionais, porém sobre o padrão anterior ao Mercosul; Izidio, Ferreira e Barros (2018) comprovam a viabilidade do YOLO em plataformas embarcadas de baixo custo, também sobre o padrão anterior, valendo-se de dados sintéticos para suprir a escassez de amostras reais; e Ribeiro et al. (2019) avançam na adaptação ao padrão vigente, mas sem completar o *pipeline* de reconhecimento. A lacuna identificada reside, portanto, na ausência de uma arquitetura de *software* que unifique essas três dimensões: desempenho dinâmico, reconhecimento completo no padrão Mercosul e viabilidade em infraestrutura convencional, em uma solução integrada e voltada especificamente às particularidades operacionais da gestão condominial de pequeno e médio porte.

Com base nesse cenário, este trabalho parte da hipótese de que a integração dessas tecnologias em um sistema ALPR de ponta a ponta pode preencher essa lacuna, superando as limitações dos métodos tradicionais em um contexto de restrições operacionais. Para orientar a investigação, formularam-se os seguintes questionamentos: como integrar as tecnologias de visão computacional em uma solução unificada e acessível para condomínios de pequeno e médio porte? Qual a eficácia de um sistema ALPR baseado em YOLO para o padrão Mercosul em infraestrutura convencional? De que forma a automação do reconhecimento de placas pode mitigar as falhas inerentes aos controles manuais?

Este trabalho propõe uma arquitetura de *software* ALPR que uni-

fica detecção baseada em YOLOv8, reconhecimento de caracteres por modelo treinado especificamente para o padrão Mercosul e lógica de controle de acesso, desenvolvida e validada sob as restrições operacionais de condomínios residenciais de pequeno e médio porte. O objetivo geral consiste em avaliar a eficácia de um sistema de controle de acesso veicular baseado em inteligência artificial para o Reconhecimento Automático de Placas Veiculares, visando superar as limitações dos métodos tradicionais e oferecer maior precisão, agilidade e confiabilidade ao processo.

Para alcançar esse objetivo, definem-se três objetivos específicos: desenvolver um protótipo integrado em Python, adaptando o modelo YOLOv8 para o padrão Mercosul em contexto de infraestrutura convencional; validar a eficácia do sistema em cenários que simulem a infraestrutura condominial; e analisar o desempenho do modelo por meio de métricas de acurácia, precisão, revocação e FPS, confrontando os resultados com os valores reportados pelos trabalhos correlatos.

A relevância do tema é respaldada pelo aumento das taxas de crimes contra o patrimônio no Brasil, conforme dados do Anuário Brasileiro de Segurança Pública (Fórum Brasileiro de Segurança Pública, 2023), e pelo expressivo número de condomínios no país. O desenvolvimento de um sistema ALPR baseado em inteligência artificial justifica-se por aliar praticidade e segurança, com benefícios diretos que incluem a redução do tempo de liberação de veículos e o aumento da segurança perimetral. A viabilidade do protótipo é reforçada pela adoção da linguagem Python e do algoritmo YOLO (Redmon et al., 2016), cuja integração nativa com *frameworks* de aprendizado profundo viabiliza uma solução de rápida prototipagem para o problema de acesso condominial.

2 MATERIAIS E MÉTODOS

Esta seção descreve a metodologia e as etapas operacionais adotadas no desenvolvimento do sistema ALPR voltado à segurança condominial. O trabalho consistiu na implementação de uma solução de *software* baseada em visão computacional, com o propósito de avaliar se um sistema integrado, executado em estação de trabalho convencional, pode reconhecer placas do padrão Mercosul com precisão e velocidade compatíveis ao contexto de controle de acesso em tempo real.

2.1 ENQUADRAMENTO METODOLÓGICO

Quanto à natureza, a pesquisa classificou-se como aplicada, pois objetivou gerar conhecimento direcionado à solução de um problema específico: o controle de acesso veicular em portarias de condomínios de pequeno e médio porte. Quanto à abordagem, caracterizou-se como quantitativa, uma vez que os resultados foram expressos em métricas estatísticas (percentuais de acerto, tempo de processamento em milissegundos e quadros por segundo, FPS) para mensurar o desempenho do modelo desenvolvido. Quanto aos procedimentos técnicos, tratou-se de uma pesquisa experimental e bibliográfica: bibliográfica por fundamentar-se em trabalhos anteriores sobre ALPR e redes neurais convolucionais (Laroca et al., 2018; Izidio, Ferreira e Barros, 2018; Ribeiro et al., 2019); e experimental por envolver a manipulação direta de variáveis de entrada, especificamente imagens de veículos reais com placas do padrão Mercosul, em um protótipo controlado, para observar e mensurar os efeitos produzidos pelo sistema de detecção e reconhecimento.

2.2 DESCRIÇÃO DO OBJETO DE ESTUDO

O objeto de estudo consistiu em placas veiculares brasileiras do padrão Mercosul, vigente no país desde 2018, caracterizadas por fundo branco, faixa azul superior com a inscrição “BRASIL” e o símbolo do Mercosul, e por sequência alfanumérica composta por três letras, um dígito, uma letra e dois dígitos (por exemplo: ABC1D23).

Para o treinamento do modelo de reconhecimento de caracteres, foi utilizado o *dataset* msipl/pr-hlm2d, disponibilizado publicamente na plataforma Roboflow sob licença de uso acadêmico. Esse conjunto foi selecionado por ser composto por 9.330 imagens de placas veiculares completas fotografadas em carros reais, em condições variadas de ângulo, iluminação e distância, distribuídas em 38 classes (dígitos de 0 a 9 e letras de A a Z, além das variantes *i* e *o*). Diferentemente de *datasets* compostos por caracteres isolados, este conjunto contém fotografias de veículos em cenários reais, o que confere ao modelo maior capacidade de generalização para condições condominiais. O conjunto foi dividido em 6.536 imagens para treinamento, 1.862 para validação e 932 para teste.

2.3 PROCEDIMENTOS DE DESENVOLVIMENTO

O desenvolvimento do sistema foi conduzido em etapas sequenciais, descritas a seguir em ordem cronológica.

2.3.1 Configuração do Ambiente de Desenvolvimento

O ambiente de desenvolvimento foi configurado no sistema operacional Microsoft Windows 11, com Python 3.11 e ambiente virtual isolado (*virtualenv*) para gerenciamento de dependências. A aceleração por GPU foi habilitada por meio do *framework* PyTorch (Paszke et al., 2019) com suporte a CUDA 12.8, compatível com a placa gráfica NVIDIA GeForce RTX 3060 (12 GB de VRAM) disponível na estação de trabalho. A disponibilidade da GPU foi verificada programaticamente antes do início dos experimentos. O código-fonte foi versionado com Git e hospedado em repositório remoto no GitHub, permitindo rastreabilidade das alterações ao longo do desenvolvimento.

2.3.2 Treinamento do Modelo de Reconhecimento de Caracteres

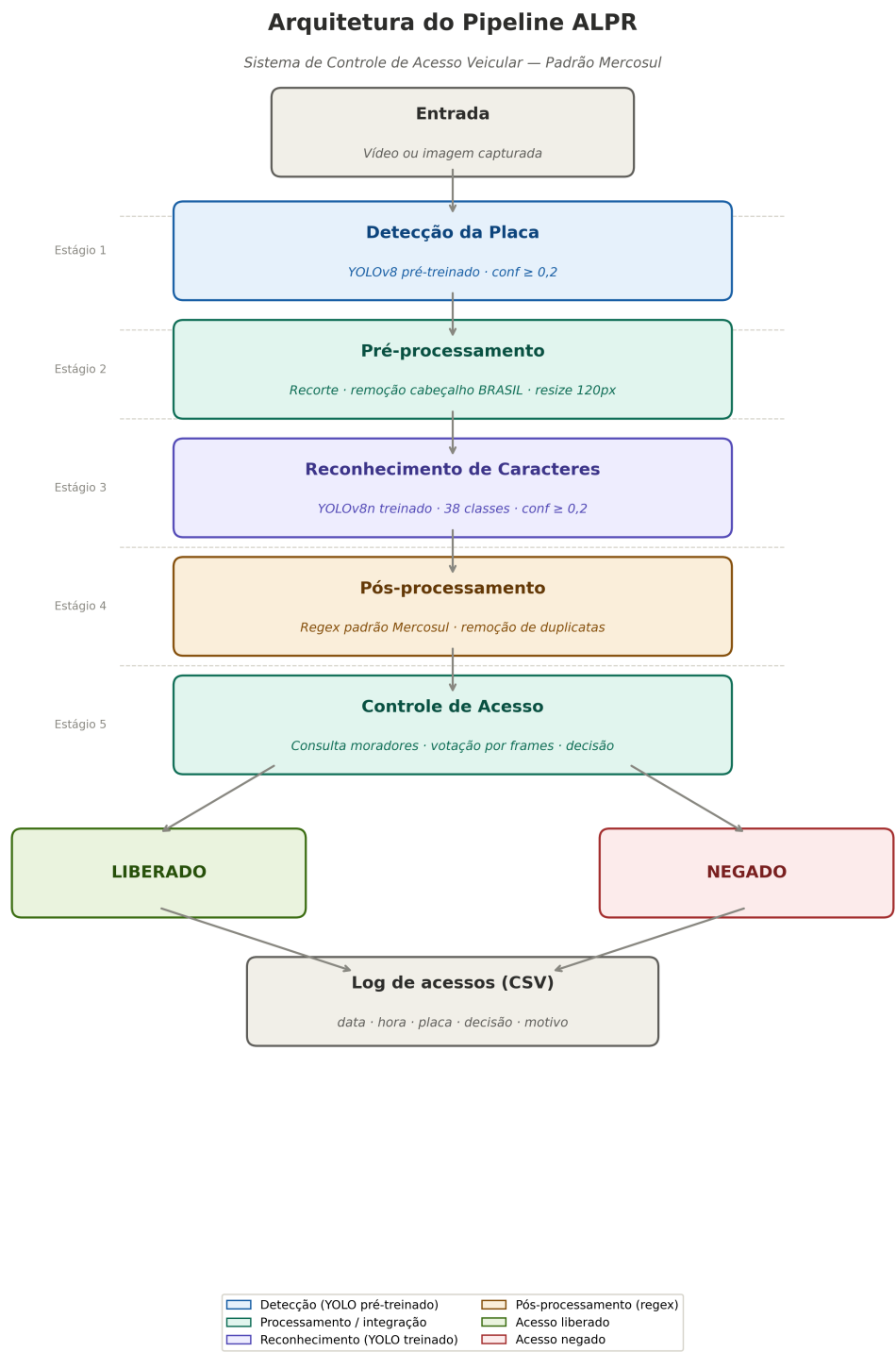
O modelo de reconhecimento de caracteres foi treinado a partir da arquitetura YOLOv8n (*nano*), com aproximadamente três milhões de parâmetros, escolhida por sua relação favorável entre desempenho e custo computacional (Jocher, Chaurasia e Qiu, 2023). O treinamento utilizou as 6.536 imagens do conjunto de treinamento do *dataset* msipl, com validação sobre as 1.862 imagens do conjunto de validação. Foram configuradas 50 épocas com mecanismo de parada antecipada (*early stopping* dez épocas), *batch* de 16 imagens, resolução de entrada de 640×640 pixels e otimizador Adam. O treinamento foi concluído em aproximadamente uma hora na estação de trabalho com GPU.

2.3.3 Implementação do *Pipeline* de Detecção e Reconhecimento

O núcleo do sistema foi implementado como um *pipeline* de cinco estágios sequenciais, executados para cada imagem ou quadro de vídeo processado. A Figura 1 ilustra a arquitetura completa do *pipeline* desenvolvido: (1) detecção da placa na imagem de entrada por modelo YOLOv8 pré-treinado; (2) pré-processamento do recorte, com remoção do cabeçalho “BRASIL” e padronização de dimensões; (3) reconhecimento individual dos caracteres pelo modelo YOLOv8n treinado para o padrão Mercosul; (4) pós-processamento com remoção de duplicatas e validação por expressão regular; e (5) controle de acesso com consulta ao cadastro de moradores e emissão da decisão LIBERADO ou NEGADO, registrada automaticamente em log.

No primeiro estágio, um modelo YOLOv8 pré-treinado para detecção de placas veiculares localizou a região de interesse (*bounding box*) dentro da imagem de entrada, com limiar de confiança mínimo de 20%

Figura 1 - Arquitetura do *pipeline* ALPR proposto com os cinco estágios de processamento



Fonte: Elaborado pelo autor.

(conf = 0.2).

No segundo estágio, a região detectada foi recortada com uma margem de dez pixels em cada borda, e a faixa superior correspondente a

30% da altura do recorte, onde se localiza o cabeçalho “BRASIL”, foi removida antes da etapa seguinte. O recorte resultante foi redimensionado para altura padronizada de 120 pixels, mantendo a proporção original.

No terceiro estágio, o modelo de caracteres treinado foi aplicado sobre o recorte pré-processado, com limiar de confiança mínimo de 20% ($conf = 0.20$), para detectar e classificar cada caractere individualmente. Os caracteres detectados foram ordenados pela posição horizontal (da esquerda para a direita) e concatenados para compor a leitura final da placa.

No quarto estágio, aplicou-se uma etapa de pós-processamento baseada em expressões regulares para validar e corrigir a sequência reconhecida, removendo duplicatas consecutivas e extraíndo o padrão estrutural do Mercosul (LLLNLNN) por meio de busca por correspondência de padrão (*regex*).

No quinto estágio, a placa validada foi consultada na lista de moradores cadastrados, e a decisão de acesso foi emitida e registrada automaticamente no log do sistema, conforme detalhado na Subseção seguinte.

2.3.4 Sistema de Controle de Acesso

O *pipeline* de reconhecimento foi integrado a um sistema de controle de acesso capaz de processar vídeos em tempo real. O sistema consultava uma lista de moradores cadastrados em arquivo CSV, contendo identificador do veículo, nome do morador, número do apartamento e status de atividade, e tomava a decisão de liberar ou negar a entrada do veículo com base na correspondência exata entre a placa reconhecida e o cadastro. A decisão, juntamente com os dados do morador e o registro de horário, foi armazenada automaticamente em arquivo de log para fins de auditoria, observando-se que em ambiente de produção os dados pessoais identificáveis devem ser tratados em conformidade com a Lei Geral de Proteção de Dados (LGPD). O sistema implementou ainda um filtro de validação que rejeitou leituras com formato inválido, evitando o registro de sequências espúrias no log de acessos.

Para mitigar leituras inconsistentes causadas por quadros desfocados ou parcialmente obstruídos, implementou-se um mecanismo de confirmação por votação: a placa foi considerada confirmada somente quando a mesma sequência de caracteres foi lida de forma consistente nos cinco quadros mais recentes dentro de uma janela deslizante de dez leituras.

2.3.5 Testes em Lote e Registro de Resultados

Para possibilitar a avaliação sistemática do sistema, foi desenvolvido um *script* de teste em lote que processou as 1.862 imagens do conjunto de validação do *dataset* msipl, comparando a leitura produzida pelo sistema com o gabarito real (*ground truth*) definido pelos rótulos do conjunto de dados. Os resultados foram registrados automaticamente em arquivo CSV, servindo de base para o cálculo das métricas de desempenho. Cabe destacar que, durante a análise manual dos resultados, identificou-se que parte dos erros computados decorreu de inconsistências nas próprias anotações do *dataset*, nas quais o gabarito apresentava caracteres incorretos em relação à placa visível na imagem, o que sugere que a acurácia mensurada pode estar subestimada.

2.4 ANÁLISE DOS DADOS

A análise de desempenho do sistema foi realizada sobre o conjunto de validação, composto exclusivamente por amostras não utilizadas durante o treinamento do modelo. Os dados de saída foram confrontados com o gabarito real, e as seguintes métricas foram calculadas: acurácia de leitura completa, definida como a proporção de placas lidas de forma idêntica ao gabarito em todos os sete caracteres; precisão (*precision*), que mede a proporção de detecções corretas entre todas as detecções realizadas; revocação (*recall*), que mede a proporção de caracteres corretamente detectados em relação ao total existente; mAP50, que corresponde à precisão média calculada com limiar de sobreposição de 50% entre a caixa detectada e o gabarito; mAP50-95, versão mais restritiva que calcula a média do mAP em limiares de sobreposição variando de 50% a 95%; e FPS, medido como o inverso do tempo médio de inferência por imagem, para avaliar a viabilidade operacional em tempo real. A comparação desses resultados com os valores reportados pelos trabalhos correlatos é apresentada na Seção 3.3.

2.5 RECURSOS UTILIZADOS

Os experimentos foram conduzidos em estação de trabalho convencional com processador Intel Core i5, 16 GB de memória RAM, placa gráfica NVIDIA GeForce RTX 3060 com 12 GB de VRAM e suporte à tecnologia CUDA, sistema operacional Microsoft Windows 11 e dispositivo de captura de vídeo smartphone iPhone. O ambiente de desenvolvimento foi composto pelas seguintes ferramentas: Python 3.11, PyTorch 2.x com

CUDA 12.8, biblioteca Ultralytics para YOLOv8, OpenCV 4.x (Bradski, 2000), Git para controle de versão com repositório hospedado no GitHub e Visual Studio Code como ambiente de desenvolvimento integrado.

3 RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos nas etapas de treinamento do modelo de reconhecimento de caracteres e de avaliação do *pipeline* completo, seguidos de sua interpretação e comparação com os trabalhos correlatos.

3.1 DESEMPENHO DO MODELO DE RECONHECIMENTO DE CARACTERES

O modelo YOLOv8n treinado sobre o *dataset* msipl apresentou métricas elevadas no conjunto de validação ao término do treinamento. A precisão atingiu 99,2% e a revocação 99,0%, enquanto o mAP50 alcançou 99,3%. O mAP50-95, versão mais restritiva que calcula a média do mAP em limiares de sobreposição entre 50% e 95%, atingiu 79,3%, valor esperado para esse critério dado o maior grau de exigência no alinhamento das caixas detectadas. Esses resultados indicam que o modelo apresentou boa capacidade discriminativa para as 38 classes treinadas, com desempenho consistente mesmo em condições variadas de iluminação, ângulo e distância, características inerentes ao *dataset* utilizado.

3.2 DESEMPENHO DO PIPELINE COMPLETO

O teste em lote sobre as imagens do conjunto de validação produziu os resultados consolidados nas Tabelas 1 a 4. O sistema obteve acurácia de leitura completa de 92,37% e velocidade de 25,85 quadros por segundo, acima do limiar mínimo de 25 FPS considerado adequado para aplicações em tempo real.

A configuração final adotada nos testes utilizou limiar de confiança de 20% para a detecção da placa (`conf_placa = 0.2`) e de 20% para o reconhecimento de caracteres (`conf_chars = 0.20`).

Optou-se por não apresentar a matriz de confusão completa do modelo de caracteres, uma vez que suas 38 classes resultariam em uma matriz 38×38 de difícil leitura e com a maioria das células nulas. Em seu lugar, os resultados são decompostos na distribuição dos erros por categoria (Tabela 2), nas confusões específicas entre os pares de caracteres mais frequentes (Tabela 3), que concentram a informação mais relevante

da diagonal secundária, e na acurácia por posição na placa (Tabela 4).

Tabela 1 - Desempenho geral do sistema de reconhecimento

Métrica	Valor
Total de imagens testadas	1.862
Reconhecimentos corretos	1.720
Reconhecimentos incorretos	142
Taxa de acerto (acurácia)	92,37%
Taxa de erro	7,63%

Fonte: Elaborado pelo autor.

A análise dos 142 erros revelou que a maioria (66,9%) decorre da inserção de caracteres extras, geralmente pela detecção duplicada de um mesmo caractere ou pela captação de ruídos visuais como parafusos e bordas da placa (Tabela 2). Notavelmente, nenhuma placa deixou de ser detectada (0,0% de não detecção), o que indica que o detector se mostrou eficiente frente às variações de ângulo, iluminação e distância presentes no *dataset*.

Tabela 2 - Distribuição dos erros por categoria

Categoria do erro	Casos	% dos erros	% do total
Caracteres extras	95	66,9%	5,1%
Caracteres faltando	6	4,2%	0,3%
Caracteres trocados	41	28,9%	2,2%
Placa não detectada	0	0,0%	0,0%
Total	142	100,0%	7,6%

Fonte: Elaborado pelo autor.

Entre as substituições, destaca-se a confusão entre a letra $\mathbb{L}$ e o caractere $\mathbb{J}$ (Tabela 3), ambiguidade visual decorrente da similaridade morfológica entre esses caracteres em determinadas fontes e ângulos de captura.

A análise posicional (Tabela 4) indica acurácia elevada e homogênea em todas as posições, variando entre 95,5% e 98,1%, sem a queda progressiva nas posições finais típica de modelos treinados com caracteres isolados. Essa distribuição uniforme pode ser atribuída ao uso de imagens de placas completas em carros reais durante o treinamento.

Tabela 3 - Principais confusões entre caracteres

Caractere real	Lido como	Ocorrências
L	J	5
O	D	3
5	E	2
D	O	2
N	M	2
M	W	2
1	B	1
O	E	1
2	Z	1
F	E	1

Fonte: Elaborado pelo autor.

Tabela 4 - Acurácia por posição do caractere (padrão Mercosul ABC1D23)

Posição	Tipo	Acertos	Total	Acurácia
1	Letra	1.820	1.855	98,1%
2	Letra	1.804	1.855	97,3%
3	Letra	1.791	1.855	96,5%
4	Número	1.785	1.855	96,2%
5	Letra	1.772	1.855	95,5%
6	Número	1.776	1.855	95,7%
7	Número	1.774	1.855	95,6%

Fonte: Elaborado pelo autor.

A Figura 2 ilustra um caso de reconhecimento bem-sucedido, em que o sistema leu corretamente todos os sete caracteres da placa. Em contraste, a Figura 3 exemplifica um erro por caractere extra, categoria mais frequente nos erros observados.

3.3 COMPARAÇÃO COM OS TRABALHOS CORRELATOS

A Tabela 5 sintetiza a comparação entre os resultados obtidos e os valores reportados pelos trabalhos correlatos.

Tabela 5 - Comparação de desempenho entre o sistema proposto e os trabalhos correlatos

Trabalho	Acurácia	FPS	Padrão de placa
Laroca et al. (2018)	93,53%	47	Anterior (LLLNNNN)
Izidio, Ferreira e Barros (2018)	97,00%	–	Anterior (LLLNNNN)
Este trabalho	92,37%	25,85	Mercosul (LLLNLNN)

Fonte: Elaborado pelo autor.

A comparação direta com Laroca et al. (2018) e Izidio, Ferreira e Barros (2018) não é metodologicamente equivalente, dado que ambos os

Figura 2 - Exemplo de reconhecimento correto realizado pelo sistema



Fonte: Elaborado pelo autor.

Figura 3 - Exemplo de erro por inserção de caractere extra



Fonte: Elaborado pelo autor.

trabalhos avaliaram seus sistemas sobre placas do padrão anterior ao Mercosul (formato cinza, sequência LLLNNNN), enquanto este trabalho focou exclusivamente no padrão vigente desde 2018 (formato branco, sequência LLLNLNN), que apresenta maior complexidade visual pelo acréscimo de um caractere alfabético na quarta posição. Ainda assim, a acurácia de 92,37% obtida neste trabalho é próxima à de 93,53% reportada por Laroca et al. (2018), e a velocidade de 25,85 FPS indica viabilidade operacional para

controle de acesso em tempo real, operando acima do limiar mínimo de 25 FPS considerado adequado para esse tipo de aplicação. Cabe ressaltar que não foram identificados trabalhos correlatos que reportem acurácia de reconhecimento completo de caracteres especificamente para o padrão Mercosul: Ribeiro et al. (2019) restringiram-se à etapa de detecção, sem avançar para o reconhecimento dos caracteres, o que reforça o ineditismo da avaliação conduzida neste trabalho.

O desempenho do modelo de reconhecimento de caracteres, com precisão de 99,2% e revocação de 99,0%, sugere que o modelo aprendeu as características visuais dos caracteres do padrão Mercosul em condições reais de captura. A velocidade de 25,85 FPS foi obtida em estação de trabalho convencional com GPU NVIDIA RTX 3060, sem recorrer a plataformas embarcadas especializadas.

3.4 TESTES EM VÍDEO REAL

Complementarmente aos testes em lote, foram realizados testes com vídeos capturados em ambiente real com câmera de *smartphone* (iPhone), a fim de avaliar o comportamento do protótipo em condições não controladas. Durante a fase de testes, avaliou-se o posicionamento da câmera em relação à trajetória do veículo, testando-se um posicionamento levemente deslocado à esquerda em comparação a um posicionamento mais centralizado em relação à faixa de circulação; optou-se pelo posicionamento centralizado nos vídeos finais, por proporcionar enquadramento mais consistente da placa ao longo de toda a aproximação do veículo. Foram gravados três vídeos correspondentes a três cenários distintos de controle de acesso: morador com cadastro ativo (decisão esperada: LIBERADO), morador com cadastro inativo (decisão esperada: NEGADO) e visitante não cadastrado (decisão esperada: NEGADO).

O sistema apresentou funcionamento adequado nos três cenários avaliados, emitindo as decisões esperadas em todos os casos. A placa foi corretamente identificada em 456 dos 1.394 quadros do primeiro vídeo (32,7%; morador ativo, decisão LIBERADO), em 277 dos 1.038 quadros do segundo vídeo (26,7%; morador inativo, decisão NEGADO) e em 240 dos 752 quadros do terceiro vídeo (31,9%; visitante não cadastrado, decisão NEGADO). Embora a taxa de acerto por quadro individual seja da ordem de 30% inferior à acurácia do conjunto de validação, devido a desfoque de movimento, ângulo e iluminação não controlados o mecanismo de confirmação por votação (Seção 2.3.4), que exige cinco leituras consis-

tes numa janela de dez quadros, garantiu a decisão correta nos três cenários. O sistema registrou automaticamente cada decisão no log (Tabela 6). Ressalta-se que esses testes representam uma validação funcional inicial do *pipeline*, não uma avaliação estatística ampla em ambiente de produção.

Tabela 6 - Extrato do log de acessos gerado pelo sistema nos testes em vídeo real

Placa	Decisão	Morador	Apto	Motivo
MHO8C27	LIBERADO	Morador A	101	Acesso autorizado
FEF5A88	NEGADO	Morador B	102	Cadastro inativo
MIU4838	NEGADO	—	—	Não cadastrado

Fonte: Elaborado pelo autor.

A principal contribuição deste trabalho em relação aos estudos correlatos reside na integração de um *pipeline* ALPR completo, abrangendo detecção, reconhecimento e controle de acesso com registro em log, voltado especificamente ao padrão Mercosul e às restrições operacionais do ambiente condominial. Enquanto Laroca et al. (2018) e Izidio, Ferreira e Barros (2018) demonstraram a eficácia do YOLO para placas do padrão anterior, e Ribeiro et al. (2019) abordaram a detecção do padrão Mercosul sem avançar para o reconhecimento completo dos caracteres, este trabalho unifica essas contribuições em uma solução funcional e reproduzível, avaliada tanto em *dataset* público quanto em vídeos capturados em ambiente real.

Como limitação, destaca-se que a acurácia de 92,37% pode não ser suficiente para cenários de alta segurança em que se exige correspondência exata da placa. O mecanismo de confirmação por votação mitiga parte desse problema ao exigir consistência entre leituras consecutivas antes de emitir uma decisão. O aprimoramento do conjunto de treinamento com imagens capturadas especificamente em condições de portaria condominial representa o caminho mais direto para elevar a acurácia em trabalhos futuros.

Ressalta-se ainda que o modelo foi treinado exclusivamente sobre placas do padrão Mercosul; o *dataset* UFPR-ALPR, do padrão anterior, foi descartado por incompatibilidade de formato (Seção 2.2). O sistema, portanto, não reconhece veículos com placas do padrão anterior ao Mercosul, e estender o modelo para ambos os padrões é uma oportunidade para trabalhos futuros.

4 CONCLUSÃO

Este trabalho propôs e avaliou um sistema ALPR voltado ao controle de acesso em condomínios residenciais de pequeno e médio porte, com foco no padrão Mercosul. A hipótese central, de que a integração de técnicas de aprendizado profundo em uma arquitetura de *software* de ponta a ponta pode contribuir para superar as limitações dos métodos tradicionais de controle de acesso, foi sustentada pelos resultados obtidos nesta pesquisa.

O primeiro objetivo específico foi plenamente atingido com o desenvolvimento do *pipeline* de cinco estágios, treinado sobre imagens de placas reais em carros e executado em estação de trabalho convencional com GPU. O segundo objetivo foi alcançado: o protótipo operou em tempo real a 25,85 FPS e apresentou funcionamento adequado nos três cenários de controle de acesso testados em vídeo real, mediante mecanismo de confirmação por votação que garantiu a decisão correta nos três cenários mesmo diante de taxas de acerto por quadro individual da ordem de 30%. O terceiro objetivo foi cumprido com a análise das métricas obtidas: acurácia de 92,37%, próxima à de 93,53% reportada por Laroca et al. (2018) para o padrão anterior, em um problema de maior complexidade visual.

Entre as principais dificuldades enfrentadas, destaca-se a escassez de *datasets* públicos de placas Mercosul em condições reais de portaria, o que exigiu a adoção de um *dataset* composto por imagens de veículos em cenários variados. Identificou-se ainda a presença de inconsistências nas anotações do *dataset*, nas quais o gabarito apresentava caracteres incorretos em relação à placa visível na imagem, o que pode ter subestimado a acurácia real do sistema.

Como sugestões de trabalhos futuros, são propostas: a construção de um *dataset* de placas Mercosul capturadas especificamente em condições reais de portaria condominial; a investigação de arquiteturas mais leves, como YOLOv8s ou modelos quantizados, para avaliar a viabilidade do *pipeline* em plataformas embarcadas; e o desenvolvimento de um mecanismo de correspondência aproximada (*fuzzy matching*) com limiar configurável, para avaliar o equilíbrio entre segurança e taxa de falsos negativos em diferentes perfis de risco condominial.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que ferramentas de Inteligência Artificial generativa foram utilizadas exclusivamente como apoio auxiliar à elaboração deste trabalho, incluindo assistência na revisão linguística, estruturação textual e organização de ideias. Nenhuma ferramenta de IA foi utilizada para substituição da autoria intelectual, análise científica, interpretação dos resultados ou tomada de decisões acadêmicas. A responsabilidade integral pelo conteúdo, originalidade, veracidade das informações e conformidade com os princípios de integridade científica permanece integralmente atribuída aos autores.

REFERÊNCIAS

- ALMEIDA, R.; OLIVEIRA, D. et al. RodoSol-ALPR dataset: A challenging license plate recognition dataset acquired at toll stations. In: **Proceedings of the Conference on Graphics, Patterns and Images (SIBGRAPI)**. Foz do Iguaçu, Brazil: IEEE, 2018. Disponível em: <<https://github.com/RodoSol-ALPR/Dataset>>. Acesso em: 5 maio 2026.
- Associação Brasileira das Administradoras de Imóveis (ABADI). **Mercado Condominial no Brasil**. 2022. Website Institucional. Disponível em: <<https://abadi.com.br/>>. Acesso em: 3 maio 2026.
- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, v. 25, n. 11, p. 120–125, 2000. Disponível em: <<https://github.com/opencv/opencv>>. Acesso em: 6 maio 2026.
- DU, S. et al. Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 23, n. 2, p. 311–325, feb 2013. Disponível em: <<https://ieeexplore.ieee.org/document/62135199>>. Acesso em: 4 maio 2026.
- Fórum Brasileiro de Segurança Pública. **17º Anuário Brasileiro de Segurança Pública**. São Paulo: Fórum Brasileiro de Segurança Pública, 2023. Disponível em: <<https://forumseguranca.org.br/wp-content/uploads/2023/07/anuario-2023.pdf>>. Acesso em: 7 maio 2026.
- IZIDIO, D. M. F.; FERREIRA, A. P. A.; BARROS, E. N. S. An Embedded Automatic License Plate Recognition System using Deep Learning. In: **Anais do VIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)**. Salvador, Brazil: SBC, 2018. p. 50–57. Disponível em: <<https://sol.sbc.org.br/index.php/sbesc/article/view/10926>>. Acesso em: 5 maio 2026.

JOCHER, G.; CHAURASIA, A.; QIU, J. **Ultralytics YOLOv8**. Ultralytics, 2023. Disponível em: <<https://github.com/ultralytics/ultralytics>>. Acesso em: junho 2026. Disponível em: <<https://github.com/ultralytics/ultralytics>>.

LAROCA, R. et al. A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. In: **Proceedings of the International Joint Conference on Neural Networks (IJCNN)**. Rio de Janeiro, Brazil: IEEE, 2018. p. 1–8. Disponível em: <<https://doi.org/10.1109/IJCNN.2018.8489629>>. Acesso em: 4 maio 2026.

LAROCA, R. et al. Leveraging model fusion for improved license plate recognition. In: **Iberoamerican Congress on Pattern Recognition (CIARP)**. [S.l.: s.n.], 2023. p. 60–75. Disponível em: <<https://arxiv.org/abs/2309.04331>>. Acesso em: junho 2026.

LAROCA, R. et al. An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector. **IEEE Transactions on Intelligent Transportation Systems**, v. 22, n. 2, p. 1136–1149, 2021. Disponível em: <<https://arxiv.org/abs/1909.01754>>. Acesso em: 20 maio 2026.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, may 2015. Disponível em: <<https://www.nature.com/articles/nature14539>>. Acesso em: 6 maio 2026.

PASZKE, A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: **Advances in Neural Information Processing Systems 32**. Curran Associates, Inc., 2019. p. 8024–8035. Disponível em: <<https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>>. Acesso em: 6 maio 2026.

REDMON, J. et al. You Only Look Once: Unified, Real-Time Object Detection. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, 2016. p. 779–788. Disponível em: <<https://ieeexplore.ieee.org/document/7780460>>. Acesso em: 4 maio 2026.

RIBEIRO, V. et al. Brazilian Mercosur License Plate Detection: a Deep Learning Approach Relying on Synthetic Imagery. In: **Anais do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)**. Natal, Brazil: SBC, 2019. p. 177–184. Disponível em: <<https://sol.sbc.org.br/index.php/sbesc/article/view/10283>>. Acesso em: 5 maio 2026.

SELMİ, Z. et al. DELP-DAR system for license plate detection and recognition. **Pattern Recognition Letters**, v. 129, p. 213–223, 2020. Disponível em: <<https://arxiv.org/abs/1910.01853>>. Acesso em: junho 2026.