



REDE NEURAL PARA DETECÇÃO DE PNEUS COM YOLOV8: UMA FERRAMENTA PARA O COMBATE À DENGUE

NEURAL NETWORK FOR TIRE DETECTION USING YOLOV8: A TOOL FOR DENGUE CONTROL

RED NEURONAL PARA DETECCIÓN DE NEUMÁTICOS CON YOLOV8: UNA HERRAMIENTA PARA EL COMBATE AL DENGUE

Andrey Alencar Quadros¹
Murilo Fuza da Cunha²

DOI: 10.54751/revistafoco.v18n4-021

Received: Mar 3rd, 2025

Accepted: Mar 25th, 2025



RESUMO

O aumento da incidência de dengue no Brasil, especialmente em áreas urbanas, representa um desafio significativo para a saúde pública. Este artigo apresenta uma solução inovadora que utiliza drones e redes neurais profundas para a detecção de pneus, que são potenciais criadouros do mosquito *Aedes aegypti*. O principal objetivo da pesquisa é desenvolver um sistema eficiente para identificar e mapear esses pneus em áreas endêmicas, empregando a arquitetura YOLOv8 para a detecção de objetos. A metodologia adotada envolve uma abordagem aplicada, utilizando técnicas de aprendizado de máquina e processamento de imagens, com a implementação do sistema em uma infraestrutura baseada em *containers Docker*, o que garante portabilidade e eficiência. A coleta de dados foi realizada em locais estratégicos, utilizando drones equipados com câmeras de alta resolução. Os resultados obtidos demonstraram a eficácia da API e do sistema de detecção, que se mostraram robustos e escaláveis para integração em sistemas de vigilância. A utilização de redes neurais profundas, como YOLO, possibilitou a detecção precisa de pneus em imagens, contribuindo para um mapeamento mais eficiente dos focos de dengue. O estudo conclui que a integração de tecnologias de inteligência artificial e drones pode revolucionar o combate à dengue, permitindo uma resposta mais rápida e eficaz a surtos da doença, além de sugerir direções futuras para a pesquisa, incluindo a ampliação do *dataset* e a melhoria contínua dos algoritmos de detecção.

Palavras-chave: Dengue; drones; inteligência artificial; detecção automatizada de pneus.

¹ Mestre Em Transferência De Tecnologia Para Inovação. Instituto Federal De Rondônia. Rodovia, RO-257, s/n - Zona Rural, Ariquemes - RO, CEP: 76870-000. E-mail: andrey.quadros@ifro.edu.br

² Graduado em Ciência da Computação. Universidade Federal de Rondônia (UNIR). Av. Pres. Dutra, 2965 - Olaria, Porto Velho - RO, CEP: 76801-058. E-mail: contato@murilofuza.com.br

ABSTRACT

The increasing incidence of dengue in Brazil, particularly in urban areas, presents a significant challenge for public health. This article presents an innovative solution that uses drones and deep neural networks for tire detection, as tires are potential breeding grounds for the *Aedes aegypti* mosquito. The primary goal of the research is to develop an efficient system to identify and map these tires in endemic areas, employing the YOLOv8 architecture for object detection. The methodology adopted involves an applied approach, utilizing machine learning techniques and image processing, with the implementation of the system on a Docker container-based infrastructure, ensuring portability and efficiency. Data collection was carried out in strategic locations using drones equipped with high-resolution cameras. The results obtained demonstrated the effectiveness of the API and the detection system, which proved to be robust and scalable for integration into surveillance systems. The use of deep neural networks, such as YOLO, enabled accurate tire detection in images, contributing to a more efficient mapping of dengue hotspots. The study concludes that the integration of artificial intelligence technologies and drones can revolutionize the fight against dengue, enabling a faster and more effective response to disease outbreaks, as well as suggesting future directions for research, including the expansion of the dataset and the continuous improvement of detection algorithms.

Keywords: Dengue; drones; artificial intelligence; automated tire detection.

RESUMEN

El aumento de la incidencia de dengue en Brasil, especialmente en áreas urbanas, representa un desafío significativo para la salud pública. Este artículo presenta una solución innovadora que utiliza drones y redes neuronales profundas para la detección de neumáticos, que son criaderos potenciales del mosquito *Aedes aegypti*. El objetivo principal de la investigación es desarrollar un sistema eficiente para identificar y mapear estos neumáticos en áreas endémicas, empleando la arquitectura YOLOv8 para la detección de objetos. La metodología adoptada implica un enfoque aplicado, utilizando técnicas de aprendizaje automático y procesamiento de imágenes, con la implementación del sistema en una infraestructura basada en contenedores Docker, lo que garantiza portabilidad y eficiencia. La recolección de datos se llevó a cabo en lugares estratégicos, utilizando drones equipados con cámaras de alta resolución. Los resultados obtenidos demostraron la efectividad de la API y del sistema de detección, que se mostraron robustos y escalables para la integración en sistemas de vigilancia. El uso de redes neuronales profundas, como YOLO, permitió la detección precisa de neumáticos en imágenes, contribuyendo a un mapeo más eficiente de los focos de dengue. El estudio concluye que la integración de tecnologías de inteligencia artificial y drones puede revolucionar la lucha contra el dengue, permitiendo una respuesta más rápida y eficaz ante brotes de la enfermedad, además de sugerir direcciones futuras para la investigación, incluyendo la ampliación del conjunto de datos y la mejora continua de los algoritmos de detección.

Palabras clave: Fiebre del dengue; drones; inteligencia artificial; detección automática de neumática.

1. Introdução

A Dengue é definida como uma febre aguda e não contagiosa, sendo uma das arboviroses mais importantes que afetam os humanos. Isso se deve à sua alta incidência e potencial de disseminação (Andrioli; Busato; Lutinski, 2020).

No Brasil, a maior epidemia foi registrada em 2015 (1.688.688 casos), com uma incidência de 826,0 casos por 100.000 habitantes, o que ilustra a ocorrência da dengue na última década com picos epidêmicos cada vez maiores e encurtamento dos períodos interepidêmicos (Andrioli; Busato; Lutinski, 2020). Sua incidência vem aumentando nas últimas décadas, expandindo-se geograficamente para regiões subtropicais.

Um fator crítico para esta disseminação está associado à proximidade de pontos estratégicos como ferros-velhos, borracharias e materiais recicláveis no Brasil (Johansen *et al.*, 2018). Visando tal situação, é possível precaver que locais como esses se tornem focos de dengue e permitam que uma doença possivelmente letal se manifeste desde pequenas cidades até grandes metrópoles, causando surtos e até casos de óbito.

Para enfrentar este desafio de saúde pública, nos últimos anos, o uso de tecnologias de inteligência artificial, como as redes neurais artificiais, tem-se mostrado uma ferramenta promissora na identificação e mitigação de fatores de risco em saúde pública. Essas tecnologias podem ser aplicadas para monitorar áreas de risco, prever surtos e direcionar ações de combate de forma eficiente (Guilherme *et al.*, 2022).

Em Rondônia, por exemplo, são publicados boletins epidemiológicos pelo governo do estado, disponíveis em <https://rondonia.ro.gov.br/agevisa/institucional/boletim-epidemiologico/> onde é possível analisar os casos de dengue e verificar épocas em que a doença tende a apresentar picos de casos e proliferação.

Neste contexto, o presente trabalho explora a aplicação de redes neurais para a detecção de pneus, que frequentemente servem como criadouros do mosquito *Aedes aegypti*, vetor da dengue. Com a detecção automatizada de possíveis focos, é possível otimizar o trabalho de vigilância sanitária, reduzindo o

tempo e os custos envolvidos, além de contribuir diretamente para o controle da disseminação da doença.

2. Revisão de Literatura

Técnicas de IA são amplamente aplicadas à detecção de surtos de doenças infecciosas, alerta precoce, previsão de tendências e modelagem e avaliação de resposta de saúde pública. (Zeng; Cao; Neill, 2020).

Essas aplicações aproveitam o poder de modelos preditivos para identificar padrões em grandes volumes de dados de saúde, sociais e ambientais, permitindo intervenções mais rápidas e eficazes.

Uma área de destaque é o uso de redes neurais profundas, como as arquiteturas YOLO (*You Only Look Once*), para detecção e classificação de objetos em imagens e vídeos. Essas redes têm mostrado eficácia em tarefas que envolvem análise de imagens de alta resolução, sendo utilizadas em aplicações que vão desde a identificação de condições médicas em exames de imagem até o monitoramento ambiental (Redmon *et al.*, 2016).

Sistemas de vigilância baseados em IA demonstraram capacidade significativa na detecção precoce de surtos de doenças infecciosas, como evidenciado durante a pandemia de COVID-19 (Bullock *et al.*, 2020). A integração de algoritmos de aprendizado de máquina com dados de vigilância sindrômica tem permitido a identificação mais rápida de anomalias e padrões emergentes em dados de saúde pública (Chimmula; Zhang, 2020).

2.1 Tecnologias no Combate à Dengue

Os drones podem ser uma ferramenta valiosa na vigilância e controle do *Aedes aegypti*, especialmente em áreas de difícil acesso, como telhados e quintais. (Valdez-delgado *et al.*, 2021)

Além desta capacidade de identificação, os drones equipados com câmeras de alta resolução podem cobrir grandes áreas em menos tempo quando comparados à inspeção manual tradicional. Esta tecnologia permite o acesso a

locais de difícil alcance, como telhados altos, terrenos acidentados e áreas remotas, possibilitando uma vigilância mais abrangente e eficiente.

A integração destes equipamentos com sistemas de geolocalização também facilita o mapeamento preciso dos pontos críticos, permitindo um planejamento mais estratégico das ações de controle e prevenção da dengue.

O uso de tecnologias geoespaciais, como Sistema de Informação Geográfica (SIG) e GPS, em conjunto com drones, tem demonstrado grande eficácia na identificação e monitoramento de áreas de risco (Akter *et al.*, 2019)

Estudos recentes demonstram que a combinação de imagens aéreas de alta resolução com dados de localização permite criar mapas detalhados de risco, identificando padrões espaciais de distribuição do vetor e orientando intervenções mais direcionadas e eficientes (Valdez-delgado *et al.*, 2023). Esta abordagem tecnológica não apenas otimiza recursos, mas também permite um monitoramento contínuo das áreas endêmicas, facilitando a detecção precoce de potenciais surtos.

2.2 Yolov8 e Detecção de Objetos

O YOLOv8 representa a mais recente evolução da arquitetura YOLO (*You Only Look Once*), desenvolvida pela *Ultralytics*. Este modelo mantém o princípio fundamental de detecção de objetos em uma única passagem, característica que o tornou conhecido por sua velocidade e eficiência (Jocher; Chaurasia; Qiu, 2023).

Uma das inovações significativas do YOLOv8 é a adoção de uma abordagem sem âncoras (*anchor-free*), que simplifica o processo de detecção e melhora a precisão em objetos de diferentes escalas. Além disso, a integração de uma nova espinha dorsal (*backbone*) aprimora a extração de características, permitindo que o modelo capture detalhes mais refinados das imagens. Essas melhorias resultam em um equilíbrio notável entre velocidade e precisão, posicionando o YOLOv8 como uma solução de ponta para aplicações em tempo real. (Terven; Córdova-esparza; Romero-gonzález, 2023)

O desempenho do YOLOv8 foi avaliado em *benchmarks* renomados,

como o *Microsoft COCO*, onde demonstrou alta precisão e capacidade de operação em tempo real em diversas plataformas de hardware. Além disso, o modelo apresenta melhorias na facilidade de uso para desenvolvedores, incluindo um pacote *Python* unificado e uma interface de linha de comando (CLI), que simplificam o treinamento e a implantação do modelo. Essas características reforçam a posição do YOLOv8 como uma ferramenta avançada e acessível para detecção de objetos em aplicações diversas. (Yaseen, 2024)

3. Materiais e Métodos

O método a ser utilizado neste trabalho é o da pesquisa aplicada, baseada em hipótese-dedução, utilizando referências científicas e tecnológicas para definição do problema, especificação da hipótese de solução e sua avaliação (Wazlawick, 2008). Para a análise do problema, foram realizados testes em máquinas pessoais e máquinas de alta performance para processamento e treinamento com *datasets* de imagens.

Técnicas e metodologias modernas de infraestrutura de aplicativos incentivam uma adoção acelerada de tecnologias de computação em nuvem, bem como várias tecnologias de virtualização (Scheepers, 2014). Além disso, acredita-se que o uso de técnicas de containerização e virtualização possa simplificar tanto o desenvolvimento quanto a implantação da solução proposta.

Dentre as métricas disponíveis para este estudo, considera-se que é imprescindível a avaliação da viabilidade de implementações com redes neurais para combater os problemas de propagação causados por mosquitos em determinadas épocas do ano. Visando entregar bons resultados, foi construído um plano de trabalho para garantir um bom desenvolvimento e apresentação de resultados.

O projeto foi constituído do planejamento das etapas de todo o processo de montagem de *datasets* até o desenvolvimento de ferramentas *FrontEnd* para visualização e utilização dos dados gerados através do software resultante desta pesquisa. Abordagens de desenvolvimento ágil se baseiam em uma abordagem incremental para a especificação, desenvolvimento e entrega do software,

permitindo que a equipe foque mais no software em si, e não necessariamente em sua concepção e documentação (Sommerville, 2011).

Visando o melhor aproveitamento das etapas de elaboração das atividades, sendo a equipe composta por apenas uma pessoa responsável por todo o processo, a divisão se deu em:

- A) Construção de *dataset*;
- B) Alimentação e treinamento de rede neural;
- C) Construção de API para consumo de dados e alimentação da rede;
- D) Construção de interface gráfica WEB para consumo das informações geradas e visualização em mapa.

Foi iniciado o estudo de viabilidade para a captura das imagens e identificação dos locais potenciais de focos. Com base nesses estudos, foram definidos os pontos iniciais do *dataset*, os quais serviram para criar versões base e gerar métricas de treinamento. A partir dessa análise, foi possível avaliar se seria necessário ampliar o número de imagens, considerando diferentes condições e cenários de captura, ou se as imagens iniciais seriam suficientes para o treinamento do modelo.

3.1 Equipamentos e Ferramentas

O desenvolvimento deste projeto envolveu o uso de diversos equipamentos e ferramentas, tanto de hardware quanto de software, para garantir a eficácia na detecção de pneus e no processamento das imagens capturadas pelos drones.

3.1.1 Hardware Utilizado

Para o processamento e treinamento da rede neural, foram utilizados dois computadores principais:

- A) **Computador de Desempenho Avançado:**
 - a) **Memória RAM:** 64 GB DDR4.
 - b) **Armazenamento:** 1 TB NVMe, garantindo alta velocidade de leitura e gravação de dados.

c) **Placa de Vídeo:** RTX 4090 com 24 GB de memória dedicada, proporcionando o poder de processamento necessário para treinar a rede neural em um tempo reduzido.

d) **Conexão de Rede:** 1 Gbps, assegurando transferências de dados rápidas e eficientes.

B) Notebook Avell A65ion:

a) **Memória RAM:** 32 GB.

b) **Processador:** Intel Core i7 de 12ª geração, com alto desempenho para suportar as demandas do treinamento e processamento das imagens.

c) **Placa de Vídeo:** RTX 4050 com 6 GB de memória, suficiente para a execução de modelos em tempo real e pré-processamento de dados.

d) **Conexão de Rede:** 1 Gbps, garantindo agilidade na comunicação com servidores e repositórios de dados.

Para a captura das imagens, foi utilizado o drone **DJI Mavic 2**, um modelo avançado que permite a gravação de vídeos e fotos aéreas em alta definição. O drone é controlado por um dispositivo móvel, no caso, um **Redmi Note 13 Pro**, que serve para operar o drone e visualizar as imagens em tempo real. As imagens capturadas pelo drone são armazenadas em um **cartão de memória de 128 GB**, garantindo amplo espaço para o armazenamento de dados durante as missões de captura.

3.1.2 Software e Bibliotecas

O software e as bibliotecas utilizadas no projeto foram essenciais para a construção do modelo de detecção e a interface de visualização dos resultados.

A) **RoboFlow** foi a plataforma escolhida para a anotação e pré-processamento das imagens, oferecendo ferramentas robustas para facilitar a organização e a preparação do dataset.

As bibliotecas e *frameworks* utilizados para o desenvolvimento do sistema de detecção incluem:

A) **Next.js** – *Framework* utilizado para o desenvolvimento da aplicação web que processa e exibe os resultados da detecção em tempo real.

- B) **Mantine** – Biblioteca de componentes *React* que foi usada para criar uma interface de usuário moderna e responsiva.
- C) **FastAPI** – *Framework* rápido e eficiente para a construção de APIs que facilitam a comunicação entre o *front-end* e o *back-end* da aplicação.
- D) **Ultralytics** – Biblioteca que fornece implementações otimizadas do modelo YOLOv8, permitindo o treinamento e a implementação de redes neurais para detecção de objetos.

Esses equipamentos e ferramentas formaram a base tecnológica para o desenvolvimento do projeto, proporcionando tanto a capacidade de treinamento e execução da rede neural quanto a criação de uma interface funcional para os usuários finais.

3.2 Metodologia de Coleta de Dados

A coleta de dados para o projeto se deu na cidade de Ariquemes - Rondônia, localizada a 203 quilômetros da capital Porto Velho, onde em dois locais, sendo eles a pista de kart municipal no centro da cidade e o pátio da GCM (Guarda Civil Municipal) que foi fornecido por parceria entre o projeto Cidades Inteligentes e a GCM.

As imagens foram capturadas através do drone disponibilizado pelo projeto em resolução 4k (3.840 *pixels* horizontais e 2.160 *pixels* verticais), com variações de 4 a 10 metros de altura, majoritariamente na parte inicial da manhã e final de tarde, variando de horário como 08 AM até 10 AM e no período da tarde entre 15 PM e 18 PM.

A coleta manteve um padrão de iluminação com alto índice de luz solar e sempre com o drone e sua câmera em posição contrária à luz do sol.

Figura 1. Imagem captura no período da manhã - ambiente simulado



Fonte: Captura do Autor, 2024

Figura 2. Imagem captura no período da tarde - ambiente simulado



Fonte: Captura do Autor, 2024

3.3 Treinamento do Modelo

Para a base de dados foi utilizado um *dataset* de 7100 imagens, onde todas

foram capturadas com o drone, de modo a padronizar as entradas. As imagens possuem processamentos para aumentar a base tendo um número restrito de imagens, sendo esses processamentos feitos através da plataforma Roboflow.

- A) *Virar: Horizontal, Vertical;*
- B) *Rotação: Entre -15° e +15°;*
- C) *Cisalhamento: ±10° Horizontal, ±10° Vertical;*
- D) *Escala de Cinza: Aplicar em 15% das imagens;*
- E) *Matiz: Entre -15° e +15°;*
- F) *Saturação: Entre -25% e +25%;*
- G) *Brilho: Entre -15% e +15%;*
- H) *Exposição: Entre -10% e +10%;*
- I) *Desfoque: Até 2.5px.*

3.4 Treinamento Local

Durante o treinamento da rede neural YOLOv8, utilizou-se um conjunto de aproximadamente 7.100 imagens, divididas em 6.400 para treinamento, 614 para validação e 300 para teste. A configuração do treinamento foi definida com os seguintes parâmetros:

Tabela 1. Configuração do treinamento

```
!yolo task=detect mode=train model=yolo8n.pt data='./data.yaml' \  
  dropout=0.4 \  
  cos_lr=True \  
  epochs=300 \  
  batch=16 \  
  imgsz=640 \  
  device=0 \  
  mosaic=1.0 \  
  mixup=0.3 \  
  augment=True
```

Fonte: Autores

A) *task=detect* : Este parâmetro especifica que a tarefa a ser realizada é

- de **detecção de objetos**, ou seja, o modelo irá aprender a identificar e localizar objetos nas imagens (neste caso, pneus).
- B) `mode=train`: Define que o modo de operação será de **treinamento**, permitindo que o modelo seja ajustado a partir dos dados fornecidos.
- C) `model=yolo8n.pt`: Indica o modelo pré-treinado a ser utilizado como ponto de partida. Neste caso, o modelo YOLOv8n (uma versão mais leve e otimizada do YOLOv8) será ajustado com os dados fornecidos.
- D) `data='./data.yaml'`: Especifica o caminho do arquivo `data.yaml`, que contém as configurações do conjunto de dados, como o número de classes e os caminhos para as imagens de treino, validação e teste.
- E) `dropout =0.4`: Aplica a técnica de **dropout**, que consiste em desligar aleatoriamente uma fração das unidades de uma camada durante o treinamento para evitar *overfitting*. O valor 0.4 indica que 40% das unidades serão desativadas durante o treinamento.
- F) `cos_lr=True` : Ativa o **cosine learning rate** (taxa de aprendizado cosseno). Esta técnica ajusta a taxa de aprendizado de maneira dinâmica, começando com um valor alto e diminuindo progressivamente com uma função cosseno, melhorando a convergência do modelo.
- G) `epochs =300`: Define o número de **épocas** (iterações sobre o conjunto completo de dados) que o treinamento irá realizar. Neste caso, o treinamento será realizado por 300 épocas.
- H) `batch =16`: Define o tamanho do **batch** (tamanho do lote) como 16. Isso significa que o modelo processará 16 imagens de uma vez antes de atualizar os pesos.
- I) `imgsz=640`: Define o **tamanho das imagens** de entrada para 640x640 *pixels*. Esta é a resolução das imagens que serão usadas durante o treinamento e inferência. Reduzir a resolução pode acelerar o treinamento, mas pode afetar a precisão na detecção de objetos pequenos.
- J) `device =0`: Especifica o dispositivo onde o treinamento ocorrerá. O valor 0 geralmente corresponde à **primeira GPU** disponível. Caso fosse configurado para `cpu`, o treinamento seria realizado na unidade de processamento central, mas o uso de GPU acelera significativamente o

processo.

- K) *mosaic* =1.0: A técnica de ***mosaic*** combina 4 imagens em uma única imagem de entrada, aumentando a variabilidade dos dados. O valor 1.0 indica que o mosaico será utilizado ao máximo, permitindo que o modelo aprenda diferentes contextos e variações de objetos.
- L) *mixup* =0.3: O ***mixup*** é uma técnica de aumento de dados que mistura aleatoriamente duas imagens e suas respectivas anotações. O valor 0.3 indica a intensidade dessa mistura, ou seja, 30% das imagens serão misturadas para gerar novas imagens de treinamento.
- M) *augment*=*True* : Habilita o ***data augmentation*** (aumento de dados), aplicando técnicas como rotação, zoom, corte e distorção nas imagens. Isso gera variações adicionais nos dados de treinamento, o que ajuda a evitar o *overfitting* e melhora a generalização do modelo.

O modelo YOLOv8n foi selecionado devido à sua eficiência em cenários com restrições de hardware, oferecendo um equilíbrio adequado entre precisão e desempenho. A arquitetura do YOLOv8 apresenta melhorias significativas em relação às versões anteriores, incluindo otimizações na detecção de objetos e na velocidade de inferência. (Yaseen, 2024)

Para aumentar a diversidade dos dados e melhorar a robustez do modelo, foram aplicadas técnicas de aumento de dados (*data augmentation*). O parâmetro *mosaic*=1.0 permitiu a composição de imagens mosaico, enquanto *mixup*=0.3 adicionou variações adicionais através da mistura de diferentes imagens. Essas técnicas são reconhecidas por melhorar a performance de modelos de detecção de objetos, aumentando a diversidade e a quantidade de dados de treinamento. (Kaur; Khehra; Mavi, 2021)

3.5 Roboflow

O treinamento realizado por meio da plataforma *Roboflow* utilizou o modo de treinamento *Roboflow 3.0 Object Detection (Fast)*, que se destacou pela sua eficiência e simplicidade. A plataforma demonstrou excelentes resultados ao aplicar recursos como auto-orientação das imagens, otimizando

automaticamente as entradas do modelo.

Todo o processo foi conduzido no modo gratuito, o que ressalta a acessibilidade da ferramenta sem comprometer a qualidade do treinamento, evidenciando a robustez e a eficácia da plataforma mesmo em configurações básicas.

3.6 API

A API foi desenvolvida utilizando a linguagem de programação *Python*, com o objetivo de consumir os dados processados pela rede neural e atuar como um ponto central de integração entre os sistemas consumidores de informações e a rede neural. Essa integração foi viabilizada por meio de bibliotecas especializadas, que desempenham papéis fundamentais em diferentes etapas do processo. Entre elas, destacam-se:

- A) *FastAPI*, utilizada para construir uma interface de comunicação rápida, eficiente e escalável para o consumo dos dados da rede;
- B) *Supervision*, responsável por facilitar o monitoramento e a análise dos dados processados;
- C) *Ultralytics*, que fornece suporte direto à implementação e ao uso da rede neural baseada no modelo YOLO.

3.7 Frontend

O desenvolvimento da interface web foi um componente fundamental do sistema, utilizando tecnologias modernas para garantir desempenho e usabilidade. O *Next.js* foi escolhido como *framework* principal devido à sua renderização híbrida, que combina *Server-Side Rendering* (SSR) e *Static Site Generation* (SSG). Essa abordagem proporciona carregamentos mais rápidos e uma experiência do usuário otimizada.

A arquitetura da interface foi estruturada com a biblioteca *Mantine*, que se destacou por suas funcionalidades avançadas, como sistema de temas personalizável, suporte nativo ao modo escuro e compatibilidade otimizada com

SSR no *Next.js*. Essas características ofereceram uma base robusta e moderna para a construção da interface gráfica.

A aplicação foi desenvolvida em *JavaScript* e incorporou a biblioteca *React Query* para gerenciamento eficiente de estados e comunicação com o *backend*. Essa combinação tecnológica possibilitou uma interface intuitiva, responsiva e otimizada, garantindo atualização em tempo real e integração eficiente com os sistemas subjacentes.

4. Resultados

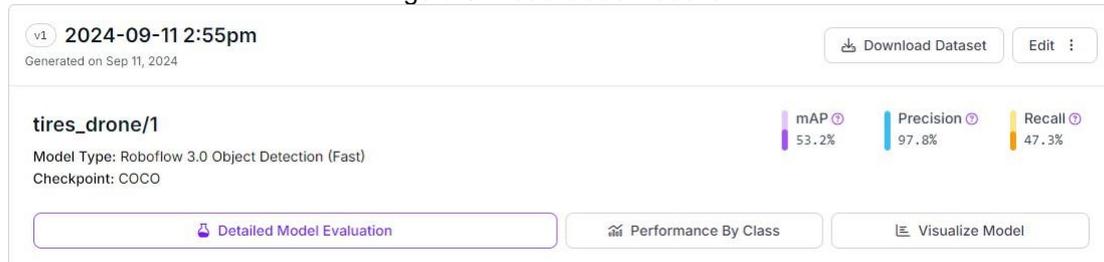
Os resultados obtidos demonstram a eficácia do modelo na detecção de pneus em imagens capturadas por drones, com indicadores de precisão, *recall* e *mean average precision* (mAP) que refletem a capacidade do modelo em identificar objetos de forma consistente em diferentes condições. Além disso, o sistema desenvolvido mostrou-se aplicável no monitoramento de focos de dengue, gerando informações georreferenciadas que podem auxiliar a vigilância sanitária.

4.1 Desempenho do Modelo

4.1.1 RoboFlow

O modelo de detecção de objetos foi treinado utilizando o modo *Roboflow 3.0 Object Detection (Fast)* da plataforma *Roboflow*, explorando os recursos disponíveis no modo gratuito, como a auto-orientação automática das imagens. Após o treinamento, o modelo foi avaliado com base em métricas amplamente utilizadas em visão computacional: precisão (*Precision*), revocação (*Recall*) e média da precisão média (*mean Average Precision* - mAP).

Figura 3. Resultados Roboflow



Fonte: Elaborado pelo Autor, 2025.

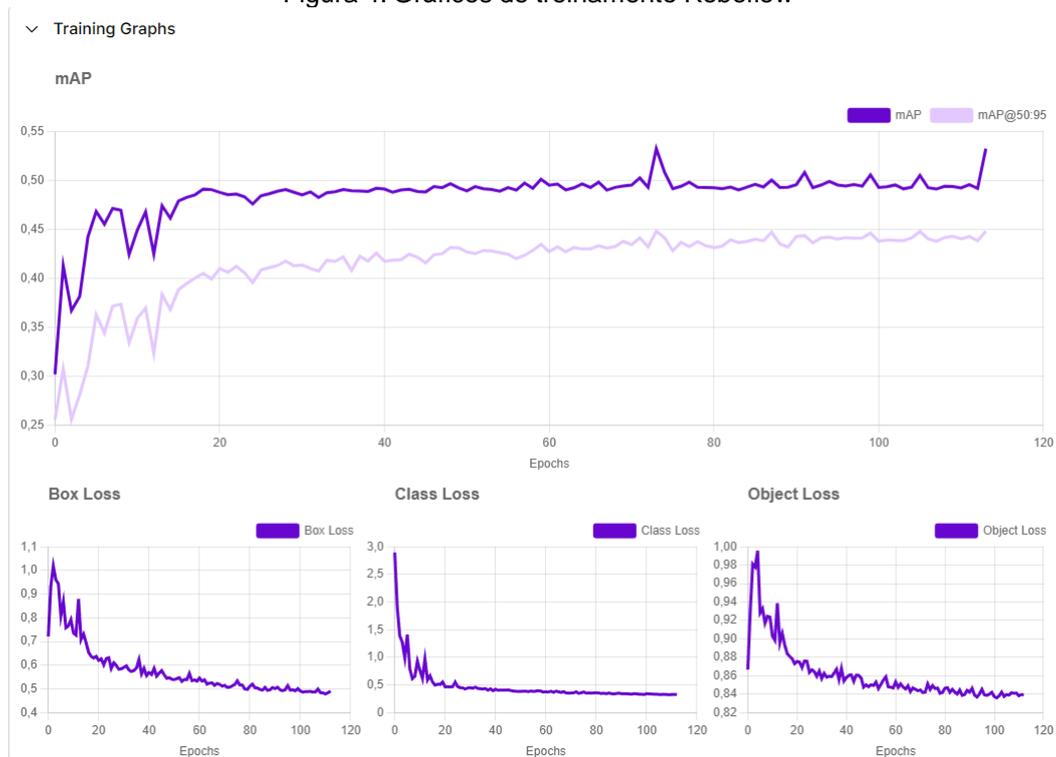
Precisão (*Precision*): o modelo alcançou um índice de 97,8%, indicando que a maioria das predições positivas realizadas foram corretas, o que reflete um alto nível de confiabilidade nas detecções realizadas.

Revocação (*Recall*): apresentou um valor de 47,3%, mostrando que quase metade dos objetos relevantes no conjunto de dados foi identificada corretamente, destacando um potencial para melhoria na cobertura das detecções.

Média da Precisão Média (*mAP*): o mAP foi de 53,2%, demonstrando um equilíbrio entre precisão e revocação ao considerar a acurácia do modelo em todas as classes detectadas.

Esses resultados evidenciam a capacidade do modelo em realizar detecções com alta precisão, embora haja oportunidades para aprimorar a identificação completa de todos os objetos relevantes. Essas métricas são fundamentais para entender a eficácia do modelo no contexto do problema e orientar melhorias futuras no processo de treinamento e ajuste.

Figura 4. Gráficos de treinamento Roboflow



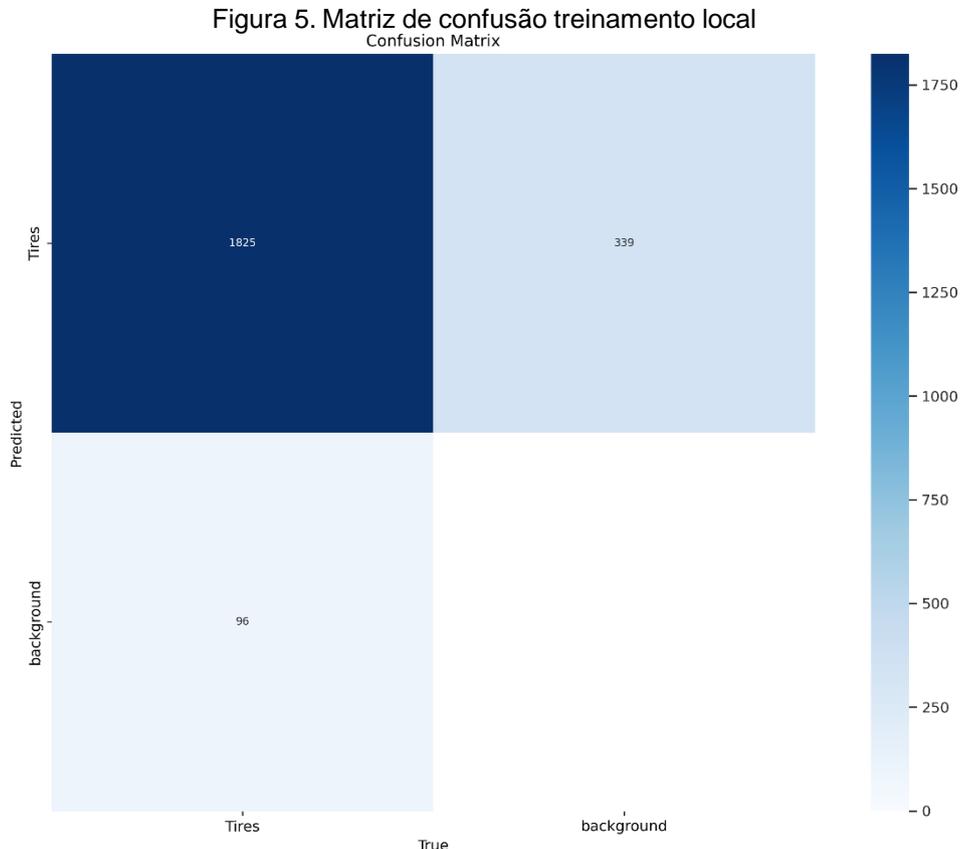
Fonte: Elaborado pelo Autor, 2025.

O treinamento, representado nas Figuras 3 e 8, foi realizado ao longo de 120 épocas e apresentou resultados satisfatórios, demonstrando um desempenho confiável para o modelo. Os valores obtidos para as métricas de avaliação, como precisão (*Precision*), revocação (*Recall*) e média da precisão média (*mean Average Precision - mAP*), evidenciam a consistência do treinamento e a adequação do modelo para a tarefa proposta, mesmo considerando possíveis melhorias futuras.

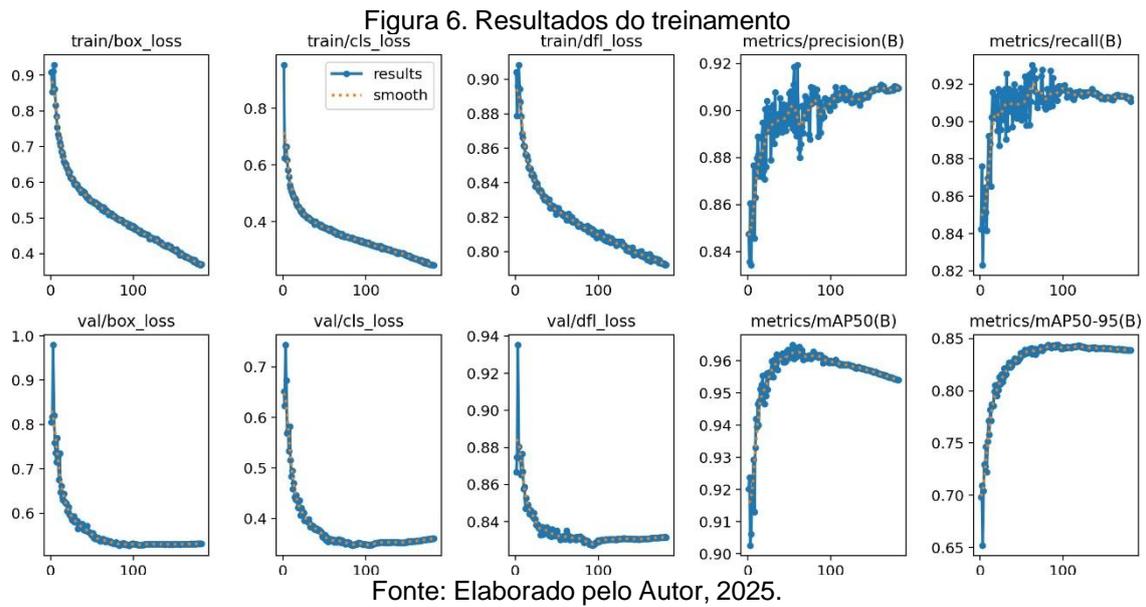
4.1.2 YoloV8 Local

O modelo de detecção de pneus demonstrou um desempenho robusto, alcançando métricas significativas na tarefa de detecção. A matriz de confusão revela que o modelo identificou corretamente 1.825 instâncias de pneus verdadeiros positivos, com apenas 339 falsos positivos e 96 falsos negativos, indicando uma alta precisão na classificação. A curva de precisão-*recall* apresenta um comportamento estável, mantendo uma precisão próxima a 1,0 até

aproximadamente 0,8 de *recall*, com um *mAP@0.5* de 0,962 para a classe de pneus.



As métricas de treinamento demonstram uma convergência adequada do modelo, com as curvas de perda (*boxloss*, *clsloss* e *dflloss*) exibindo um padrão de redução consistente tanto no conjunto de treino quanto de validação. A precisão do modelo atingiu valores superiores a 0,90, enquanto o *recall* se estabilizou em aproximadamente 0,91, evidenciando um equilíbrio satisfatório entre estas métricas fundamentais. O valor de *mAP50-95(B)* apresentou uma evolução positiva ao longo do treinamento, estabilizando-se em aproximadamente 0,84, o que indica um bom desempenho do modelo em diferentes limiares de IoU (*Intersection over Union*).



A análise das curvas de treinamento sugere que o modelo não apresentou *overfitting* significativo, dado que as perdas de validação acompanharam de forma consistente as perdas de treinamento. A estabilidade das métricas nas últimas épocas de treinamento indica que o modelo atingiu um ponto de convergência satisfatório, oferecendo um equilíbrio adequado entre precisão e generalização na tarefa de detecção de pneus.

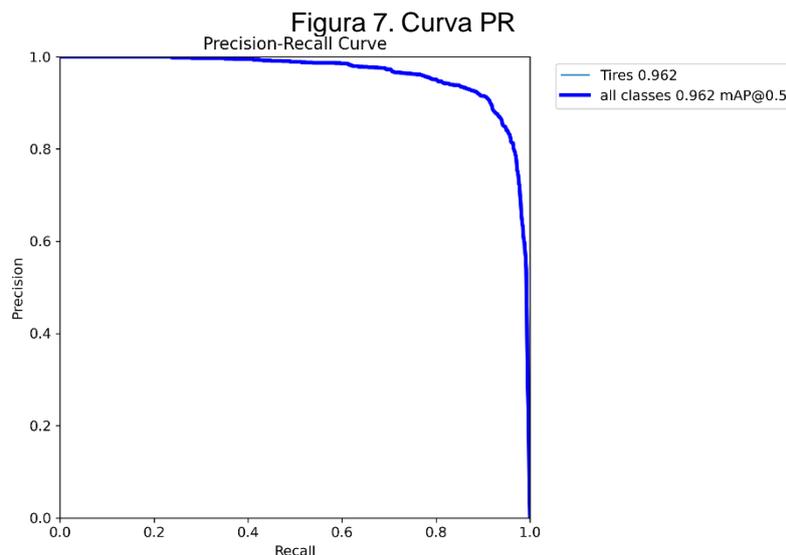


Figura 8. Imagem gerada através do treinamento



Fonte: Elaborado pelo Autor, 2025.

4.2 Desenvolvimento do Sistema

4.2.1 API

A API desenvolvida demonstrou alto desempenho e confiabilidade na integração do modelo YOLO com sistemas consumidores de informações, proporcionando uma solução eficiente para a detecção de objetos em imagens. Os testes realizados evidenciaram os seguintes resultados:

Desempenho no processamento de inferências: O tempo de resposta da API foi otimizado, com capacidade para lidar com imagens de alta complexidade visual. O uso de dispositivos acelerados por GPU (*CUDA*) garantiu maior eficiência computacional quando disponível.

Arquitetura modular e escalável: A implementação em FastAPI assegurou uma estrutura robusta, permitindo fácil integração com sistemas externos e simplificação no gerenciamento das operações de inferência.

Gerenciamento eficiente de recursos computacionais: Durante o ciclo de vida da aplicação, os modelos de aprendizado de máquina foram carregados e descarregados dinamicamente, otimizando o uso da memória e outros recursos do hardware.

Esses resultados demonstram a robustez e a eficiência da API como um componente essencial no pipeline de detecção de objetos, tornando-a uma solução viável e escalável para integração em sistemas de vigilância, diagnóstico ou monitoramento.

4.2.2 Sistema WEB

O sistema de monitoramento implementado consiste em três componentes principais:

- A) Um painel web desenvolvido usando *Next.js* e *TypeScript*;
- B) Um sistema de consumo de imagens baseado em *drone*;
- C) Um módulo de detecção alimentado por IA.

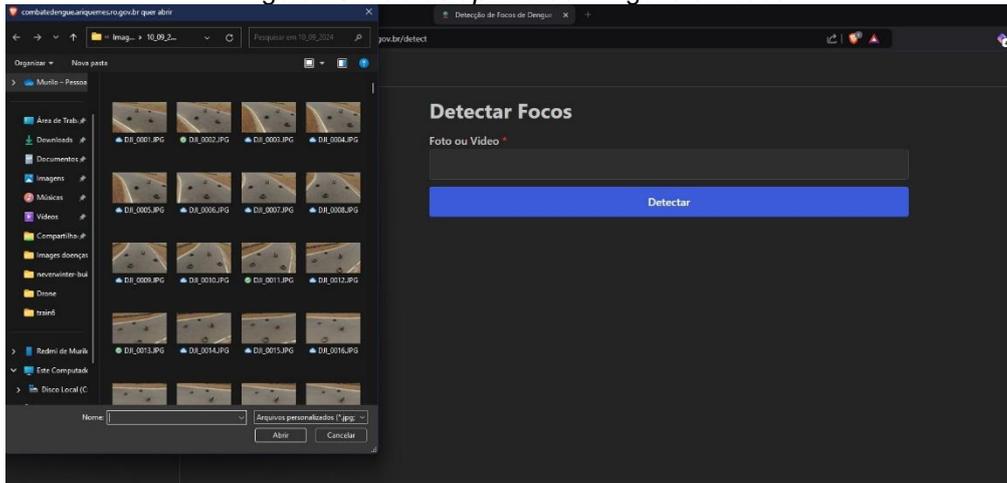
Figura 9. Tela inicial sistema de combate a dengue



Fonte: Elaborado pelo Autor, 2025.

O sistema oferece uma área central de *upload* claramente demarcada, onde os usuários podem carregar tanto imagens quanto vídeos para análise. Esta funcionalidade é indicada pelo rótulo "Foto ou Vídeo" com um asterisco vermelho, sinalizando um campo obrigatório. Após a seleção do arquivo, um botão "Detectar" em destaque, apresentado em azul, permite iniciar o processo de análise automaticamente.

Figura 10. Tela de *upload* de imagens e vídeos

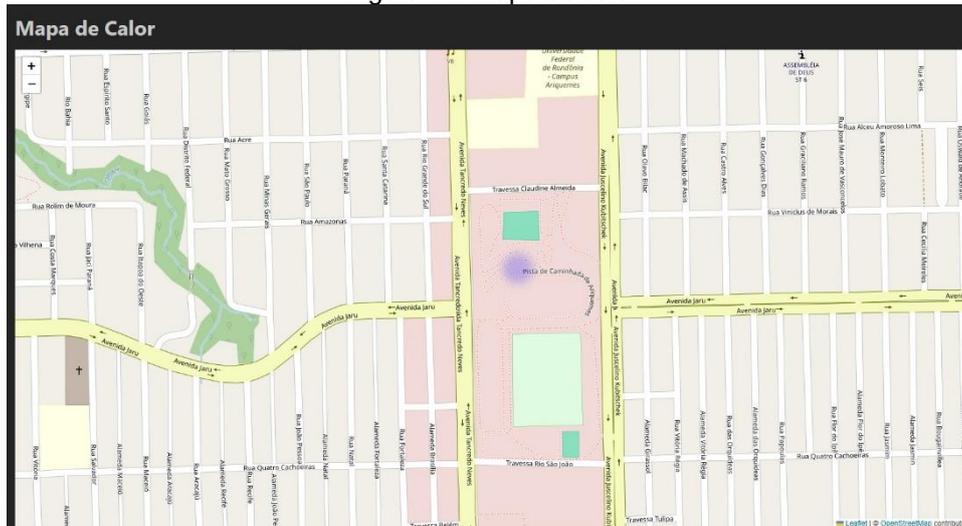


Fonte: Elaborado pelo Autor, 2025.

Após o *upload* da imagem, o sistema executa o processamento através do modelo de detecção treinado e fornece *feedback* imediato ao usuário. Como demonstrado na Figura 10, o sistema apresenta uma mensagem clara e objetiva dos resultados, indicando o número de focos de dengue detectados em uma barra de status verde. Esta resposta visual instantânea, exemplificada pela mensagem "4 focos de dengue detectados", permite que os agentes de saúde rapidamente avaliem a situação e tomem as medidas necessárias.

A interface mantém um alto contraste entre os elementos, com texto branco sobre fundo escuro e indicadores coloridos para status, seguindo boas práticas de design de interface. A confirmação visual em verde não apenas indica o sucesso da operação, mas também fornece uma resposta quantitativa precisa, fundamental para o registro e acompanhamento epidemiológico. Este *feedback* imediato otimiza o fluxo de trabalho dos agentes de saúde, permitindo uma rápida triagem das áreas analisadas e uma documentação eficiente dos focos encontrados.

Figura 11. Mapa de calor



Fonte: Elaborado pelo Autor, 2025.

O sistema incorpora uma funcionalidade avançada de visualização geoespacial através de um mapa de calor (*heatmap*) integrado, que proporciona uma representação visual dinâmica da distribuição espacial dos focos de dengue detectados. Como ilustrado na Figura 11, o mapa utiliza a biblioteca *Leaflet* em conjunto com dados do *OpenStreetMap* para apresentar a região monitorada, neste caso, uma área próxima à Universidade Federal de Rondônia, Campus Ariquemes.

A característica distintiva deste mapa de calor é sua capacidade de agregar dados georreferenciados das detecções realizadas pelo sistema. Cada imagem processada contribui com suas coordenadas geográficas para a construção de uma visualização térmica, onde áreas com maior concentração de focos são representadas por cores mais intensas - como demonstrado pela mancha azulada na região central do mapa. Esta representação visual evolui dinamicamente, intensificando-se em áreas onde múltiplas detecções são realizadas ao longo do tempo.

Esta abordagem de visualização oferece uma ferramenta crucial para a gestão epidemiológica, permitindo que as equipes de saúde identifiquem rapidamente *clusters* de focos de dengue e áreas de maior risco. A sobreposição do mapa de calor com o mapa urbano facilita a localização precisa das áreas afetadas e o planejamento eficiente das ações de combate ao vetor. O sistema

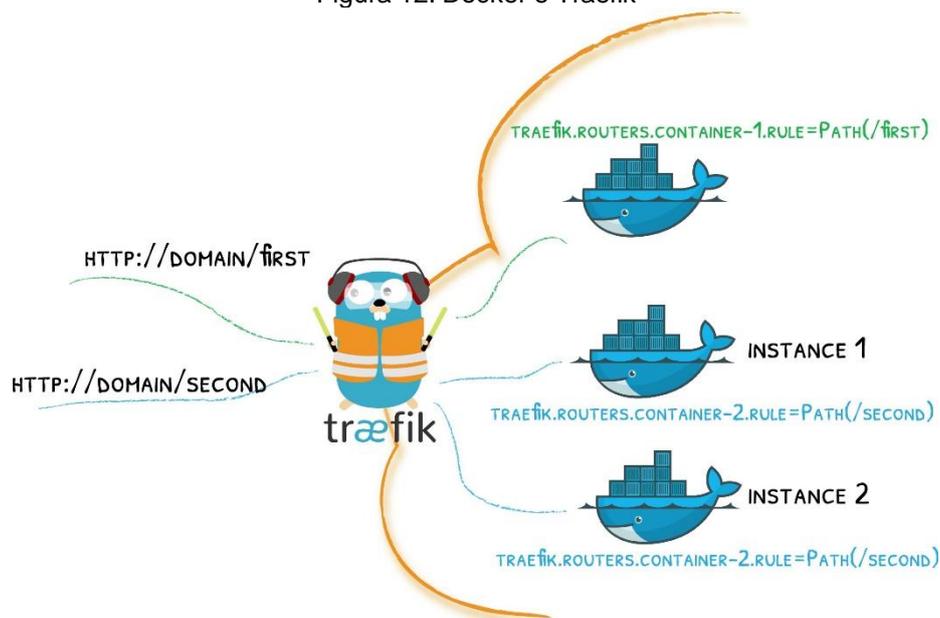
assim não apenas detecta os focos individualmente, mas também contribui para uma compreensão mais ampla do padrão de distribuição espacial dos criadouros do *Aedes aegypti* na região monitorada.

4.2.3 Infraestrutura e Arquitetura

A infraestrutura do sistema foi implementada utilizando *containers Docker*, garantindo isolamento e portabilidade dos serviços. Os principais componentes da infraestrutura incluem:

4.2.3.1 Containerização

Figura 12. Docker e Traefik



Fonte: Traefik Labs, 2025

- A) Utilização de *Docker* para encapsulamento dos serviços;
- B) Implementação de *Traefik* como *proxy* reverso, proporcionando:
 - a) Gerenciamento de SSL/TLS;
 - b) Roteamento dinâmico de requisições;
 - c) Monitoramento de saúde dos serviços.

4.2.3.2 Escalabilidade

- C) Orquestração de *containers* permitindo escala horizontal
- D) Distribuição eficiente de tráfego através do *Traefik*
- E) Auto-recuperação de serviços em caso de falhas

5. Discussão

Os resultados obtidos através do desenvolvimento e implementação do sistema de detecção de pneus demonstram a viabilidade e eficácia da integração entre *drones*, redes neurais e tecnologias web para o combate à dengue. O modelo YOLOv8 treinado apresentou métricas significativas, com uma precisão de 97,8% no ambiente Roboflow e uma matriz de confusão robusta no treinamento local, indicando alta confiabilidade na detecção de potenciais criadouros do *Aedes aegypti*.

A escolha da arquitetura YOLOv8 mostrou-se adequada para a aplicação, oferecendo um equilíbrio entre velocidade de processamento e precisão nas detecções. O uso de técnicas de *data augmentation* e a diversificação do *dataset* com diferentes condições de iluminação e ângulos contribuíram significativamente para a robustez do modelo, permitindo seu funcionamento em variadas condições ambientais.

A implementação da infraestrutura baseada em *containers Docker*, combinada com o *Traefik* como *proxy* reverso, demonstrou ser uma escolha acertada para garantir a escalabilidade e manutenibilidade do sistema. Esta arquitetura permite não apenas o crescimento horizontal da aplicação, mas também facilita sua implantação em diferentes ambientes operacionais, aspecto crucial para a adoção por diferentes municípios e órgãos de saúde.

O desenvolvimento da interface web com tecnologias modernas como *Next.js* e a implementação do mapa de calor proporcionaram uma ferramenta intuitiva e eficiente para os agentes de saúde. A visualização geoespacial dos focos detectados permite uma compreensão mais ampla da distribuição dos potenciais criadouros, facilitando o planejamento estratégico das ações de

combate ao vetor.

6. Conclusões e Trabalhos Futuros

Este trabalho demonstrou a viabilidade e eficácia de um sistema integrado para detecção automatizada de potenciais criadouros do *Aedes aegypti* utilizando drones e inteligência artificial. A implementação bem-sucedida do modelo YOLOv8, com altas taxas de precisão e *recall*, valida a abordagem proposta como uma ferramenta promissora para o auxílio no combate à dengue. A arquitetura modular e escalável desenvolvida, baseada em *containers* e microsserviços, permite não apenas a expansão do sistema, mas também sua adaptação para diferentes contextos e necessidades locais. A interface web intuitiva e o mapa de calor fornecem ferramentas valiosas para a visualização e análise dos dados coletados, facilitando a tomada de decisões pelos agentes de saúde pública. Como trabalhos futuros, sugere-se:

- A) A expansão do *dataset* com imagens de diferentes regiões e condições climáticas para aumentar a robustez do modelo;
- B) A implementação de funcionalidades de análise temporal para acompanhamento da evolução dos focos ao longo do tempo;
- C) O desenvolvimento de um sistema de alertas automáticos baseado na concentração de focos detectados;
- D) A integração com sistemas de informação em saúde existentes para melhor coordenação das ações de combate à dengue;
- E) A adaptação do modelo para detectar outros tipos de criadouros além de pneus;
- F) A realização de testes em campo em diferentes municípios para validação da eficácia do sistema em diferentes contextos urbanos.

Por fim, o sistema desenvolvido representa um avanço significativo na aplicação de tecnologias modernas para o combate a doenças transmitidas por vetores, oferecendo uma ferramenta promissora para auxiliar as autoridades de saúde pública no controle e prevenção da dengue.

REFERÊNCIAS

- AKTER, Rokeya; NAISH, Suchithra; GATTON, Michelle; BAMBRICK, Hilary; HU, Wenbiao; TONG, Shilu. Spatial and temporal analysis of dengue infections in queensland, australia: Recent trend and perspectives. **PLOS ONE**, Public Library of Science, v. 14, n. 7, p. 1–13, 07 2019. Disponível em:
<<https://doi.org/10.1371/journal.pone.0220134>>.
- ANDRIOLI, Denise Catarina; BUSATO, M. A.; LUTINSKI, J. Spatial and temporal distribution of dengue in brazil, 1990 - 2017. **PLoS ONE**, v. 15, 2020.
- BULLOCK, Joseph; LUCCIONI, Alexandra; PHAM, Katherine Hoffman; LAM, Cynthia Sin Nga; LUENGO-OROZ, Miguel. Mapping the landscape of artificial intelligence applications against covid-19. **Journal of Artificial Intelligence Research**, v. 69, p. 807–845, 2020.
- CHIMMULA, Vinay Kumar Reddy; ZHANG, Lei. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. **Chaos, Solitons & Fractals**, Elsevier, v. 135, p. 109864, June 2020. PMID: 32390691; PMCID: PMC7205623.
- GUILHERME, Iwldson; SANTOS, Silva; JÚNIOR, José Francisco de Oliveira; BARBOSA, Isnaldo Isaac; FELIPE, Luis; SILVA, Francisco Ferreira da; MAX, William; ROMÃO, Oliveira; MARQUES, Vitória Rejane; SANTOS dos; ROSALVO, Kelvy; CARDOSO, Alencar; CRISTINA, Caroline; ANDRADE, Silva de. Artificial neural network applied to reported dengue cases in maceió – alagoas. 2022.
- JOCHER, Glenn; CHAURASIA, Ayush; QIU, Jing. Ultralytics yolov8. 2023. URL <https://github.com/ultralytics/ultralytics>, 2023.
- JOHANSEN, I. C.; CARMO, Roberto L. do; ALVES, L. Correia; BUENO, Maria do Carmo Dias. Environmental and demographic determinants of dengue incidence in brazil. **Revista de salud publica**, v. 20 3, p. 346–351, 2018.
- KAUR, Parvinder; KHEHRA, Baljit Singh; MAVI, Er. Bhupinder Singh. Data augmentation for object detection: A review. In: **2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)**. [S.l.: s.n.], 2021. p. 537–543.
- REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. You Only Look Once: Unified, Real-Time Object Detection . In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Los Alamitos, CA, USA: IEEE Computer Society, 2016. p. 779–788. ISSN 1063-6919. Disponível em:
<<https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91>>.
- SCHEEPERS, Mathijs Jeroen. Virtualization and containerization of application infrastructure: A comparison. In:

21st twente student conference on IT. Enschede: University of Twente, 2014. v. 21. Citado na página 6. SOMMERVILLE, Ian. Engenharia de software, 9a. São Paulo, SP, Brasil, p. 529, 2011.

TERVEN, Juan; CÓRDOVA-ESPARZA, Diana-Margarita; ROMERO-GONZÁLEZ, Julio-Alejandro. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. **Machine Learning and Knowledge Extraction**, MDPI, v. 5, n. 4, p. 1680–1716, 2023.

VALDEZ-DELGADO, Kenia Mayela; GARCIA-SALAZAR, Octavio; MOO-LLANES, David A.; IZCAPA-TREVIÑO, Cecilia; CRUZ-PLIEGO, Miguel A.; DOMÍNGUEZ-POSADAS, Gustavo Y.; ARMENDÁRIZ-VALDEZ, Moisés O.; CORREA-MORALES, Fabián; CISNEROS-VÁZQUEZ, Luis Alberto; ORDÓÑEZ-GONZÁLEZ, José Genaro; FERNÁNDEZ-SALAS, Ildefonso; DANIS-LOZANO, Rogelio. Mapping the urban environments of aedes aegypti using drone technology. **Drones**, v. 7, n. 9, 2023. ISSN 2504-446X. Disponível em: <<https://www.mdpi.com/2504-446X/7/9/581>>. Acesso em 21 de dezembro de 2024.

VALDEZ-DELGADO, K. M.; MOO-LLANES, D. A.; DANIS-LOZANO, R.; CISNEROS-VÁZQUEZ, Luis Alberto; FLORES-SUAREZ, A.; PONCE-GARCIA, G.; GARZA, C. E. Medina-De la; DÍAZ-GONZÁLEZ, E. E.; FERNÁNDEZ-SALAS, I. Field effectiveness of drones to identify potential aedes aegypti breeding sites in household environments from tapachula, a dengue-endemic city in southern mexico. **Insects**, v. 12, 2021.

WAZLAWICK, Raul Sidnei. **Metodologia de pesquisa para ciência da computação.** Rio de Janeiro: Elsevier, 2008. 184 p. ISSN 978-85-352-3522-7. YASEEN, Muhammad. What is yolov9: An in-depth exploration of the internal features of the next-generation object detector. **arXiv preprint arXiv:2409.07813**, 2024. Citado na página 5.

Yaseen, Muhammad. What is YOLOv9: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector. **arXiv e-prints**, p. arXiv:2409.07813, set. 2024. Citado na página 11.

ZENG, D.; CAO, Zhidong; NEILL, Daniel B. Artificial intelligence-enabled public health surveillance—from local detection to global epidemic monitoring and control. **Artificial Intelligence in Medicine**, p. 437 – 453, 2020. Citado na página 4.