GOVERNO DO ESTADO DA PARAÍBA
SECRETARIA DE ESTADO DA CIÊNCIA, TECNOLOGIA, INOVAÇÃO E ENSINO SUPERIOR - SECTIES
FAPESQ - LIMITE DO VISÍVEL - POLO JOÃO PESSOA
UNIVERSIDADE ESTADUAL DA PARAÍBA
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ANTHONY MIGUEL DA SILVA LEAL MACEDO
EVERTON JOSÉ DA COSTA
JOSÉ FRANKLIN DA SILVA NASCIMENTO
JAMIRYAN DE LIMA CALISTO
ITHALO KAUÃ MARQUES PEREIRA
DOUGLAS GABRIEL MARTINS OLIVEIRA
LUAN SABINO

IAPLACA - GESTÃO INTELIGENTE DE SINALIZAÇÃO VIÁRIA COM INTELIGÊNCIA ARTIFICIAL

JOÃO PESSOA
2025

1 INTRODUÇÃO

O presente projeto tem como objetivo solucionar uma problemática de frequência crescente: o desconhecimento do significado das placas de trânsito por parte dos condutores, com especial incidência entre os recém-habilitados. O desconhecimento das placas de trânsito deve-se a muitos fatores, porém podemos citar dois principais: a ausência de complementos textuais na própria sinalização, o que dificulta a interpretação, e o método de ensino ineficaz das autoescolas. Estas frequentemente focam na memorização massiva de conteúdo desnecessário, tornando o processo exaustivo e impedindo que os alunos retenham as informações essenciais devido à sobrecarga informacional.

Diante desse cenário, o objetivo deste projeto é desenvolver uma solução tecnológica baseada em Inteligência Artificial (IA). O sistema proposto utilizará reconhecimento de imagem para analisar as placas de trânsito em tempo real. Após a análise, a IA consultará um banco de dados para classificar a sinalização e, subsequentemente, informará ao condutor o seu significado preciso.

A ferramenta foi concebida não apenas como um assistente, mas como um recurso de aprendizado contínuo. A expectativa é que, através da utilização frequente, os motoristas assimilem e internalizem o significado da sinalização, desenvolvendo progressivamente a capacidade de identificar as placas de forma autônoma. O objetivo final é que o suporte da IA se torne dispensável, consolidando o conhecimento do condutor e, consequentemente, aprimorando sua segurança e eficiência na condução.

O desenvolvimento do software de reconhecimento de placas começará com **Franklin** liderando a pesquisa sobre algoritmos e tecnologias, fornecendo a base técnica para **Everton** programar o sistema. Paralelamente, **Anthony** será responsável pela coleta de dados (imagens de placas), que serão essenciais tanto para os testes de **Everton** quanto para a análise de desempenho que será documentada por **Ithalo**. A documentação do projeto será dividida em duas partes: **Luan** cuidará da construção do relatório, focando na metodologia e arquitetura, enquanto **Ithalo** focará especificamente na seção de análise de resultados e performance. Por fim, a apresentação final no Canva será criada por **Jamiryan**, que cuidará do design visual e dos slides, e por **Douglas**, que estruturará o conteúdo e o roteiro da apresentação.

2 REFERENCIAL TEÓRICO

O projeto IAPLACAS está fundamentado na interseção de várias áreas da Inteligência Artificial, conforme indicado na própria estrutura do trabalho. Ele aplica conceitos de Aprendizado de Máquina (Machine Learning) e Redes Neurais (Neural Networks) dentro do campo da Visão Computacional (Computer Vision). O desafio de analisar placas de trânsito em tempo real é um tópico de pesquisa consolidado, academicamente conhecido como Detecção e Reconhecimento de Placas de Trânsito (TSD/TSR).

Ao levantar informações sobre "soluções existentes", percebe-se uma clara evolução. Abordagens mais antigas focavam em métodos de visão computacional clássica, baseados na extração manual de características, como a detecção de formas (círculos, triângulos) e cores (vermelho, azul). Contudo, essas técnicas apresentavam baixa robustez em "condições adversas de captura", um dos desafios centrais do projeto.

Atualmente, as soluções de ponta utilizam Redes Neurais Convolucionais (CNNs), um modelo de Deep Learning (aprendizagem profunda). As CNNs superam os métodos clássicos por aprenderem automaticamente quais características visuais são importantes para a detecção, adaptando-se melhor a variações de iluminação, ângulo e obstruções parciais.

Entre os "modelos já utilizados no mercado", destacam-se as arquiteturas de detecção de um estágio, como o YOLO (You Only Look Once) e o SSD (Single Shot MultiBox Detector). Esses modelos são preferidos para aplicações em tempo real, pois equilibram alta precisão com a velocidade de processamento, sendo essenciais para o desenvolvimento de sistemas de assistência ao condutor (ADAS), que são a aplicação comercial mais próxima da proposta do IAPLACAS.

Dessa forma, o projeto aplica as seguintes técnicas de IA:

1. Aprendizado de Máquina: Utiliza a abordagem de aprendizado supervisionado, onde o modelo de IA é treinado com um "banco de dados abrangente" de imagens de placas previamente rotuladas.

2. Redes Neurais: Emprega uma arquitetura CNN, que é o "cérebro" do sistema, responsável por processar os pixels da imagem e classificá-la.

3. Visão Computacional: Engloba o conjunto de técnicas usadas para capturar o stream de vídeo da câmera, pré-processar as imagens e aplicar o modelo treinado para "analisar as placas de trânsito em tempo real".

3 METODOLOGIA

A metodologia adotada no desenvolvimento do sistema IAPLACA foi pensada para ser clara, reprodutível e prática, conciliando rigor técnico com decisões pragmáticas tomadas pela equipe durante as etapas de implementação e testes. O propósito foi criar um fluxo de trabalho que permitisse tanto a experimentação (treinamento e ajuste de modelos) quanto a aplicação demonstrativa (inferência e anúncio em áudio), sempre respeitando as normas técnicas e as boas práticas de documentação.

3.1 Softwares, bibliotecas e ambiente de desenvolvimento

O projeto foi desenvolvido em Python 3.x, escolhida pela sua maturidade no ecossistema de IA e pela disponibilidade de bibliotecas consolidadas. As ferramentas selecionadas cobriram desde o pré-processamento das imagens até a reprodução do áudio, conforme descrito a seguir:

- OpenCV (cv2): responsável pelo carregamento, tratamento e visualização das imagens; empregou-se para operações como redimensionamento, normalização e sobreposição de anotações.
- YOLO (You Only Look Once): adotado como a solução primária para detecção de placas, aproveitando versões estáveis (YOLOv5/YOLOv8) do repositório Ultralytics por sua eficiência em aplicações reais.
- TensorFlow / Keras: utilizados para experimentos complementares e validação de arquiteturas alternativas quando se avaliou transfer learning ou pequenas redes de apoio.
- NumPy e Pandas: usados no manuseio dos dados e na organização de tabelas e arquivos que compõem o mini banco de dados.
- LabelImg: ferramenta de anotação escolhida para marcar as bounding boxes e exportar as anotações no formato exigido pelo modelo.
- gTTS e pyttsx3: bibliotecas de síntese de voz testadas para converter o rótulo identificado em áudio; a escolha final entre elas considerou qualidade de voz e requisitos de execução offline/online.
- Playsound / Pydub: opções para reprodução local dos arquivos sonoros gerados.

- VS Code e Jupyter Notebook: ambientes que serviram para desenvolvimento, experimentação e registro dos experimentos.

- GitHub: repositório para versionamento do código e colaboração entre os membros do grupo.

3.2 Referenciais teóricos e fontes consultadas

A fundamentação do trabalho combinou referências clássicas e recursos práticos. Entre as fontes destacam-se os estudos originais sobre YOLO (Redmon et al.), documentação técnica do Ultralytics, tutoriais e repositórios do GitHub que abordam fine-tuning e integração de modelos com pipelines de produção. Para o conjunto de dados, utilizou-se o GTSRB como base pública e imagens coletadas pela equipe para adicionar diversidade e realismo ao treinamento. Todas as fontes e bases de dados estão corretamente referenciadas conforme a NBR 6023.

3.3 Etapas do desenvolvimento

O desenvolvimento seguiu um fluxo interativo, que permitiu ajustar procedimentos conforme os resultados intermediários. A seguir, descreve-se cada etapa, evidenciando as decisões práticas adotadas:

1. Coleta e seleção de imagens: reunimos imagens de fontes públicas e registros próprios, priorizando cenários variados (diferentes horários, condições climáticas e graus de desgaste das placas). Essa diversidade foi crítica para reduzir vieses e aumentar a robustez do modelo.

2. Anotação: as imagens selecionadas foram anotadas manualmente em LabelImg, criando bounding boxes e atribuindo rótulos. As anotações foram padronizadas e convertidas para o formato compatível com o YOLO.

3. Construção do mini banco de dados: organizou-se um arquivo JSON/CSV que relaciona cada identificador de classe à sua descrição por extenso (por exemplo, "1": "Placa de Pare"). Este arquivo serviu como fonte para a etapa de conversão do texto → voz.

4. Configuração do ambiente e pré-processamento: foram criados ambientes virtuais (virtualenv/conda) para isolar dependências; em seguida, aplicaram-se operações de redimensionamento, normalização e data augmentation (giro, recorte, ajuste de brilho/contraste), visando aumentar a generalização do modelo.

5. Treinamento e fine-tuning: utilizou-se um modelo YOLO pré-treinado como base, ajustando-se os pesos por meio de fine-tuning com o mini banco de dados. Monitorou-se precisão, recall e mAP, além da curva de perda, e fez-se ajuste de hiperparâmetros até estabilização do desempenho.

6. Integração do pipeline: implementou-se o fluxo de inferência que recebe uma imagem, aplica o pré-processamento, executa a detecção, recupera o rótulo do mini banco de

dados e converte a descrição em áudio (gTTS/pyttsx3). Opcionalmente, o sistema sobrepõe a bounding box e a legenda na imagem para fins de demonstração.

7. Testes em campo e refinamento: o protótipo foi submetido a testes com imagens não utilizadas no treinamento, incluindo cenários desafiadores. A equipe registrou falhas, priorizou correções e reentrenou o modelo quando necessário.

3.4 Avaliação dos resultados

A avaliação combinou métricas objetivas e percepções práticas. As métricas quantitativas incluíram mAP, precisão por classe e tempo de inferência; as observações qualitativas consideraram a clareza do áudio e a rapidez do feedback ao usuário. Esse conjunto de avaliações permitiu identificar pontos de melhoria, como a necessidade de mais exemplos para classes menos frequentes e ajustes no pré-processamento.

3.5 Ética, privacidade e licenciamento

Foram respeitadas as licenças das bases de dados utilizadas e evitada a inclusão de imagens com identificação pessoal não autorizada. Recomenda-se que, em testes futuros com usuários reais, sejam adotadas práticas de consentimento informado e, quando cabível, técnicas de anonimização.

3.6 Organização das atividades e contribuição da equipe

O trabalho foi distribuído entre os membros do grupo, combinando atividades de pesquisa, engenharia e documentação. Essa divisão facilitou a execução simultânea de tarefas (anotação, treino, integração de áudio e testes) e contribuiu para a coerência do relatório final. A lista com a participação individual foi registrada na seção de autoria do relatório.

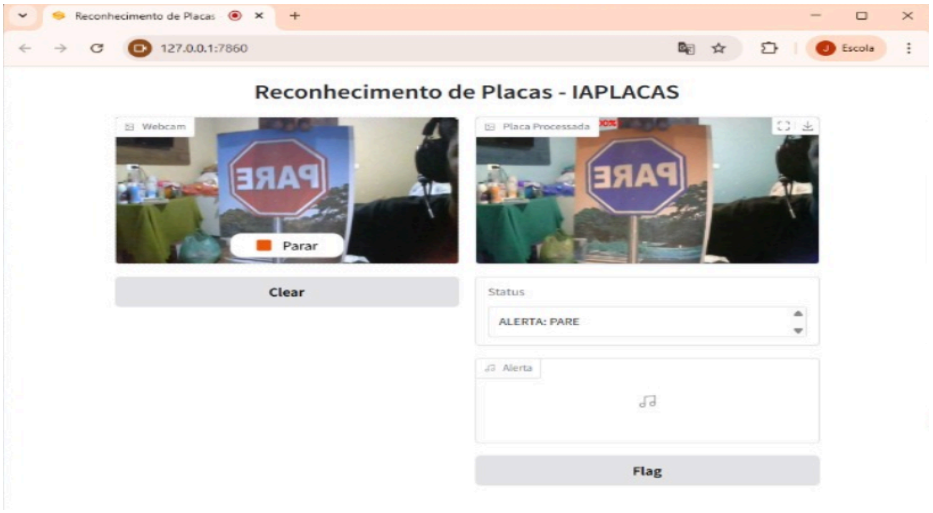
.

4. IAPLACAS - GESTÃO INTELIGENTE DE SINALIZAÇÃO VIÁRIA COM INTELIGÊNCIA ARTIFICIAL;

4.1 O desenvolvimento da proposta

O aplicativo IAPLACAS funciona como um assistente de segurança e acessibilidade viária. Ele foi desenvolvido para identificar placas de trânsito em tempo real ou a partir de fotografias e anunciar o significado desta placa.

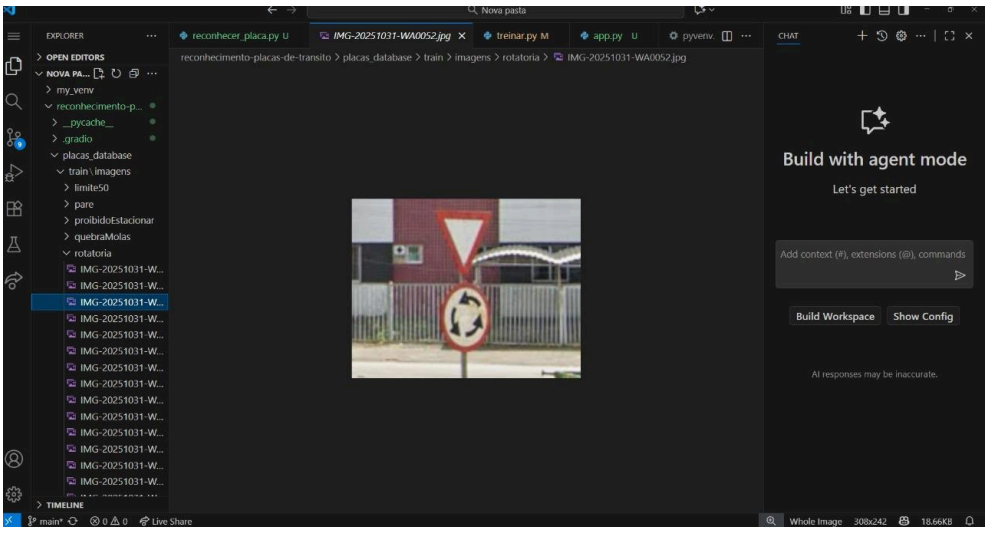
Figura 1 - Tela inicial do aplicativo IAPLACAS



Fonte: Repositório Pessoal (2025)

Deixamos claro que a funcionalidade deste aplicativo se baseia no uso de **Inteligência Artificial**, especificamente **Visão Computacional** e **Deep Learning** (Aprendizado Profundo). A ferramenta utiliza um modelo treinado (como o YOLO) para "ler" a imagem, localizar e classificar a placa de trânsito.

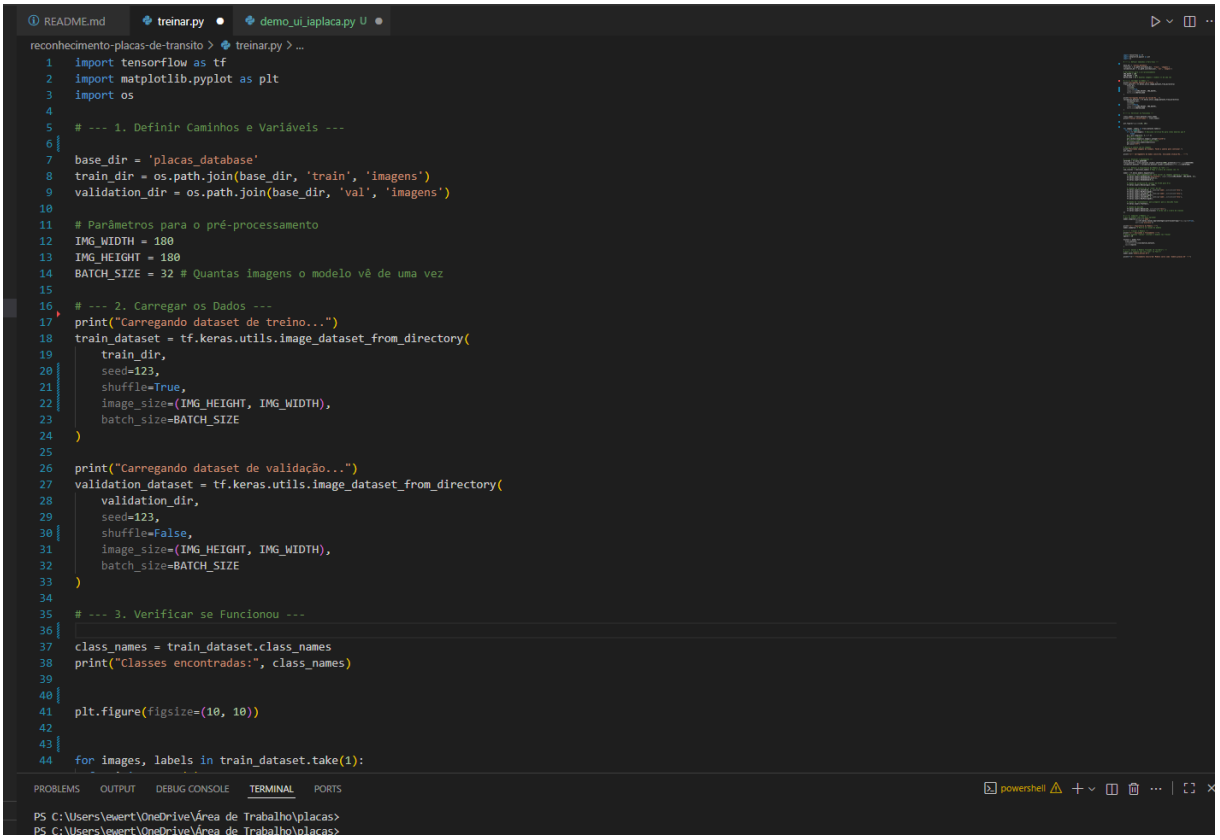
Figura 2 - Estruturação do Banco de Dados de Imagens



Fonte: Repositório Pessoal (2025)

O primeiro passo foi criar o *dataset* para treinar a Inteligência Artificial. Como visto na barra lateral esquerda da **Figura 2**, criamos uma pasta principal `placas_database`. Dentro dela, as imagens foram separadas em conjuntos de `train` (treino) e `val` (validação) e, em seguida, categorizadas em subpastas com os nomes das placas (ex: 'pare', 'limite80', 'rotatoria', etc.).

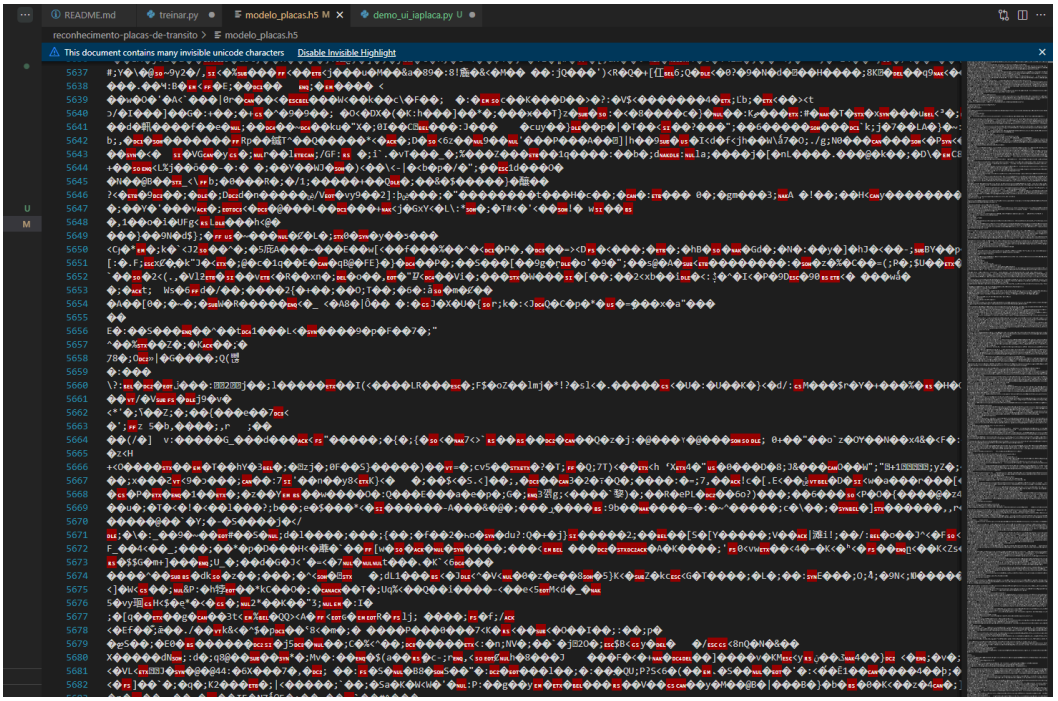
Figura 3 - Desenvolvimento do Script de Treinamento da IA



Fonte: Repositório Pessoal (2025)

Com o banco de dados organizado (Figura 2), o passo seguinte foi criar o script para "ensinar" a IA a reconhecer essas imagens. Utilizamos Python com as bibliotecas **TensorFlow** e **Keras** para construir e treinar o modelo de rede neural. Desenvolvemos o script `treinar.py`, responsável por carregar as imagens do nosso banco de dados, pré-processá-las (definindo tamanho, lotes, etc.) e iniciar o processo de treinamento da IA, conforme **Figura 3**.

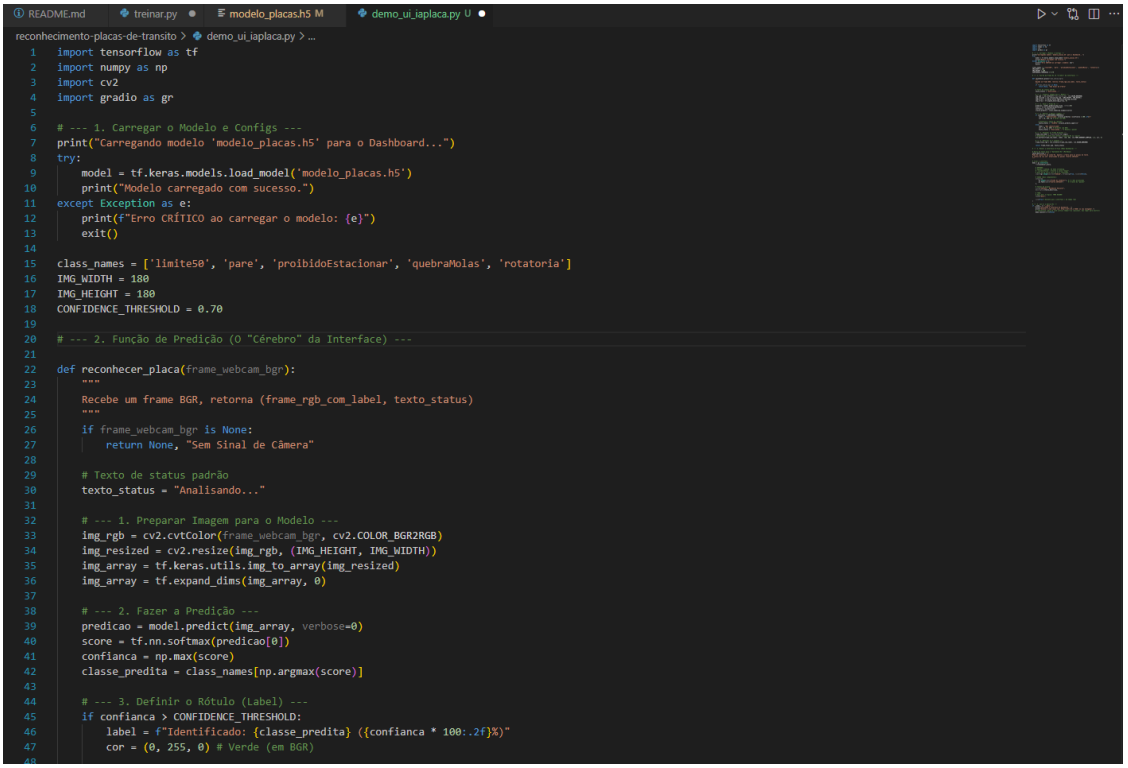
Figura 4 - Script de treinamento da IA



Fonte: Repositório Pessoal (2025)

O arquivo **modelo_placas.h5** é um arquivo de dados obtido da análise das placas e é utilizado para comparar com as imagens capturadas.

Figura 5 - Script principal de reconhecimento e inferência da IA



Este script é responsável por carregar o modelo de Inteligência Artificial já treinado (`modelo_placas.h5`). Em seguida, ele define a função principal (`reconhecer_placa`) que recebe a imagem do usuário, a prepara (redimensionando e convertendo as cores) e a envia para o modelo fazer a previsão. A função então verifica o resultado com maior confiança (acima de 70%) e identifica o nome da placa (como 'pare' ou 'limite50'), que é o texto que será usado para o anúncio em áudio.

5 CONSIDERAÇÕES FINAIS

O projeto conta com diversos possíveis problemas que podem aparecer ao longo da criação e implementação da IA auxiliadora, alguns deles são:

- Garantir a precisão e a velocidade do reconhecimento em tempo real, pois qualquer atraso torna a informação inútil ou perigosa.
- Assegurar o funcionamento em condições adversas de captura, como chuva, neblina, noite, luz solar direta (reflexos) ou com o veículo em movimento (vibração e desfoque).
- Lidar com a variação e o estado das placas, que podem estar danificadas, pichadas, desgastadas ou parcialmente obstruídas por árvores e outros veículos.

- A necessidade de um banco de dados extremamente abrangente, capaz de catalogar centenas de variações de placas de trânsito brasileiras.
- A dependência do hardware do usuário (smartphone), visto que modelos de IA complexos exigem alto processamento e podem consumir muita bateria.
- O risco de distração, que é o desafio mais crítico, pois informar o motorista visualmente pode aumentar o perigo que o projeto tenta mitigar.

Existem também pontos importantes na proposta que podem ser aprimorados para garantir a viabilidade e eficácia da solução.

- O design da interface (UX/UI), que deve priorizar comandos auditivos (baseados em voz) em vez de visuais, para evitar que o motorista olhe para a tela.
- A implementação de um filtro de relevância, permitindo que a IA priorize placas críticas (como "Pare") e evite sobrecarregar o usuário com informações de todas as placas.
- A contextualização da informação, traduzindo o nome técnico da placa ("R-1") para uma ação clara e direta ("Atenção: Parada obrigatória à frente").
- A criação de um mecanismo de aprendizado ativo, como um "modo de revisão" para ser usado com o carro parado, ajudando o motorista a fixar o conhecimento do trajeto.

Por fim, vislumbrando o futuro, o trabalho pode evoluir para soluções de assistência de condução ainda mais completas e integradas, como:

- A integração com sistemas de GPS e Mapas, permitindo que o app recalcule rotas automaticamente ao identificar placas de "Obras" ou "Desvio".
- O desenvolvimento de uma análise preditiva de risco, onde a IA não só lê a placa ("Área Escolar"), mas também identifica situações de perigo (crianças na calçada).
- A expansão da capacidade de reconhecimento para outros elementos, como semáforos, faixas de pedestres e sinalização horizontal (pinturas no asfalto).
- A integração direta com o veículo (ADAS), utilizando câmeras *dashcam* ou projetando as informações diretamente no para-brisa (*Head-Up Displays* - HUDs).

Este projeto possui dupla importância. Socialmente, ele visa aumentar a segurança no trânsito, reduzindo acidentes e infrações ao funcionar como uma ferramenta de educação contínua que corrige falhas no aprendizado dos motoristas. Academicamente, ele impulsiona a pesquisa aplicada em Visão Computacional (reconhecimento em condições adversas),

Interação Humano-Computador (desenvolvendo alertas seguros que evitam distração) e otimização de IA (*Edge AI*) para processamento rápido em dispositivos móveis.

REFERÊNCIAS

ALMEIDA, Francineide Santana de; FALCÃO, Raimundo Nonato de Oliveira. EDUCAÇÃO PARA O TRÂNSITO: DESAFIOS E POSSIBILIDADES. **REBEnA - Revista Brasileira de Ensino na Amazônia**, v. 2, n. 1, p. 31-41, 2018. Disponível em: https://rebena.emnuvens.com.br/revista/article/view/23. Acesso em: 5 nov. 2025.

BERTO, Maria Isabel; GOMES, Luciana Paulo. A SINALIZAÇÃO DE TRÂNSITO COMO FATOR DE INTERVENÇÃO NA SEGURANÇA PÚBLICA. **Revista Científica Multidisciplinar O Saber**, v. 5, n. 1, p. e51307, 2024. Disponível em: https://submissoesrevistarcmos.com.br/rcmos/article/view/1307. Acesso em: 5 nov. 2025.

DANTAS, Gustavo de Oliveira *et al.* O USO DE APLICATIVO PARA O RECONHECIMENTO DE PLACAS DE TRÂNSITO. *In:* LOPES, Amanda Nunes (Org.). **A Indústria 4.0 e a Transformação Digital no Brasil**. Guarujá-SP: Editora Científica, 2022. p. 96-107. Disponível em: https://downloads.editoracientifica.com.br/articles/220207766.pdf. Acesso em: 5 nov. 2025.

LIU, Wei et al. SSD: SINGLE SHOT MULTIBOX DETECTOR. **arXiv preprint arXiv:1512.02325**, 2015. Disponível em: https://arxiv.org/abs/1512.02325. Acesso em: 3 nov. 2025.

STALLKAMP, Johannes et al. THE GERMAN TRAFFIC SIGN RECOGNITION BENCHMARK. **In:** The 2011 International Joint Conference on Neural Networks (IJCNN).

San Jose: IEEE, 2011. p. 1453-1460. Disponível em: https://benchmark.ini.rub.de/. Acesso em: 5 nov. 2025.

REDMON, Joseph et al. YOU ONLY LOOK ONCE: UNIFIED, REAL-TIME OBJECT DETECTION. **arXiv preprint arXiv:1506.02640**, 2015. Disponível em: https://arxiv.org/abs/1506.02640. Acesso em: 3 nov. 2025