

EyeTruck: Sistema de Prevenção de Colisão com Viadutos e Pontes

EyeTruck: Collision Prevention System for Viaducts and Bridges

EyeTruck: Sistema de Prevención de Colisiones con Viaductos y Puentes

Murillo Castro De Jesus¹
murillo.jesus4@etec.sp.gov.br

Rafael Ferreira Lopes¹
rafael.lopes161@etec.sp.gov.br

Vinicius Rafael Bomfim Rodrigues¹
vinicius.rodrigues232@etec.sp.gov.br

Vitor Daisuke Iwamoto¹
vitor.iwamoto@etec.sp.gov.br

Jeferson Roberto de Lima¹
jeferson.lima17@etec.sp.gov.br

Palavras-chave:

Sistemas embarcados.
Altura máxima permitida.
Reconhecimento de
placas de trânsito.
Prevenção de colisões.

Keywords:

Embedded systems.
Maximum permitted
height.
Traffic sign recognition.
Collision prevention.

Palabras clave:

Sistemas empuados.
Altura máxima permitida.
Reconocimiento de placas
de tráfico.
Prevención de colisiones.

Apresentado em:

03 dezembro, 2025

Evento:

8º EnGeTec

Local do evento:

Fatec Zona Leste

Avaliadores:

Avaliador 1
Avaliador 2



Resumo:

Este artigo apresenta o EyeTruck, um sistema embarcado de apoio à prevenção de colisões de caminhões com viadutos e pontes por excesso de altura. Os objetivos desse projeto são detectar a sinalização R-15, extrair a altura máxima permitida e comparar com a altura cadastrada do veículo para alertar o condutor. A metodologia aborda pesquisa quantitativa, de caráter documental, utilizando dados secundários como notícias e relatórios setoriais para contextualização do problema e testes em ambientes controlados com o protótipo para validação funcional. A problemática envolve acidentes por excesso de altura que ocorrem recorrentemente, gerando riscos e custos à infraestrutura, como apontado por levantamentos públicos e setoriais. As principais conclusões mostram que em ambiente controlado, o protótipo identificou a placa R-15 com alta confiança, extraiu corretamente o valor de altura e emitiu alertas quando o limite foi excedido, indicando viabilidade técnica. A aplicação em cenários reais depende de ampliar a base de imagens e de testes de campo.

Abstract:

This article presents EyeTruck, an embedded system designed to support the prevention of truck collisions with viaducts and bridges due to excess height. The objectives of this project are to detect R-15 signs, extract the maximum permitted height, and compare it with the registered height of the vehicle in order to warn the driver. The methodology involves quantitative, documentary research, using secondary data such as news reports and sectoral studies to contextualize the problem, as well as tests in controlled environments with the prototype for functional validation. The issue addressed involves height-related accidents that occur recurrently, generating risks and costs for infrastructure, as indicated by public and sectoral surveys. The main findings show that, in a controlled environment, the prototype identified the R-15 sign with high confidence, correctly extracted the height value, and issued alerts when the limit was exceeded, indicating technical feasibility. Application in real-world scenarios depends on expanding the image database and conducting field tests.

Resumen:

Este artículo presenta EyeTruck, un sistema embebido para prevenir colisiones de camiones con viaductos y puentes por exceso de altura. Los objetivos del proyecto son detectar la señal R-15, extraer la altura máxima permitida y compararla con la altura registrada del vehículo para alertar al conductor. La metodología emplea investigación documental cuantitativa, utilizando datos secundarios como artículos de prensa e informes del sector para contextualizar el problema, y pruebas en entornos controlados con el prototipo para su validación funcional. El problema radica en los accidentes recurrentes por exceso de altura, que generan riesgos y costes para la infraestructura, según indican encuestas públicas y del sector. Las principales conclusiones muestran que, en un entorno controlado, el prototipo identificó la señal R-15 con alta precisión, extrajo correctamente el valor de la altura y emitió alertas al excederse el límite, demostrando su viabilidad técnica. Su aplicación en escenarios reales depende de la ampliación de la base de datos de imágenes y la realización de pruebas de campo.

¹Etec Zona Leste

1. Introdução

O EyeTruck é um dispositivo para prevenção de colisões de veículos de carga com viadutos e pontes, por meio do reconhecimento das placas R-15, que indicam a altura máxima permitida.

Para cumprir esse objetivo, o sistema é composto por hardware, câmera e uma aplicação móvel, sendo capaz de comparar a altura máxima informada na sinalização com a altura do caminhão, inserida previamente pelo motorista. Dessa forma, caso haja risco de colisão, alertas serão emitidos ao condutor, contribuindo para a segurança viária.

A problemática que motivou a realização deste projeto está relacionada à recorrência de acidentes e bloqueios de vias provocados por veículos de carga que ultrapassam a altura máxima permitida em trechos específicos. Embora exista uma lei que obriga a existência de sinalização indicando a altura máxima em viadutos e pontes, esses métodos não têm sido eficientes o bastante para evitar os acontecimentos. Além disso, pode-se observar que as estruturas citadas apresentam variações nas alturas, o que leva a situações em que o veículo consegue passar por uma via, mas não por outra.

Inicialmente, acreditava-se que a principal causa do problema fosse a falta de sinalização para alertar o motorista. No entanto, foi descoberto que é obrigatória a existência de sinalizações especificando a altura máxima permitida dos veículos, mas certas variáveis, como a falta de padronização no tamanho das estruturas, a desatenção e a falta de familiaridade dos motoristas com determinadas vias, contribuem para que os acidentes continuem ocorrendo.

Diante desse cenário, o desenvolvimento do EyeTruck justifica-se pela necessidade de um outro método de prevenção de acidentes, pela redução de custos relacionados a danos na infraestrutura viária e pela segurança tanto para o motorista do caminhão quanto para os demais motoristas, assim, contribuindo para uma viagem mais segura.

Quanto ao escopo, este projeto está delimitado à cidade de São Paulo e à sua região metropolitana, utilizadas como estudo de caso, devido ao tráfego intenso de veículos de grande porte e ao elevado número de infraestruturas, como viadutos e pontes, visando atender cidades com acesso à internet e grande concentração de estruturas que contenham restrições de altura.

A metodologia usada para evidenciar esse projeto é a quantitativa, de caráter documental com o uso de dados secundários como notícias e relatórios somados a testes controlados do protótipo.

Por fim, este artigo tem como objetivo apresentar o desenvolvimento do EyeTruck, um sistema de prevenção de colisões de caminhões com viadutos e pontes que integra um aplicativo móvel, um dispositivo embarcado baseado em Raspberry Pi, técnicas de visão computacional, detecção de objetos com YOLO, reconhecimento óptico de caracteres e armazenamento de dados em nuvem.

2. Fundamentação Teórica

Neste capítulo serão apresentados os principais conceitos, tecnologias e uma visão geral da problemática, tendo em vista melhor entendimento do contexto em que este trabalho foi desenvolvido.

2.1. Acidentes envolvendo caminhões e viadutos

Segundo notícia publicada pela Folha de S. Paulo (2019), ao menos 15 caminhões por mês ficam bloqueados por ultrapassarem a altura permitida em pontes e viadutos na cidade de São Paulo, totalizando 213 ocorrências entre janeiro de 2017 e outubro de 2018. Esses episódios geraram riscos a outros condutores e danos às infraestruturas de transporte.

Dados recentes reforçam a magnitude do problema. Segundo análise da Pamcary (2025), em 2024 a taxa de sinistros envolvendo caminhões nas rodovias brasileiras foi 4,5 vezes superior à de furtos de carga, e 36% desses episódios envolveram terceiros. Esses indicadores evidenciam o potencial de dano

humano e material associado a ocorrências com veículos de grande porte, justificando iniciativas tecnológicas voltadas à prevenção de colisões por excesso de altura, como o EyeTruck.

Ainda de acordo com levantamento da CET citado pela R7 (2019), a cada dois dias ocorria um acidente envolvendo caminhões com excesso de altura que não conseguiram passar por viadutos e pontes na capital paulista, chegando a 157 casos apenas entre janeiro e setembro, com destaque para 22 ocorrências em setembro, o que evidencia também impactos significativos no tráfego, como congestionamentos e longas filas de veículos.

2.2. Internet das Coisas e dispositivos embarcados

De acordo com Santos (2019), a Internet das Coisas é a conexão de dispositivos capazes de se comunicar entre si para realizar tarefas de forma autônoma, sem a necessidade de intervenção humana constante.

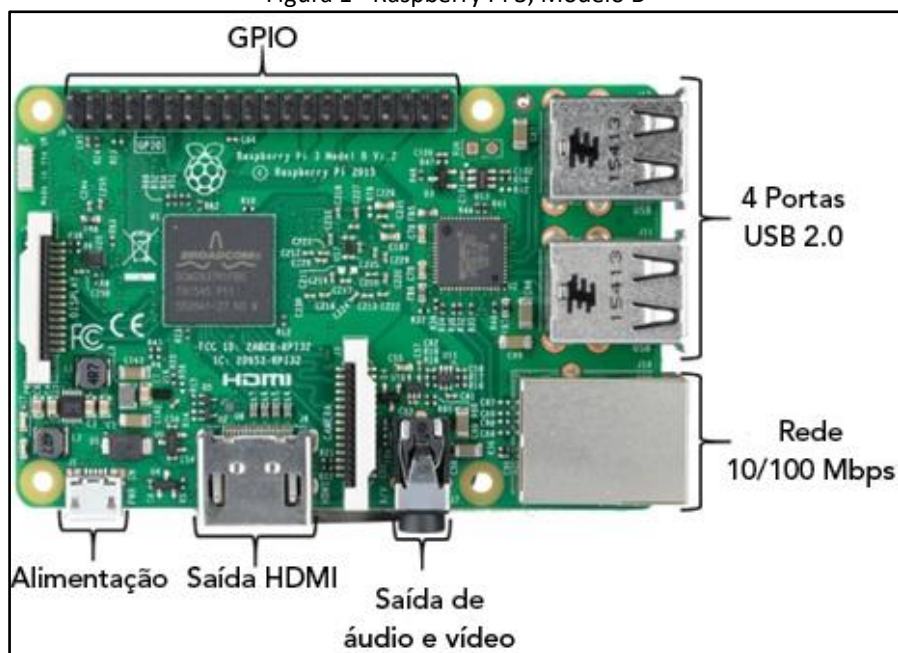
Carrion e Quaresma (2019) destacam que as primeiras experiências com dispositivos conectados remontam à década de 1990, como o experimento de uma torradeira que podia ser ligada e desligada pela internet, considerado um marco inicial do conceito de IoT.

Mais recentemente, Oliveira (2021) enfatiza que a Internet das Coisas vai além de conectar equipamentos à rede, buscando tornar as “coisas” inteligentes, capazes de coletar dados do ambiente e utilizá-los para tomar decisões ou apoiar outras aplicações.

Nessa perspectiva, plataformas embarcadas como o Raspberry Pi permitem transformar soluções de IoT em projetos reais. Conforme Ebermam et al. (2017), o Raspberry Pi surgiu como um computador de baixo custo voltado à educação, com o objetivo de aproximar estudantes de conceitos de hardware e software. Posteriormente, consolidou-se como um microcomputador versátil para diferentes áreas, podendo ser usado tanto por iniciantes quanto por especialistas em computação.

Oliveira, Nabarro e Zanetti (2018) apontam que existem diversos modelos de Raspberry Pi, que variam em capacidade de processamento, quantidade de interfaces disponíveis e recursos de conectividade, o que o torna adequado para aplicações de monitoramento e automação, como o sistema proposto neste trabalho.

Figura 1 - Raspberry Pi 3, Modelo B



Fonte: Oliveira, Nabarro e Zanetti (2018)

2.3. Visão computacional e detecção de objetos

Segundo Barelli (2018), visão computacional é o estudo de máquinas capazes de reconhecer elementos de um ambiente por meio de imagens capturadas por câmeras, sensores e outros dispositivos, permitindo que essas informações sejam processadas de acordo com os objetivos definidos pelo desenvolvedor.

Milano e Honorato (2014) lembram que, embora as primeiras pesquisas na área remontem à década de 1950, o avanço significativo da visão computacional está diretamente ligado ao desenvolvimento da inteligência artificial e ao melhor entendimento dos processos biológicos envolvidos na percepção visual.

Entre os modelos de detecção de objetos em tempo real, destaca-se o YOLO (You Only Look Once), criado por Joseph Redmon e Ali Farhadi em 2015. De acordo com Araújo (2022), seu diferencial está em processar a imagem inteira de uma só vez, dividindo-a em regiões e estimando, para cada uma, caixas delimitadoras e probabilidades de presença de objetos. Essa abordagem permite altas taxas de processamento, embora possa resultar em alguma perda de acurácia em comparação com métodos mais lentos.

2.4. Reconhecimento de caracteres em placas de trânsito

Com o objetivo de facilitar o desenvolvimento de modelos de aprendizado de máquina, o Facebook lançou em 2016 o PyTorch, uma biblioteca de código aberto baseada em Python (FLORENCIO, 2020).

Souza (2024) destaca que a flexibilidade dessa ferramenta, aliada ao suporte a GPUs, contribui para a construção de modelos dinâmicos e otimizados para tarefas de visão computacional. A partir dessa infraestrutura, foi desenvolvida a biblioteca EasyOCR, destinada ao reconhecimento óptico de caracteres (OCR).

De acordo com Cordeiro (2024), o EasyOCR é capaz de extrair textos presentes em imagens e documentos, oferecendo suporte a mais de 80 idiomas, o que o torna adequado para aplicações que demandam leitura automática de informações textuais em diferentes contextos.

2.5. Armazenamento de dados na nuvem

Gonçalves (2014) define banco de dados como um conjunto estruturado de informações relacionadas, organizado em tabelas para facilitar o armazenamento e a recuperação eficiente de dados.

Carvalho (2015) ressalta que a escolha e o projeto adequados dessa base impactam diretamente o desempenho e a manutenção dos sistemas ao longo do tempo. Nesse cenário, plataformas de nuvem surgem como alternativa para garantir escalabilidade e alta disponibilidade.

Machado (2021) aponta o Firebase, solução oferecida pelo Google, como uma plataforma robusta que integra autenticação, banco de dados e outros serviços voltados a aplicações web e móveis.

Segundo documentação oficial do Firebase (GOOGLE, 2025), um único projeto pode reunir múltiplos aplicativos clientes compartilhando a mesma base de dados, o que facilita a construção de sistemas distribuídos, como o EyeTruck.

3. Método

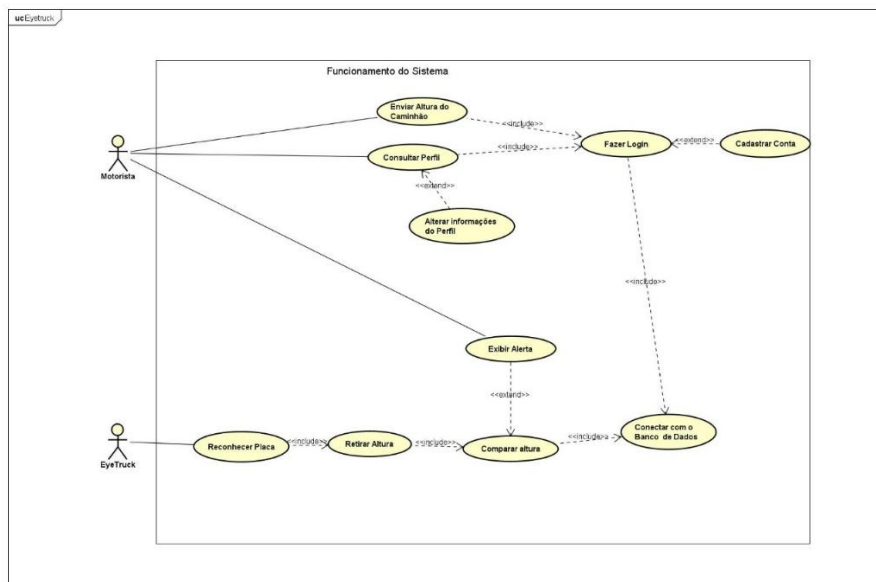
Nesse capítulo, será apresentado o desenvolvimento do EyeTruck, estruturado em etapas que vão desde a modelagem do sistema até a implementação do dispositivo embarcado e do aplicativo móvel.

A pesquisa adota abordagem quantitativa, de caráter documental, conforme Lakatos e Marconi (2021), utilizando dados secundários provenientes de notícias e relatórios para dimensionar o problema, além de testes controlados do protótipo para validação funcional.

3.1. Modelagem do sistema

Inicialmente, realizou-se o levantamento dos requisitos do sistema e a modelagem dos requisitos funcionais por meio de diagramas de caso de uso, que descrevem as interações entre o motorista e o sistema e as principais funcionalidades, como cadastro, login, envio da altura do caminhão, reconhecimento das placas e exibição de alertas. Esses casos de uso foram detalhados em tabelas de documentação, especificando fluxos principais, fluxos alternativos, pré-condições, pós-condições e regras de negócio associadas a cada funcionalidade.

Figura 2 - Diagrama de Caso de Uso



Fonte: Autoria Própria (2025)

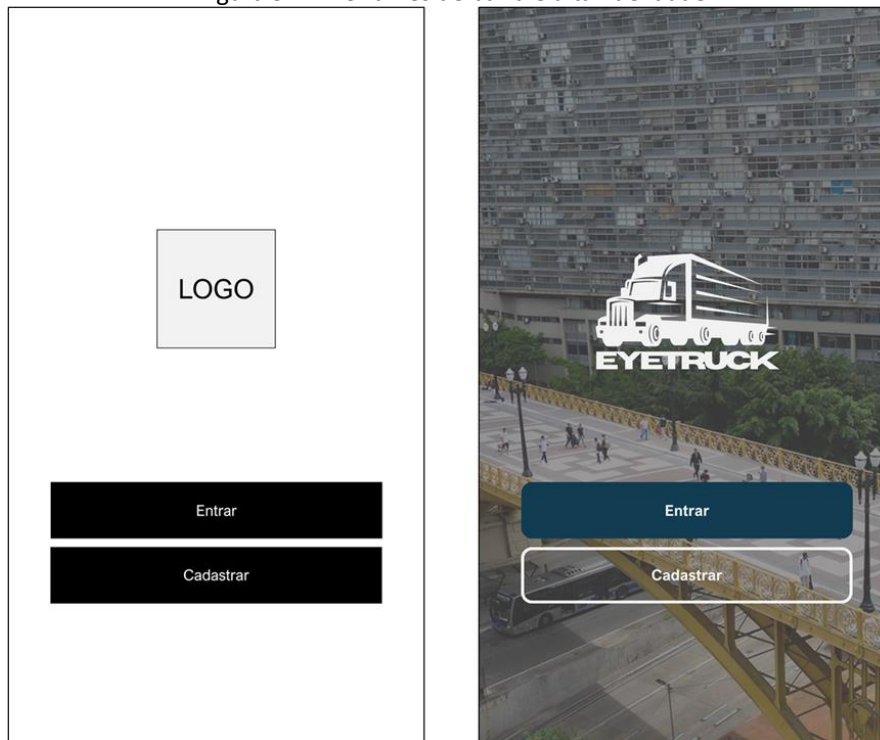
Na sequência, foram elaborados diagramas de atividade, sequência e máquina de estados, com o objetivo de representar o comportamento do sistema. Os diagramas de atividade foram utilizados para descrever, de forma visual, o fluxo de ações dos casos de uso, auxiliando no planejamento, na construção das funcionalidades do projeto e dividindo as etapas do processo para facilitar a compreensão (BOOCH, 2006).

O diagrama de sequência foi empregado para representar as mensagens trocadas entre os principais componentes do sistema, sendo eles o aplicativo móvel, o dispositivo embarcado e o banco de dados, durante a execução dos processos, explicitando a ordem em que essas interações ocorrem (GUEDES, 2018). Já os diagramas de máquina de estados foram utilizados para detalhar o comportamento dos objetos em resposta a eventos, mostrando as transições de estado ao longo do ciclo de execução das funcionalidades do EyeTruck (FOWLER, 2005).

3.2. Prototipação e desenvolvimento do aplicativo

Com a modelagem consolidada, passou-se à prototipação das interfaces do aplicativo móvel. Para isso, foram produzidos wireframes de baixa e alta fidelidade para telas de carregamento, início, busca de dispositivo, cadastro, login, tela inicial (home), edição de altura e edição de perfil. Os protótipos foram usados para validar a navegação, a disposição dos elementos de interface e a clareza das informações apresentadas ao usuário antes da implementação em código.

Figura 3 - Wireframes de baixa e alta fidelidade



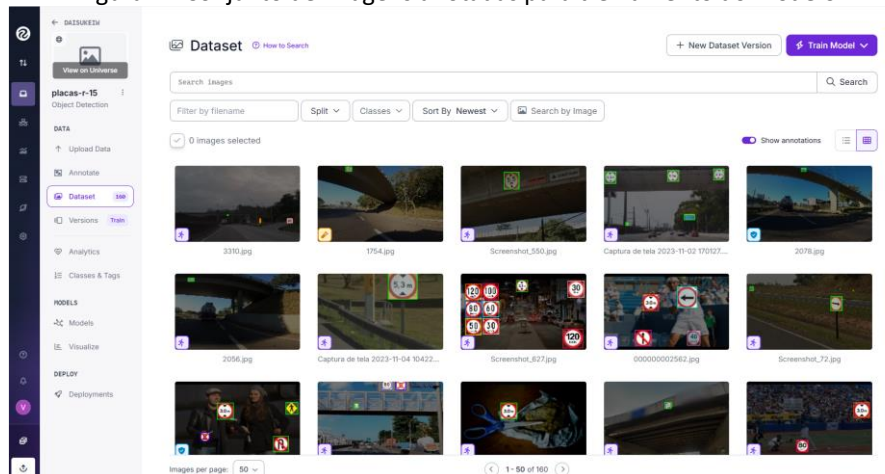
Fonte: Autoria Própria (2025)

Após a validação dos protótipos, o aplicativo foi implementado em React Native, que, de acordo com Galvão (2018), é uma tecnologia flexível, podendo ser utilizada em dispositivos Android e iOS. Para compilar o aplicativo, foi utilizado o Expo, um framework que simplifica a configuração do ambiente e colabora para a eficiência do processo de desenvolvimento (COELHO; ANTUNES, 2023), contemplando as mesmas telas definidas na etapa de prototipação. O app permite o cadastro e autenticação de usuários, o registro e a edição da altura do caminhão. Os dados são armazenados e sincronizados por meio do Firebase, que reúne as informações necessárias para que o hardware embarcado realize a comparação entre a altura informada e o valor extraído da placa de sinalização.

3.3. Desenvolvimento do hardware de detecção de placas

Para o desenvolvimento do hardware responsável pela detecção das placas, foi utilizado o Raspberry Pi, escolhido por apresentar uma boa relação custo-benefício e capacidade de processamento suficiente para executar o YOLO, uma das principais tecnologias utilizadas no projeto. Inicialmente, foi construído um dataset, e as imagens foram anotadas para o treinamento do modelo através da plataforma Roboflow. Ao todo, foi realizada a anotação de 160 imagens das placas R-15 e A-37, ambas relacionadas à limitação de altura.

Figura 4 - Conjunto de imagens anotadas para treinamento do modelo



Fonte: Autoria Própria (2025)

Após a etapa de anotação, o modelo YOLO foi treinado para a detecção de placas por meio do Python e com o apoio da biblioteca Ultralytics. O número inicial de 160 imagens foi considerado suficiente para a realização de testes, embora possa ser expandido em trabalhos futuros com o objetivo de aprimorar a acurácia do modelo.

Com o modelo de detecção treinado, o próximo passo foi integrar uma etapa de reconhecimento óptico de caracteres para extrair o valor numérico presente na sinalização. Para isso, adotou-se a biblioteca EasyOCR, responsável por identificar o texto na região da placa previamente detectada. Esse valor extraído é então comparado com a altura do caminhão cadastrada no banco de dados hospedado no Firebase, provido previamente pelo aplicativo móvel, de modo a identificar situações em que a altura do veículo supera o limite indicado na placa e, assim, alertar o motorista.

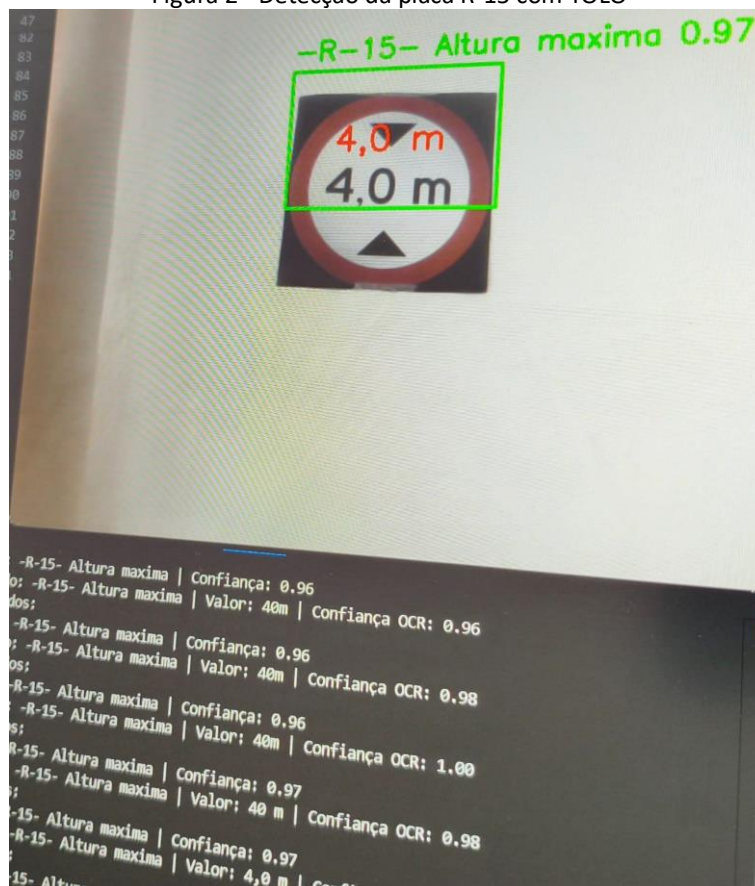
4. Resultados e Discussões

Nesta seção são apresentados os testes realizados e a discussão dos resultados obtidos.

4.1. Validação dos requisitos funcionais

Foram realizados testes em ambiente controlado com o sistema de detecção de placas. Utilizando uma placa R-15 impressa, simulando a sinalização de altura máxima permitida, o modelo YOLO foi capaz de identificar a placa na imagem capturada pela câmera acoplada ao Raspberry Pi, atribuindo alta confiança de 0,97 à detecção e delimitando a região correspondente. Na sequência, o OCR, implementado com a biblioteca EasyOCR, extraiu o valor numérico presente na placa, conforme ilustrado na Figura 5.

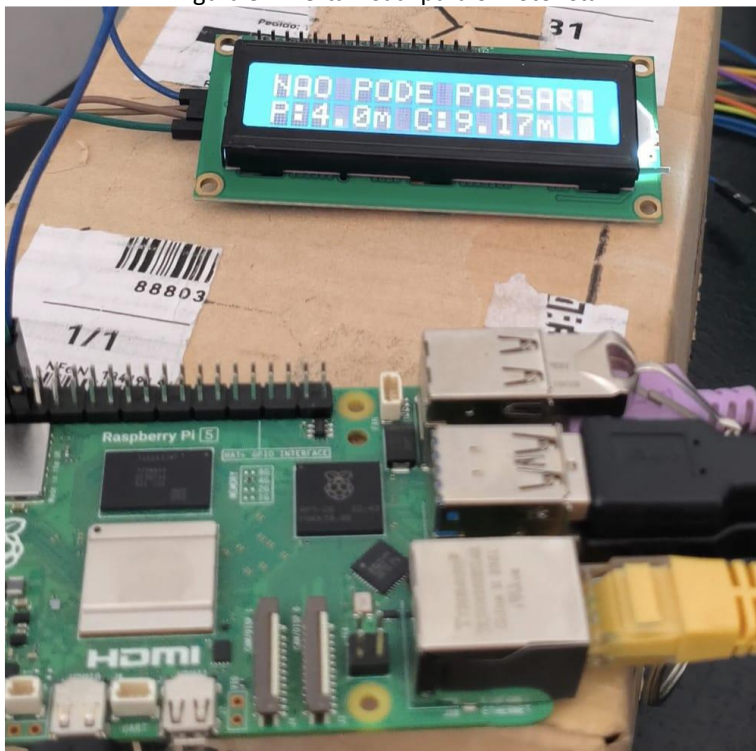
Figura 2 - Detecção da placa R-15 com YOLO



Fonte: Autoria Própria (2025)

Com o valor de altura máxima permitido extraído da placa e a altura do caminhão previamente cadastrada no aplicativo, o sistema realizou a comparação entre esses dois parâmetros. Como no cenário de teste a altura do veículo excedeu o valor da placa, foi exibido um alerta visual com a mensagem “NÃO PODE PASSAR”. A letra “P” representa o valor extraído da placa e “C” a altura do caminhão retirada do banco de dados, conforme apresentado na Figura 6. Em cenário inverso, em que a altura do caminhão fosse menor que o valor indicado na sinalização, a mensagem exibida ao motorista seria “PODE PASSAR”.

Figura 3 - Alerta visual para o motorista



Fonte: Autoria Própria (2025)

4.2. Viabilidade e limitações

Os resultados obtidos indicam que o EyeTruck é tecnicamente capaz de executar as etapas essenciais para apoiar o motorista na prevenção de colisões por excesso de altura, integrando aplicativo móvel, hardware embarcado e serviços em nuvem.

Por outro lado, algumas limitações importantes precisam ser consideradas. A quantidade de dados utilizada para o treinamento do modelo YOLO é consideravelmente pequena, com 160 imagens de placas R-15 e A-37, o que restringe a capacidade de detecção do modelo em cenários mais complexos, como diferentes condições de iluminação, variação de distância e ângulo da câmera em relação à sinalização. A precisão do OCR também pode ser impactada por fatores como reflexos, desgaste da placa ou baixa qualidade da imagem, o que tende a reduzir a confiança da leitura do valor de altura em ambientes reais.

Outro ponto relevante é que os testes realizados até o momento foram conduzidos em ambiente controlado, utilizando cenários simulados com placas impressas e condições favoráveis de luminosidade. Assim, ainda não é possível afirmar que o uso do EyeTruck, em larga escala, reduziria efetivamente o número de colisões de caminhões com viadutos e pontes, pois essa afirmação exigiria testes em cenários reais ao longo de um período prolongado. O que os resultados obtidos permitem concluir é que o protótipo é funcional e promissor como solução tecnológica, mas que sua efetividade prática na redução de acidentes depende de etapas adicionais de validação em ambiente real.

5. Considerações Finais

A incidência de acidentes envolvendo caminhões por excesso de altura representa um risco significativo para a segurança viária, tanto pelo porte desses veículos quanto pelo potencial de danos materiais e humanos envolvidos. Nesse contexto, este trabalho apresentou o EyeTruck, um sistema de prevenção de colisões de veículos de carga com viadutos e pontes que integra visão computacional, reconhecimento óptico de caracteres, aprendizado de máquina, hardware embarcado e um aplicativo móvel conectado a serviços em nuvem.

O sistema desenvolvido combina um aplicativo em que o motorista cadastra a altura do caminhão, um banco de dados em nuvem utilizando Firebase e um Raspberry Pi, responsável por reconhecer a placa R-15, extrair o valor de altura máxima permitida e compará-lo com a altura registrada no banco de dados. Nos testes realizados em ambiente controlado, utilizando uma placa R-15 impressa, o modelo YOLO foi capaz de identificar corretamente a sinalização com alta confiança e, em seguida, o módulo de OCR extraiu o valor numérico da placa. Quando a altura cadastrada do caminhão excedeu o limite indicado, o sistema emitiu um alerta visual ao motorista, evidenciando a viabilidade funcional do protótipo como sistema de apoio à decisão.

Apesar dos resultados alcançados, algumas limitações foram identificadas. A quantidade de dados utilizada no treinamento do modelo de detecção é consideravelmente pequena, o que restringe a capacidade de detecção em cenários mais complexos, como diferentes condições de iluminação, distâncias e ângulos de captura da placa. A precisão do OCR também pode ser afetada por reflexos, desgaste da sinalização e baixa qualidade da imagem, comprometendo a confiança da leitura da altura em ambientes reais. Somando isso ao fato de que os testes foram conduzidos apenas em ambiente controlado, o que impede, neste momento, afirmar que o uso do EyeTruck em larga escala reduziria de forma comprovada o número de colisões em viadutos e pontes.

Em trabalhos futuros, destaca-se a necessidade de ampliar a base de dados com imagens de placas R-15 e de outras sinalizações relacionadas à limitação de altura, capturadas em diferentes contextos e condições climáticas, de modo a aumentar a robustez do modelo de detecção e a precisão do OCR. Também se propõe a realização de testes em vias urbanas e rodovias, com acompanhamento dos resultados ao longo do tempo.

Em síntese, o EyeTruck configura-se como uma solução tecnológica promissora para apoiar a prevenção de colisões de caminhões por excesso de altura, contribuindo potencialmente para a redução de acidentes e de custos associados a danos estruturais e materiais. Embora ainda se encontre em estágio de protótipo, o sistema demonstra potencial para ser aperfeiçoado e, futuramente, aplicado em contextos reais, reforçando sua relevância prática e social.

Referências

ARAÚJO, Aline Moura. **Detecção e Destaque em Vídeo de Objetos utilizando YOLO**. 2022. 69 p. Dissertação (Programa de Pós-Graduação em Informática) - Universidade Federal da Paraíba, João Pessoa, 2022.

BARELLI, Felipe. **Introdução à Visão Computacional: Uma abordagem prática com Python e OpenCV**. 1. São Paulo: Casa do Código, 2018.

CARRION, Patricia; QUARESMA, Manuela. **Internet da Coisas (IoT): Definições e aplicabilidade aos usuários finais**. Human Factors in Design, Florianópolis, v. 8, n. 15, p. 049–066, 2019. DOI: 10.5965/2316796308152019049. Disponível em: <https://www.revistas.udesc.br/index.php/hfd/article/view/2316796308152019049>.

CARVALHO, Vinícius. **MySQL: Comece com o principal banco de dados open source do mercado**. 1. Ed. São Paulo: Casa do Código, 2015.

COELHO, Pedro Citadin; ANTUNES, Luciano. **Aplicação móvel voltada ao compartilhamento de informações acerca de praias integrando a ferramenta Google Maps e a Biblioteca React Native**. 2023. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade do Extremo Sul Catarinense, UNESC, Criciúma, 2023.

EBERMAM, Elivelto; PESENTE, Guilherme Moraes; RIOS, Renan Osório; PULINI, Igor Carlos. **Programação para leigos com Raspberry Pi**. Vitória: Edifes; João Pessoa: Editora IFPB, 2017.

GALVÃO, Pedro Sereno. **Comprehensive Repository Analysis of Mobile Projects Built with React Native**. 2018. 28 f. Dissertação (Programa de Graduação em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2018.

GOOGLE. **Firebase**. Disponível em: <https://firebase.google.com/>. Acesso em: 11 ago. 2025.

GONÇALVES, Eduardo. **SQL: Uma abordagem para bancos de dados Oracle**. 1. ed. São Paulo: Casa do Código, 2014.

LAKATOS, Eva Maria; MARCONI, Marina De Andrade. **Fundamentos de Metodologia Científica**. 9. ed. São Paulo: GEN Atlas, 2021.

MACHADO, Kheronn Kennedy. **Angular 11 e Firebase: Construindo uma aplicação integrada com a plataforma do Google**. São Paulo: Casa do Código, 2021.

MILANO, Danilo de; HONORATO, Luciano Barrozo. **Visão Computacional**. UNICAMP Universidade Estadual de Campinas, FT Faculdade de Tecnologia, 2014.

OLIVEIRA, Cláudio Luis Vieira; NABARRO, Cristina Becker Matos; ZANETTI, Humberto Augusto Piovesana. **Raspberry Pi Descomplicado**. 1. ed. São Paulo: Editora Érica, 2018. 224 p.

OLIVEIRA, Sérgio De. **Internet Das Coisas Com Esp8266, Arduino E Raspberry Pi**. 2. ed. São Paulo: Novatec, 2021.

PAMCARY. **Acidentes com caminhões são quase 5 vezes que furto de cargas**. Pamcary, 2025. Disponível em: <https://www1.gps-pamcary.com.br/acidentes-com-caminhoes-sao-quase-5-vezes-que-furto-de-cargas/>. Acesso em: 20 out. 2025.

RIBEIRO, Joyce. **A cada dois dias, um caminhão entala em viadutos de São Paulo**. R7, 2019. Disponível em: <https://noticias.r7.com/sao-paulo/a-cada-dois-dias-um-caminhao-entala-em-viadutos-de-sao-paulo-30102019/>. Acesso em: 10 set. 2025.

RODRIGUES, Arthur. **Quinze caminhões ficam presos em viadutos todo mês em SP e abalam estruturas**. Folha de S.Paulo, 2019. Disponível em: <https://www.folha.uol.com.br/cotidiano/2019/02/quinze-caminhoes- ficam-presos-em-viadutos-todo-mes-em-sp-e-abalam-estruturas>. Acesso em: 07 abr. 2025.

SANTOS, Sandro. **Introdução à IoT: desvendando a Internet das Coisas**. Santa Catarina: Clube de Autores, 2019.

SILVA, Gabriel de Oliveira; DIAS, Wanderson Roger Azevedo; ESCUDERO, Danilo Pereira. **Análise do Desempenho Computacional da Raspberry Pi Executando o Benchmark SysBench**. Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD). SBC, 2022.

"Os conteúdos expressos no trabalho, assim como os direitos autorais de figuras e dados, bem como sua revisão ortográfica e das normas são de inteira responsabilidade do(s) autor(es)."

"O(s) autor(es) do trabalho declara(m) que durante a preparação do manuscrito foi utilizada a ferramenta ChatGPT (OpenAI) de Inteligência Artificial (IA) exclusivamente para apoio na revisão de linguagem, organização textual e tradução do resumo para inglês e espanhol. Após utilizar esta ferramenta/serviço, os autores editaram e revisaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo da publicação."