

ANTT – Agência Nacional de Transportes Terrestres
RDT – Recurso de Desenvolvimento Tecnológico

RELATÓRIO FINAL

**PROJETO 12 - APLICAÇÃO DE TECNOLOGIAS AVANÇADAS DE INTELIGÊNCIA
ARTIFICIAL E VISÃO COMPUTACIONAL PARA ATENDIMENTO DAS DEMANDAS
OPERACIONAIS E MONITORAMENTO RODOVIÁRIO**

Autopista Litoral Sul

31/08/2025

SUMÁRIO

1. DESCRIÇÃO DO PROJETO	5
1.1 Título.....	5
1.2 Resumo	5
1.3 Palavras-chave	5
1.4 Justificativa	6
1.5 Objetivos.....	6
1.6 Organização do trabalho	8
1.7 Período de execução	9
1.8 Cronograma de execução.....	9
1.9 Local de execução.....	10
1.10 Equipe executora.....	10
2. MÉTODOS E TÉCNICAS UTILIZADAS.....	10
2.1 Metodologia de Desenvolvimento – Scrum.....	10
2.2 Técnicas para Contagem de Veículos	11
2.3 Técnicas para Detecção de Eventos	12
2.4 Integração, Armazenamento e Visualização de Dados	12
2.5 Síntese.....	12
3. CRIAÇÃO DE BASE DE IMAGENS INICIAL PARA O TREINAMENTO DA INTELIGÊNCIA ARTIFICIAL	13
3.1 Manipulação de Dataset.....	13
3.2 Conclusão.....	15
4. TREINAMENTO DA INTELIGÊNCIA ARTIFICIAL.....	15
4.1 Modelo de animais	15
4.2 Modelo de Animais Versão 1.0.....	15
4.3 Modelo de Contagem.....	20
4.4 Modelo de Contagem Versão 1.0.....	21
4.5 Modelo de Eventos	26
4.6 Modelo de Eventos Versão 1.0	26
4.7 Modelo de Objetos.....	31
4.8 Modelo de Objetos Versão 1.0.....	31
4.9 Modelo de Incêndio	35
4.10 Modelo de Incêndio Versão 1.0.....	35
4.11 Modelo de Pessoas.....	40

4.12 Modelo de Pessoas Versão 1.0.....	40
5. APRIMORAMENTO DA APLICAÇÃO DE DETECÇÃO DE EVENTOS.....	45
5.1 Modelo de Animais Versão 2.0.....	46
5.2 Modelo de Eventos Versão 2.0.....	52
5.3 Modelo de Eventos Versão 3.0.....	57
5.4 Modelo de Incêndio Versão 2.0.....	62
5.5 Modelo de Pessoas Versão 2.0.....	67
5.6 Modelo de Pessoas Versão 3.0.....	71
5.7 Modelo de Objetos Versão 2.0.....	77
5.8 Conclusão dos modelos.....	82
6. CONFIGURAÇÃO EDGE COMPUTING EM SERVIDORES TEMPORÁRIOS.....	83
7. APRIMORAMENTO DA APLICAÇÃO DE CONTAGEM DE VEÍCULOS.....	84
7.1 Modelo de Contagem Versão 2.0.....	84
7.2 Modelo de Contagem Versão 3.0.....	89
8. CONFIGURAÇÃO KUBERNETES EDGE EM SERVIDORES TEMPORÁRIOS.....	94
9. AQUISIÇÃO DE EQUIPAMENTOS E PREPARAÇÃO DE INFRAESTRUTURA NA RODOVIA PARA PROCESSAMENTO DO MODELO DE CONTAGEM DE VEÍCULOS.....	94
10. AQUISIÇÃO DOS SERVIDORES PARA SUSTENTAÇÃO DOS MODELOS.....	95
11. CONFIGURAÇÃO E PREPARO DE INFRAESTRUTURA DE TI ARTERIS.....	95
12. APRIMORAMENTO DA APLICAÇÃO DE CONTAGEM DE VEÍCULOS.....	96
12.1 Modelo de Contagem Versão 4.0.....	96
13. CRIAÇÃO ARQUITETURA INGESTÃO DE DADOS.....	102
13.1 Engenharia.....	102
13.2 Data Lake Storage.....	102
13.3 Workspace Azure Databricks.....	104
13.4 Modelo de dados.....	106
13.5 Dicionário de dados.....	107
13.6 Fluxo de carga diária.....	109
13.7 Ingestão da planilha de localizações das câmeras.....	110
13.8 Job Cluster.....	111
13.9 Fluxo de reprocessamento.....	112
14. DEPLOY EM PRODUÇÃO E VALIDAÇÃO DO SERVIÇO DE CONTAGEM DE VEÍCULOS.....	113
15. PROCESSAMENTO DE DADOS E CONSTRUÇÕES DE KPIS.....	114
15.1 Power BI.....	114
15.2 Conjunto de dados.....	114

15.3 Medidas e visuais do Dashboard.....	116
16. REAPONTAMENTO DE SERVIDORES ADQUIRIDOS	116
17. IMPLEMENTAÇÃO DE MLOPS	117
18. DEPLOY EM PRODUÇÃO E VALIDAÇÃO DO SERVIÇO DE DETECÇÃO DE EVENTOS	118
19. REFINAMENTO FINAL DA SOLUÇÃO.....	119
20. HOMOLOGAÇÃO, COLETA DE RESULTADOS E ELABORAÇÃO DE RELATÓRIO FINAL	120
Execução de plano de testes de homologação com stakeholders.....	120
Coleta e consolidação de métricas de desempenho.....	121
Análise dos resultados obtidos.....	123
Conclusão de relatório final técnico e executivo.	126
21. CONCLUSÕES, CONSIDERAÇÕES FINAIS E PRODUTOS	130
21.1 Alcance dos Objetivos Gerais e Específicos.....	131
21.2 Avaliação dos Métodos e Técnicas.....	131
21.3 Avaliação dos Resultados em Relação ao Plano de Trabalho.....	132
21.4 Utilidade Prática e Contribuições	135
21.5 Vantagens, Deficiências e Possibilidades de Melhoria.....	136
21.6 Viabilidade Econômica e Custo-Benefício	136
21.7 Potencialidades e Limitações.....	136
21.8 Impactos Ambientais	137
21.9 Quadro Comparativo – Situação Atual x Solução Proposta	137
21.10 Produtos Gerados.....	137
21.11 Transferência de Tecnologia e Conhecimento.....	138
22. REFERÊNCIAS BIBLIOGRÁFICAS.....	138
23. ANEXOS	138

1. DESCRIÇÃO DO PROJETO

1.1 Título

Aplicação de tecnologias avançadas de inteligência artificial e visão computacional para atendimento das demandas operacionais e monitoramento rodoviário.

1.2 Resumo

O projeto “Desenvolvimento de Estudos e Pesquisas para Proposição e Estruturação de Modelos de Inteligência Artificial Aplicados à Detecção e Monitoramento de Eventos Críticos em Rodovias Concessionadas, com Foco no Apoio ao Monitoramento pela Agência Nacional de Transportes Terrestres – ANTT”, desenvolvido por meio de Recurso de Desenvolvimento Tecnológico (RDT), teve início no dia 1º de novembro de 2024, a partir da assinatura do contrato entre a Concessionária Arteris e a) Programmers. O presente documento é o primeiro Relatório de Acompanhamento semestral do projeto, compreendendo o período entre 1º de novembro de 2024 e 30 de maio de 2025.

O projeto foi elaborado com o objetivo de desenvolver modelos de inteligência artificial voltados à detecção de veículos, cálculo de velocidade média, identificação de animais, incêndios, objetos, pessoas e eventos críticos nas rodovias, visando apoiar e aprimorar o monitoramento realizado pela Concessionária.

Neste primeiro Relatório de Acompanhamento, serão apresentadas as atividades previstas nas etapas do projeto, a saber:

- Etapa 1 – Engenharia de Dados: Desenvolvimento de fluxos para extração, tratamento e armazenamento de dados capturados por câmeras nas rodovias, com processamento no Azure Databricks e visualização em dashboards interativos no Power BI, assegurando precisão nos indicadores estratégicos.
- Etapa 2 – Infraestrutura: Estruturação de um sistema de processamento inteligente de vídeos de tráfego, responsável por transformar gravações em dados organizados sobre contagem de veículos e eventos adversos, classificação por tipo e cálculo de velocidade média, de forma automatizada e escalável, no caso da contagem, e classificação de pedestres, objeto na pista, animais na pista, incêndio e carro parado, para o caso de eventos adversos.
- Etapa 3 – Modelos de IA: Treinamento e validação de modelos de detecção baseados em deep learning, utilizando arquiteturas como YOLOv8, para diferentes cenários em rodovias, incluindo animais, pessoas, objetos, incêndios e eventos relevantes para emissão de alertas operacionais.

As atividades foram executadas conforme cronograma aprovado pela Agência Nacional de Transportes Terrestre.

1.3 Palavras-chave

Inteligência Artificial; Visão Computacional; Aplicabilidade de Tecnologia; Cloud; Serviços Operacionais; Atendimento na rodovia; Monitoramento de tráfego; Contrato de concessão.

1.4 Justificativa

A implementação de tecnologias avançadas para o monitoramento de tráfego oferece uma série de vantagens significativas em comparação com os métodos tradicionais. Essas tecnologias permitem uma coleta de dados mais precisa e em larga escala, proporcionando uma visão abrangente do tráfego e das condições da rodovia. Além disso, a utilização dessas ferramentas permite uma maior assertividade no atendimento a demandas operacionais e no monitoramento rodoviário, garantindo além de maior segurança viária aos usuários, uma eficiência operacional significativa na fiscalização dos ativos rodoviários.

Além destes benefícios, a aplicação destas tecnologias também geram um impacto indireto, porém significativo no tocante ao meio ambiente. Propor a diminuição de janela de inspeção de um veículo que inspeciona 24 horas por dia um segmento de rodovia pelo monitoramento remoto através das soluções inovadoras, impactando a redução de emissão de poluentes, por exemplo.

A primeira experiência sobre o tema foi a realização do projeto desenvolvido em 2022 na Autopista Régis Bittencourt, denominado “Análise de viabilidade do uso da tecnologia de visão computacional na rodovia BR-116 no trecho sob concessão da autopista régis bittencourt”.

Através do conceito de MVP (Mínimo Produto Viável), desenvolveu-se uma solução/modelo de inteligência artificial e visão computacional capaz de capturar a volumetria de tráfego, a categoria dos veículos e demais requisitos mínimos de um contador de tráfego tradicional (SAT's). O principal objetivo foi analisar a viabilidade de substituição do modelo de contagem tradicional pela utilização de novas técnicas e tecnologias.

Os resultados obtidos atingiram a expectativa, onde o comparativo de detecção entre as duas modalidades ficou bem próximo ao 100%:

Comparativo	Diurno (aprox.)	Noturno (aprox.)
Volume Manual Vs. Volume Video Analytics	97%	93%
Volume Manual Vs. Volume SAT	90%	93%
Velocidade SAT Vs. Velocidade Video Analytics	92%	92%

Tabela 1 – Resultados do modelo de contagem de veículos – Projeto 13 Régis Bittencourt

Dando continuidade na exploração do uso de novas técnicas e tecnologias na operação de monitoramento de tráfego rodoviário, ainda no conceito de MVP (Mínimo Produto Variável), em 2023 realizou-se um projeto na Autopista Litoral Sul, denominado “Desenvolvimento de tecnologia e análise de viabilidade do uso de visão computacional para identificação de incidentes e necessidades de atendimento operacional na BR-116/PR no trecho sob concessão da Autopista Litoral Sul.”.

Neste projeto, desenvolveu-se um modelo de inteligência artificial e visão computacional para identificação automática de veículos parados na rodovia ou em acostamento, que necessitem de atendimento por parte das equipes operacionais e identificação de ocorrências que geram paralisação de tráfego ou congestionamento, de forma a ter maior assertividade e rapidez na prestação de serviço ao usuário da via, e contribuir para a segurança viária.

Assim como o primeiro modelo desenvolvido, a solução tecnológica deste projeto também foi aprovada e validada, demonstrando sua eficácia na identificação de possíveis incidentes e agilidade no disparo de alertas para atendimento ao usuário na rodovia.

Diante desse cenário positivo de resultados para os dois MVP's desenvolvidos, este projeto surge como uma iniciativa estratégica para aproveitar os benefícios das tecnologias de inteligência artificial e visão computacional e avançar para uma segunda fase destes projetos, a de implementação de suas funcionalidades em uma escala maior e em ambiente de produção.

1.5 Objetivos

1.5.1 Objetivo Geral

O presente plano de trabalho é uma continuação do projeto 13 da Autopista Régis Bittencourt e do projeto 11 da Autopista Litoral Sul, tendo como objetivo geral a realização de um projeto piloto na Autopista Litoral Sul que inclui a implementação e o aprimoramento das soluções baseadas em tecnologias avançadas de inteligência artificial e visão computacional desenvolvidas anteriormente, a fim de homologar com a Agência Nacional de Transportes Terrestres (ANTT) a utilização de forma efetiva dessas novas soluções de tecnologia em ambiente de produção.

1.5.2 Objetivos Específicos

Este projeto tem como objetivos específicos:

- Aprimorar e implementar em ambiente de produção o modelo de contagem, classificação e registro de velocidade de veículos, desenvolvido no projeto 13 da Autopista Régis Bittencourt, integrando-o ao sistema de monitoramento das rodovias da Litoral Sul, capaz de processar as imagens através de câmeras instaladas nos mesmos pontos onde os tradicionais contadores de tráfego (SAT's) estão localizados.
- Aprimorar o modelo desenvolvido no projeto 11 da Autopista Litoral Sul, com a identificação automática de ocorrências que geram paralisação de tráfego ou congestionamento, com a adição de novas funcionalidades de detecção como: pedestres, animais, objetos na pista, incêndios, veículos parados, além de implementar o modelo em ambiente de produção em um trecho experimental delimitado, buscando maior assertividade e rapidez na prestação de serviço ao usuário da via;
- Elaborar um relatório técnico de desempenho dos modelos em sua utilização efetiva, analisando a eficácia e precisão e levantando possíveis pontos de melhoria na operacionalização dos modelos;
- Sugerir a revisão do parâmetro contratual da Inspeção de Tráfego para a alteração de obrigatoriedades referentes ao monitoramento das rodovias com a utilização de novas tecnologias; Validar a substituição dos tradicionais contadores de tráfego (SAT) pela utilização do modelo de inteligência artificial e visão computacional para contagem, classificação e registro de velocidade dos veículos;
- Substituição/Diminuição das janelas de inspeção de tráfego em campo com a utilização dos modelos de identificação de incidentes através das câmeras de monitoramento nas rodovias.

1.6 Organização do trabalho

1. Criação de base de imagens inicial para o treinamento da Inteligência Artificial
2. Treinamento da IA
3. Aprimoramento da aplicação de detecção de eventos
4. Configuração Edge Computing em servidores temporários
5. Configuração Kubernetes Edge em servidores temporários
6. Aquisição de equipamentos e preparação de infraestrutura na rodovia para processamento do modelo de contagem de veículos
7. Aquisição dos servidores para sustentação dos modelos
8. Configuração e preparo de infraestrutura de TI Arteris
9. Aprimoramento da aplicação de contagem de veículos
10. Criação arquitetura ingestão de dados
11. Deploy em produção e validação do serviço de contagem de veículos
12. Processamento de dados e construção de KPIs
13. Reapontamento de servidores adquiridos
14. Implementação de MLOPS
15. Deploy em produção e validação do serviço de detecção de eventos
16. Refinamento final da solução
17. Homologação, coleta de resultados e elaboração do relatório final

1.7 Período de execução

01/11/2024 a 31/08/2025.

1.8 Cronograma de execução

Etapa	Atividade		Período de execução - 10 meses													
			Mês 1	Mês 2	Mês 3	Mês 4	Mês 5	Mês 6	Mês 7	Mês 8	Mês 9	Mês 10				
1	Criação de base de imagens inicial para o treinamento da inteligência artificial	Programmers	X													
2	Treinamento da inteligência artificial	Programmers		X												
3	Aprimoramento da aplicação de detecção de eventos	Programmers			X											
4	Configuração Edge Computing em servidores temporários	Programmers			X											
5	Aprimoramento da aplicação de contagem de veículos	Programmers				X										
6	Configuração Kubernetes Edge em servidores temporários	Programmers				X										
7	Aquisição de equipamentos e preparação de infraestrutura na rodovia para processamento do modelo de contagem de veículos	Arteris	X	X	X	X										
8	Aquisição dos servidores para sustentação dos modelos	Arteris								X						
9	Configuração e preparo de infraestrutura de TI Arteris	Arteris								X						
10	Aprimoramento da aplicação de contagem de veículos	Programmers/Arteris					X									
11	Criação arquitetura ingestão de dados	Programmers					X									
12	Deploy em produção e validação do serviço de contagem de veículos	Programmers						X								
13	Processamento de dados e construções de KPIs	Programmers						X								
14	Reapontamento de servidores adquiridos	Programmers								X						
15	Implementação de MLOps	Programmers								X						
16	Deploy em produção e validação do serviço de detecção de eventos	Programmers									X					
17	Refinamento final da solução	Programmers										X				
18	Homologação, coleta de resultados e elaboração de relatório final	Programmers													X	

Imagem 1 – Cronograma de atividades do projeto

1.9 Local de execução

O trabalho será desenvolvido de forma remota pelas equipes responsáveis pela programação das tecnologias Programmers Informática LTDA, com supervisão e apoio da equipe da Arteris para tratativas de infraestrutura em campo.

1.10 Equipe executora

Maurício Vitorino – Coordenador TI Arteris
Cassio Negri – Coordenador Gestão Operacional Arteris
Victor Faria – Analista Gestão Operacional Arteris
Ariel Bueno – Scrum Master Programmers
Douglas Sammur – Developer Programmers
Pablo Henrique – Developer Programmers
Bruno Roza – Infraestrutura TI Programmers
Lizandro Santos – Developer Programmers
Hellen Miranda – Engenharia de Dados Programmers

2. MÉTODOS E TÉCNICAS UTILIZADAS

O desenvolvimento do projeto foi estruturado a partir de métodos ágeis e técnicas avançadas de visão computacional, buscando equilíbrio entre inovação tecnológica, consistência nos resultados e capacidade de adaptação durante a execução.

2.1 Metodologia de Desenvolvimento – Scrum

Optou-se pelo uso da metodologia ágil Scrum, amplamente empregada em projetos de tecnologia pela sua flexibilidade e foco em entregas incrementais.

O Scrum organiza o trabalho em sprints (ciclos curtos de tempo de 2 semanas), nos quais são priorizadas e concluídas atividades específicas do projeto. Para garantir acompanhamento contínuo e alinhamento entre os times, foram adotadas as seguintes práticas:

Daily Meetings (Dailies): reuniões rápidas de 15 minutos realizadas diariamente para identificar avanços, desafios e necessidades imediatas.

Sprint Planning: encontro no início de cada ciclo para planejar as atividades a serem realizadas.

Refinamentos: sessões destinadas a revisar o backlog, ajustando prioridades conforme feedbacks recebidos ou novas demandas.

Sprint Reviews: apresentações formais dos resultados obtidos a cada ciclo, permitindo que stakeholders acompanhassem o progresso e sugerissem melhorias.

Retrospectivas: reuniões ao final de cada sprint para avaliar o que funcionou bem, o que poderia ser melhorado e quais ações seriam tomadas para o ciclo seguinte.

Essa abordagem possibilitou ajustes rápidos e direcionados ao longo da execução, assegurando que tanto os requisitos técnicos quanto os objetivos estratégicos fossem atendidos com maior precisão.

2.2 Técnicas para Contagem de Veículos

Uma das frentes do projeto foi dedicada à **contagem de veículos** (carros, caminhões e motocicletas) e ao cálculo de sua velocidade média.

Para alcançar esse objetivo, foram aplicadas as seguintes técnicas:

- Treinamento de modelos personalizados com base no algoritmo YOLOv11 (Ultralytics). O dataset utilizado foi construído de forma a refletir as condições reais das vias monitoradas, incluindo iluminação variável, ângulos distintos e diferentes densidades de tráfego.
- Inferência em produção via NVIDIA Triton Inference Server, escolhido por sua capacidade de paralelizar processos de inferência em GPU, reduzindo a latência de resposta. O Triton também possibilitou melhor aproveitamento dos recursos computacionais, sendo fundamental para manter desempenho estável em horários de pico.
- Cálculo de velocidade média implementado em módulos de pós-processamento, que analisam o deslocamento dos veículos entre quadros consecutivos. O método foi ajustado para lidar com casos de sobreposição de veículos e diferentes perspectivas de câmeras.
- Filtros e regras complementares foram aplicados para minimizar erros de contagem em cenários desafiadores, como tráfego intenso ou detecção parcial de veículos.

2.3 Técnicas para Detecção de Eventos

A segunda frente do projeto foi voltada à detecção de eventos em tempo real, contemplando a identificação de: veículos, pessoas, animais, objetos na pista e incêndios.

As técnicas utilizadas incluíram:

- Pipelines otimizados com NVIDIA DeepStream, permitindo processar fluxos de vídeo contínuos em tempo real, com baixo consumo de recursos e suporte a múltiplos modelos rodando simultaneamente.
- Modelos especializados por categoria de evento, cada um ajustado para responder a um tipo de ocorrência. Esse desenho modular possibilitou aplicar thresholds e parâmetros diferentes por classe, melhorando a precisão global do sistema.
- Integração com canais de alerta, em especial o Microsoft Teams, que recebeu notificações de eventos validados. A lógica de envio foi projetada para evitar redundância, de modo que o time operacional recebesse alertas apenas em situações relevantes.

2.4 Integração, Armazenamento e Visualização de Dados

Um diferencial da solução foi a integração ponta a ponta dos dados, desde a captura do vídeo até a visualização em dashboards de negócio.

- A API Digifort foi utilizada para acessar os fluxos de vídeo das câmeras em tempo real, servindo de ponto central para ingestão de dados.
- Tanto os serviços de contagem quanto os de eventos geraram arquivos JSON padronizados, estruturados para conter informações essenciais (identificação da câmera, timestamp, tipo de evento, métricas associadas).
- Esses arquivos foram armazenados em Blob Storage no Microsoft Azure, garantindo escalabilidade e segurança.
- O processamento e a transformação dos dados foram realizados em Databricks, que estruturou as informações para posterior análise.
- Por fim, os dados foram disponibilizados em dashboards no Power BI, divididos em dois ambientes distintos (um para contagem, outro para eventos), possibilitando que gestores e stakeholders acompanhassem métricas de forma clara e interativa.

2.5 Síntese

A combinação de métodos ágeis (Scrum) com técnicas de visão computacional e pipelines otimizados de processamento permitiu construir uma solução robusta, escalável e alinhada às necessidades do cliente. Enquanto o Scrum assegurou transparência e colaboração contínua, as

tecnologias empregadas garantiram baixa latência, confiabilidade dos resultados e integração fluida com ferramentas analíticas corporativas.

3. CRIAÇÃO DE BASE DE IMAGENS INICIAL PARA O TREINAMENTO DA INTELIGÊNCIA ARTIFICIAL

A biblioteca Albumentations é amplamente utilizada para realizar aumentos de dados (data augmentation) em projetos de visão computacional, permitindo a geração de variações de imagens que ampliam o dataset de treinamento. Isso ajuda modelos de aprendizado de máquina a se tornarem mais robustos e a generalizarem melhor em diferentes condições. Este tópico descreve alguns dos efeitos aplicáveis ao cenário atual, configurados especificamente para aumentar o dataset de imagens fornecidos pela Arteris.

3.1 Manipulação de Dataset Transformações e Configurações

1. HorizontalFlip

- **Descrição:** Realiza a inversão horizontal da imagem, criando uma variação espelhada.
- **Parâmetros:**
 - "p": 1.0: Probabilidade de aplicação do efeito. Com 1.0, a transformação será sempre aplicada.
- **Utilização:** Esse efeito é útil para gerar versões espelhadas da imagem, o que é especialmente vantajoso em cenários onde a orientação lateral não é relevante.

2. RandomBrightnessContrast

- **Descrição:** Ajusta aleatoriamente o brilho e o contraste da imagem.
- **Parâmetros:**
 - "brightness_limit": 0.2: Limite de ajuste do brilho, onde 0.2 representa uma variação de $\pm 20\%$.
 - "contrast_limit": 0.2: Limite de ajuste do contraste, onde 0.2 representa uma variação de $\pm 20\%$.
 - "p": 0.5: Probabilidade de aplicação. Neste caso, o efeito é aplicado em 50% das vezes.

- **Utilização:** Esta transformação é útil para simular condições de iluminação variadas, tornando o modelo mais robusto a diferentes níveis de luz.

3. HueSaturationValue

- **Descrição:** Altera aleatoriamente as tonalidades (hue), a saturação e o valor das cores da imagem.
- **Parâmetros:**
 - "hue_shift_limit": 20: Limite de ajuste da tonalidade em ± 20 unidades.
 - "sat_shift_limit": 30: Limite de ajuste da saturação em ± 30 unidades.
 - "val_shift_limit": 20: Limite de ajuste do valor (brilho) em ± 20 unidades.
 - "p": 0.5: Probabilidade de aplicação de 50%.
- **Utilização:** Esse efeito permite a simulação de diferentes variações de cor, úteis para melhorar a capacidade do modelo de reconhecer imagens em várias condições de iluminação e saturação.

4. Resize

- **Descrição:** Redimensiona a imagem para uma altura e largura específicas.
- **Parâmetros:**
 - "height": 512: Altura final da imagem em pixels.
 - "width": 512: Largura final da imagem em pixels.
- **Utilização:** O redimensionamento garante que todas as imagens estejam em um tamanho uniforme, adequado para o modelo de treinamento.

5. RandomCrop

- **Descrição:** Recorta aleatoriamente uma porção da imagem com dimensões definidas.
- **Parâmetros:**
 - "height": 200: Altura do recorte em pixels.
 - "width": 200: Largura do recorte em pixels.
- **Utilização:** Essa transformação simula o foco em diferentes partes da imagem, ajudando o modelo a se concentrar em áreas variadas de interesse e aumentando a diversidade do dataset.

6. ShiftScaleRotate

- **Descrição:** Realiza transformações de deslocamento, escala e rotação na imagem.
- **Parâmetros:**
 - "p": 0.5: Probabilidade de aplicação do efeito, ou seja, é aplicado em 50% das vezes.
- **Utilização:** Esse efeito é útil para criar variações na posição e orientação dos objetos, o que contribui para a robustez do modelo ao lidar com mudanças na perspectiva da imagem.

7. Blur

- **Descrição:** Aplica um efeito de desfoque na imagem.
- **Parâmetros:**
 - "blur_limit": 3: Define o limite máximo de intensidade do desfoque, com valores mais altos criando um efeito de desfoque mais forte.

Utilização: Esse efeito ajuda a simular imagens com leves distorções, como as causadas por câmeras em movimento, tornando o modelo mais resiliente a pequenas imprecisões visuais.

3.2 Conclusão

Essas transformações foram escolhidas para diversificar o dataset de imagens com o objetivo de treinar modelos de visão computacional de forma mais robusta e eficaz, cobrindo variações de cor, luz, posição e nitidez.

4. TREINAMENTO DA INTELIGÊNCIA ARTIFICIAL

4.1 Modelo de animais

Descrição do Modelo: Modelo utilizado para realizar detecções de animais com menor proximidade da rodovia e câmera, mantendo um ângulo panorâmico dos cenários.

Origem dos Dados: Câmeras posicionadas ao longo da Fernão Dias e Via Paulista, BR-381-SP, KM-931-663S, KM-043-650N; SP-330, KM-312-200N; SP-334, KM-319-600N, KM-324-150N, KM-405-800N.

4.2 Modelo de Animais Versão 1.0

Conjunto de Dados

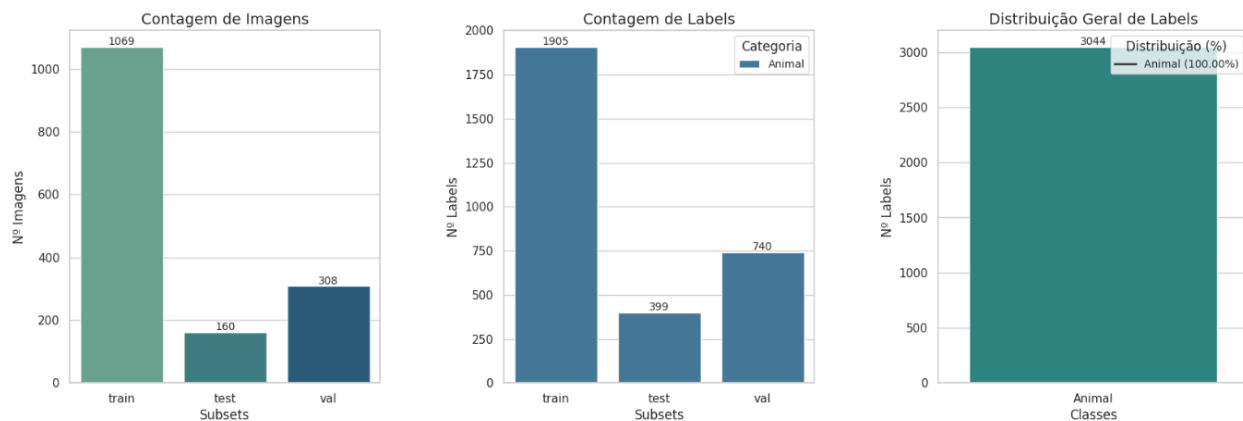
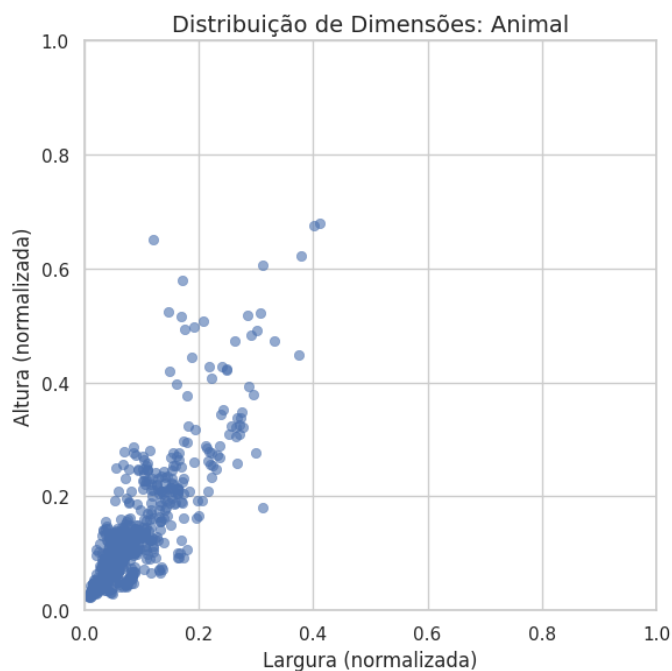


Imagem 2 – Distribuição Dataset Animal v1

- Tamanho do Dataset: Total de 1.537 de imagens, e 3.044 de labels.
- Distribuição dos Dados:
 - Train: 1.069 imagens (Animal: 1.905 labels)
 - Test: 160 imagens (Animal: 399 labels)
 - Val: 308 imagens (Animal: 740 labels)
- Data Augmentation: Horizontal Flip, RandomBrightnessContrast, HueSaturationValue e ShiftScaleRotate.



Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'Animal': As bounding boxes para a classe "Animal" estão concentradas em dimensões menores, especialmente na faixa de 0.0 a 0.2 para largura e altura.
- Há dispersões para valores maiores de altura (até 0.6), sugerindo a presença de algumas imagens com objetos maiores.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

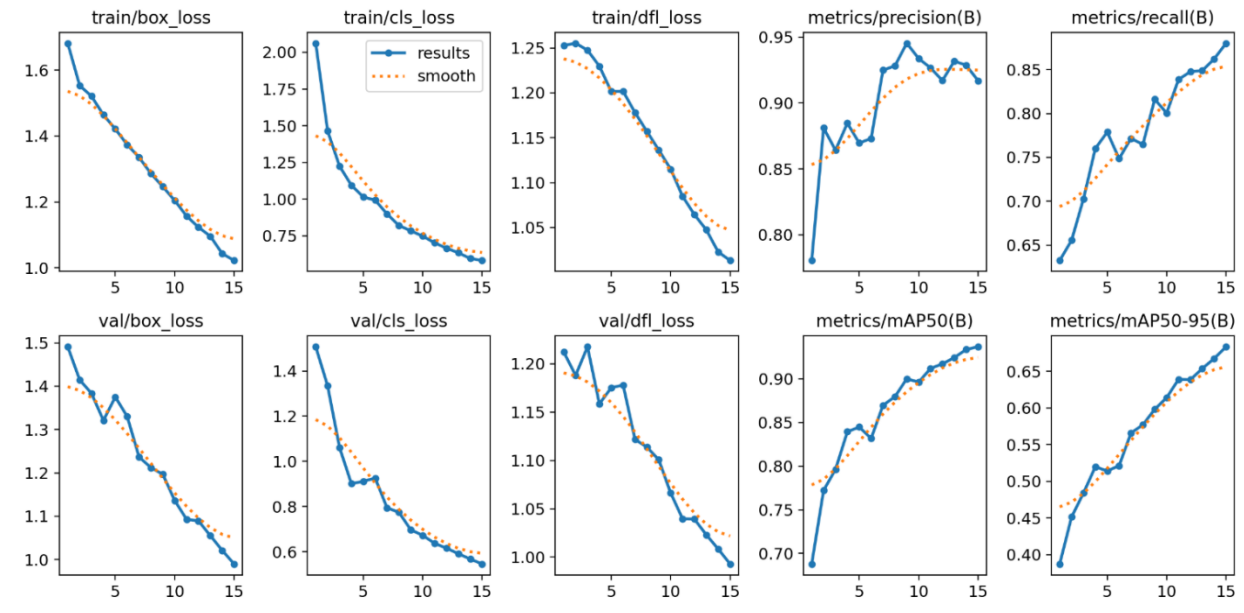


Imagem 4 – Métricas Gerais Modelo de Animais v1

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.90**, demonstrando uma taxa moderada de falsos positivos.
 - O recall cresce até **0.85**, mostrando que a maioria dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.92**, destacando um bom desempenho em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.65**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

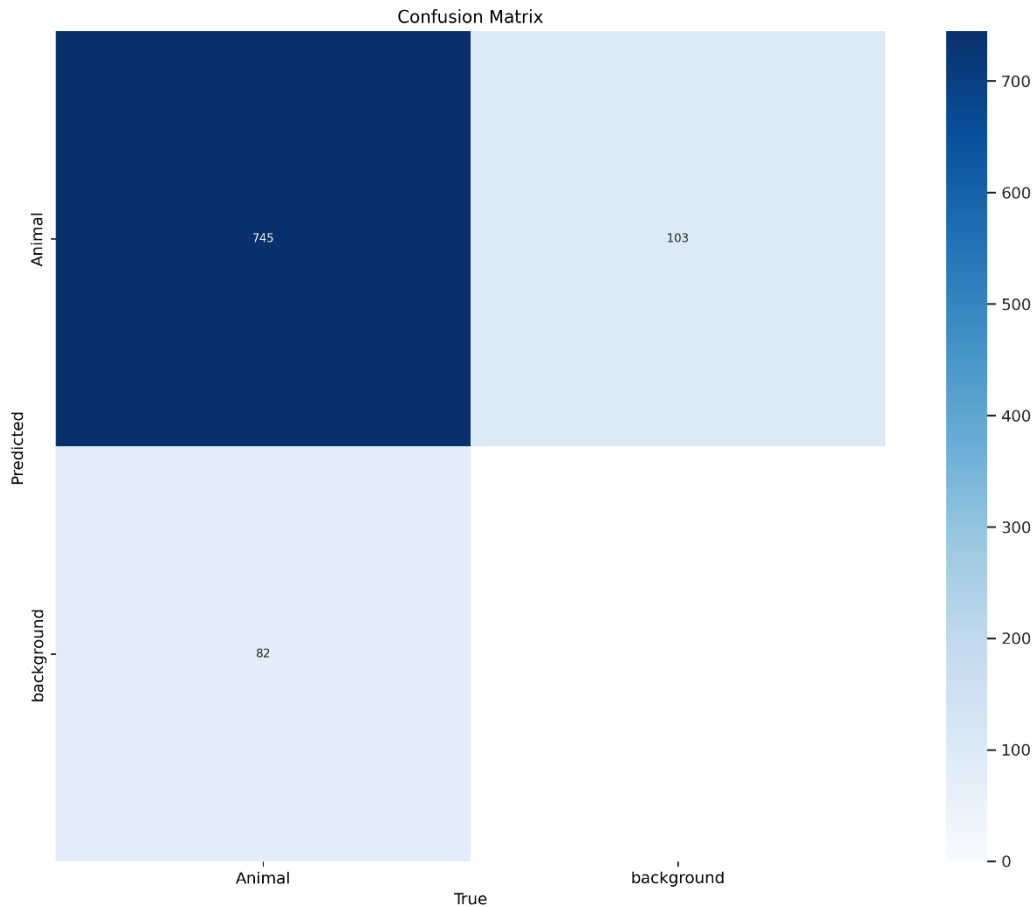


Imagem 5 – Matriz da Confusão Modelo de Animais v1

Resultados:

- Classe "Animal":
 - **TP:** 745 detecções corretas.
 - **FN:** 103 casos não detectados corretamente como "Animal".
- Classe "Background":
 - **FP:** 82 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a classe "Animal", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessa classe, com **745 detecções corretas**.
- A precisão é ligeiramente inferior, o que sugere um número moderado de falsos positivos (**FP**), especialmente na classe "Background", onde **82 erros de classificação** ocorreram.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na redução de falsos positivos (**FP**), especialmente para a classe "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados

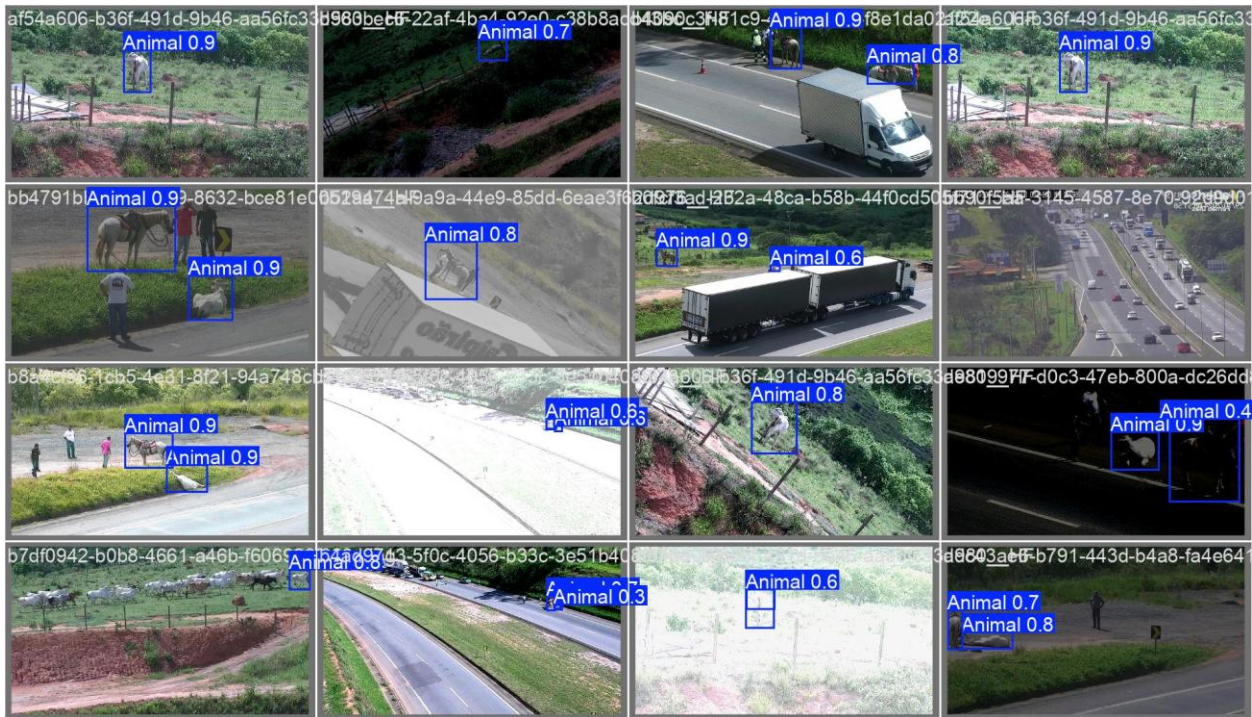


Imagem 6 – Resultados Iniciais Modelo de Animais v1

4.3 Modelo de Contagem

Descrição do Modelo: Modelo utilizado para realizar detecções com maior proximidade da rodovia e das câmeras para contagem e análise de velocidade média dos veículos.

Origem dos Dados: Câmeras posicionadas ao longo da Litoral Sul BR-101-SC, KM-043-100S, KM-044-800N, KM-137-700S, KM-137-800N, KM-206-700N, KM-207-300S, KM-233-000N, KM-233-700S. E na Régis Bittencourt, BR-116-SP KM-288-800S. (Falta KM do Contorno de Florianópolis)

4.4 Modelo de Contagem Versão 1.0

Conjunto de Dados

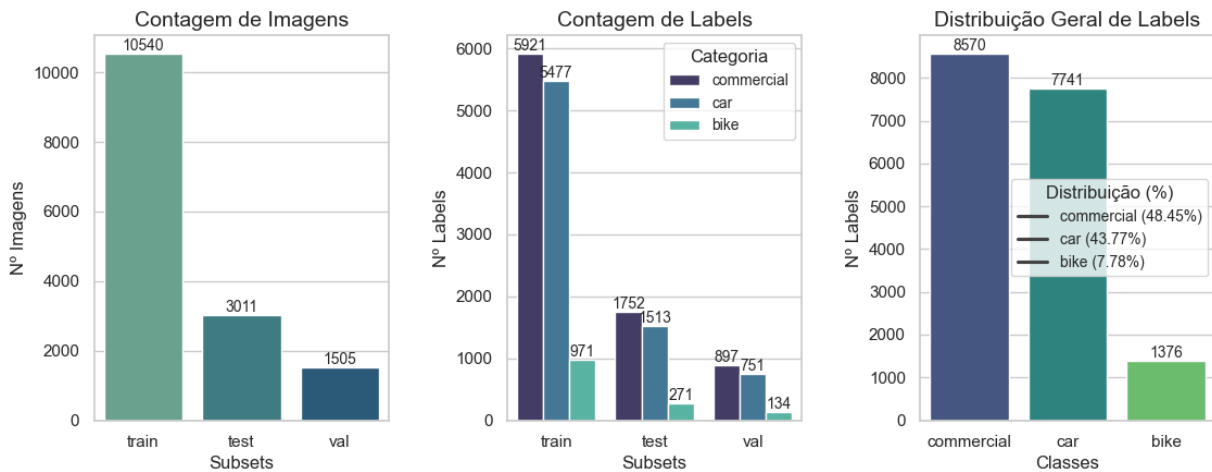


Imagem 7 – Distribuição Dataset Contagem v1

- Tamanho do Dataset: Total de 15.056 de imagens, e 17.687 de labels.
- Distribuição dos Dados:
 - Train: 10.540 imagens (commercial: 5.921 labels, car: 5.477 labels, bike: 971 labels)
 - Test: 3.011 imagens (commercial: 1.752 labels, car: 1.513 labels, bike: 271 labels)
 - Val: 1.505 imagens (commercial: 897 labels, car: 751 labels, bike: 134 labels)
- Data Augmentation: Nenhum.

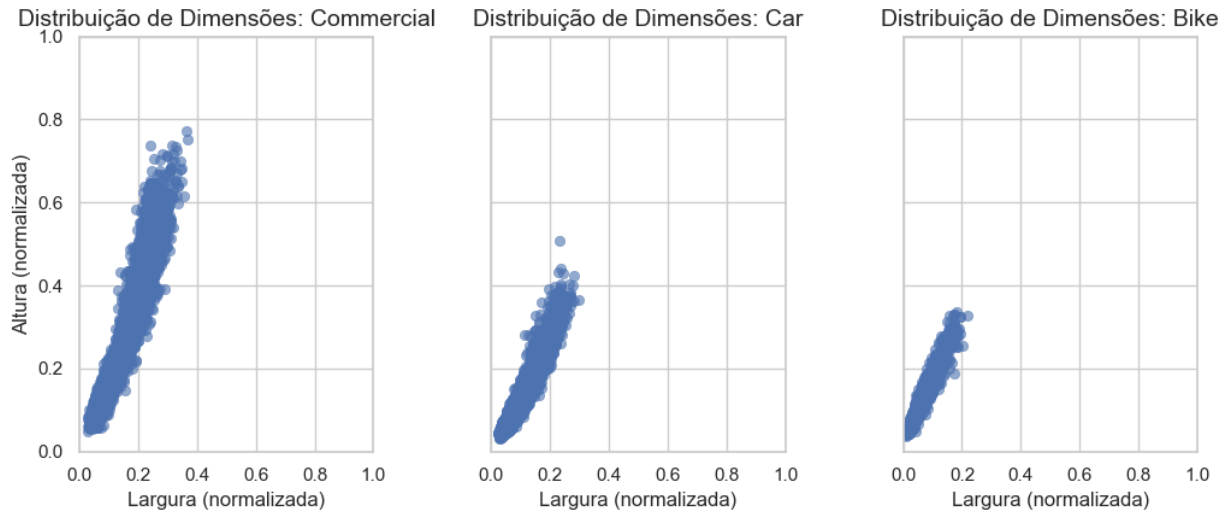


Imagem 8 – Distribuição Bouding Boxes Modelo de Contagem v1

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas, sugerindo que os objetos dessa classe ocupam uma área de eixo Y maior em relação as outras classes. Em virtude do tamanho da cabine.
- Distribuição da Classe 'car': As bounding boxes para a classe car são mais largas, com uma ampla variação de tamanhos, indicando que os carros ocupam uma área significativa nas imagens, variando em tamanho e proporção. Conforme vão se aproximando da câmera.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas, nem tão largas quanto os carros, nem tão altas quanto as comerciais.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

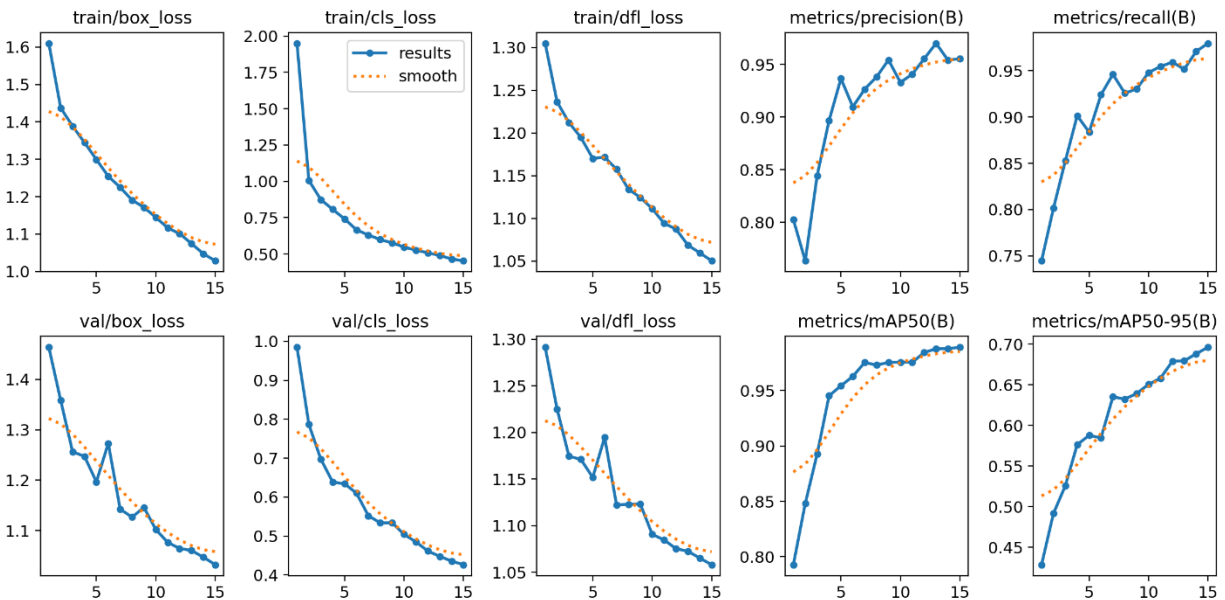


Imagem 9 – Métricas Gerais Modelo de Contagem v1

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.

- Precisão e Recall:

- A precisão atinge cerca de **0.96**, demonstrando baixa taxa de falsos positivos.
- O recall cresce até **0.97**, mostrando que a maioria dos objetos foi corretamente detectada.

- mAP:

- O **mAP@50** chega a aproximadamente **0.97**, destacando bom desempenho em detecções mais amplas.
- O **mAP@50-95** alcança cerca de **0.70**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

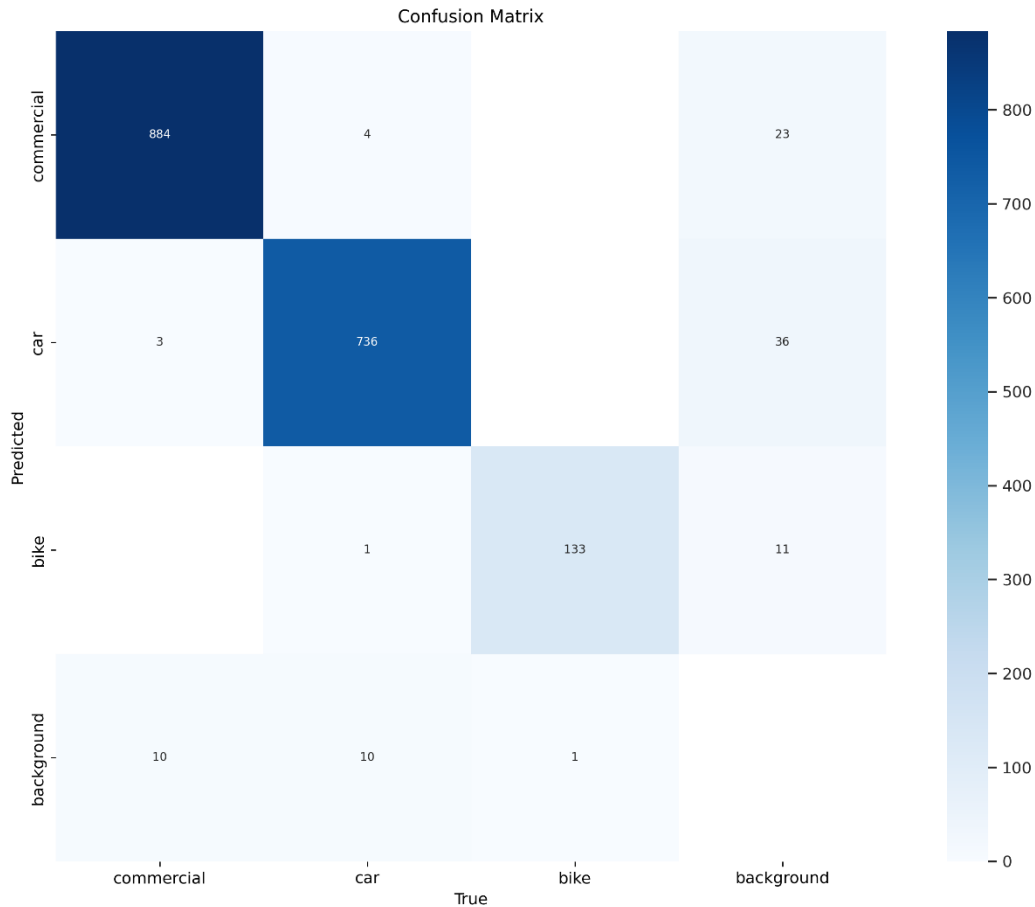


Imagem 10 – Matriz da Confusão Modelo de Contagem v1

Resultados:

- Classe "Commercial":
 - **TP:** 884 detecções corretas.
 - **FN:** 27 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 736 detecções corretas.
 - **FN:** 39 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 133 detecções corretas.
 - **FN:** 12 casos não detectados corretamente como " Bike".
- Classe "Background":

- **FP:** 21 casos foram incorretamente classificados entre os veículos.
- **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para a classe "Commercial", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para a classe "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados



Imagem 11 – Resultados Iniciais Modelo de Contagem v1

4.5 Modelo de Eventos

Descrição do Modelo: Modelo utilizado para realizar detecções de veículos com menor proximidade da rodovia e das câmeras, mantendo um ângulo panorâmico dos cenários a fim de identificar incidentes e comportamentos anormais.

Origem dos Dados: Câmeras posicionadas ao longo da Litoral Sul Litoral Sul BR-101-SC KM-130-700S, BR-101-SC KM-092-800S, BR-101-SC KM-094-700S, BR-101-SC KM-095-900S, BR-101-SC KM-096-800N, BR-116-PR KM-081-000N e BR-376-PR KM-633-000N. E na Régis Bittencourt, BR-116-MG KM-488-600S.

4.6 Modelo de Eventos Versão 1.0

Conjunto de Dados

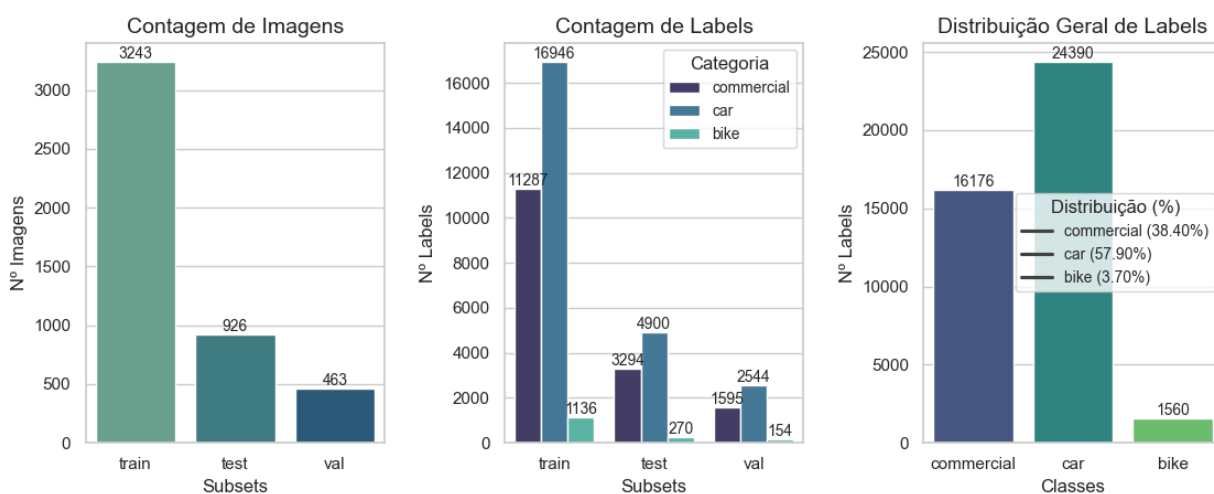


Imagem 12 – Distribuição Dataset Eventos v1

- Tamanho do Dataset: Total de 4.632 de imagens, e 42.126 de labels.
- Distribuição dos Dados:
 - Train: 3.246 imagens (commercial: 11.287 labels, car: 16.946 labels, bike: 1136 labels)
 - Test: 926 imagens (commercial: 3.294 labels, car: 4.900 labels, bike: 270 labels)
 - Val: 463 imagens (commercial: 1.595 labels, car: 2.544 labels, bike: 154 labels)
- Data Augmentation: Horizontal Flip.

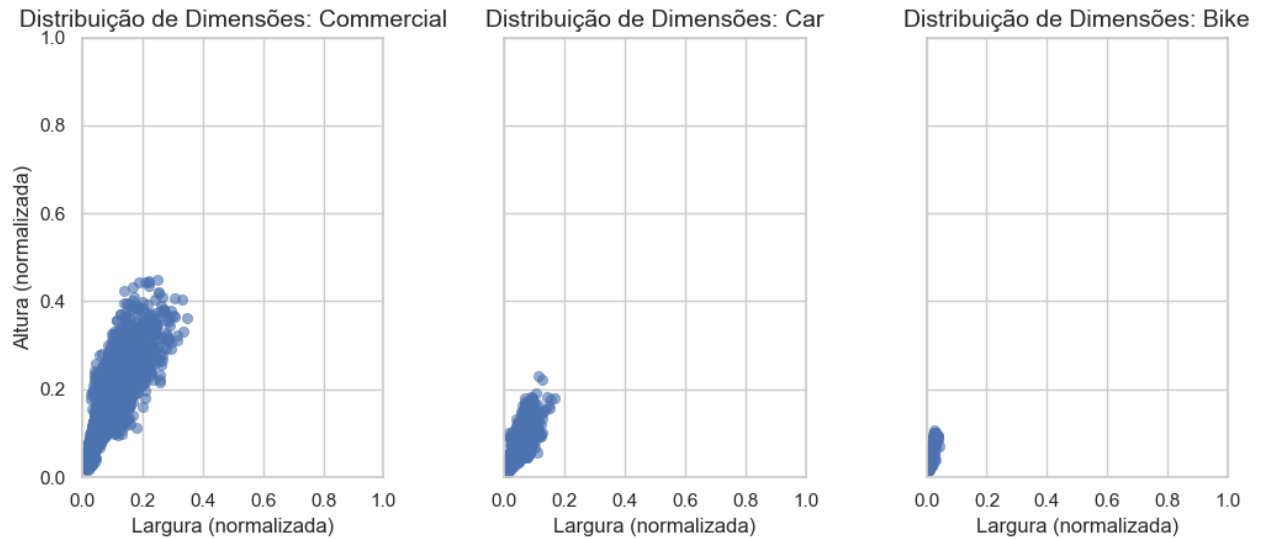


Imagem 13 – Distribuição Bouding Boxes Modelo de Eventos v1

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas e largas, sugerindo que os objetos dessa classe ocupam uma área de eixo X e Y maior em relação as outras classes (pois estamos rotulando o veículo inteiro, não apenas a cabine).
- Distribuição da Classe 'car': As bounding boxes para a classe car possuem uma ampla variação de tamanhos, porém quando submetido sobre uma visão panorâmica, é relativamente difícil distinguir padrões entre suas alturas e larguras.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas e centradas, nem tão grandes quanto os comerciais, nem tão dinâmicas quanto os carros.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

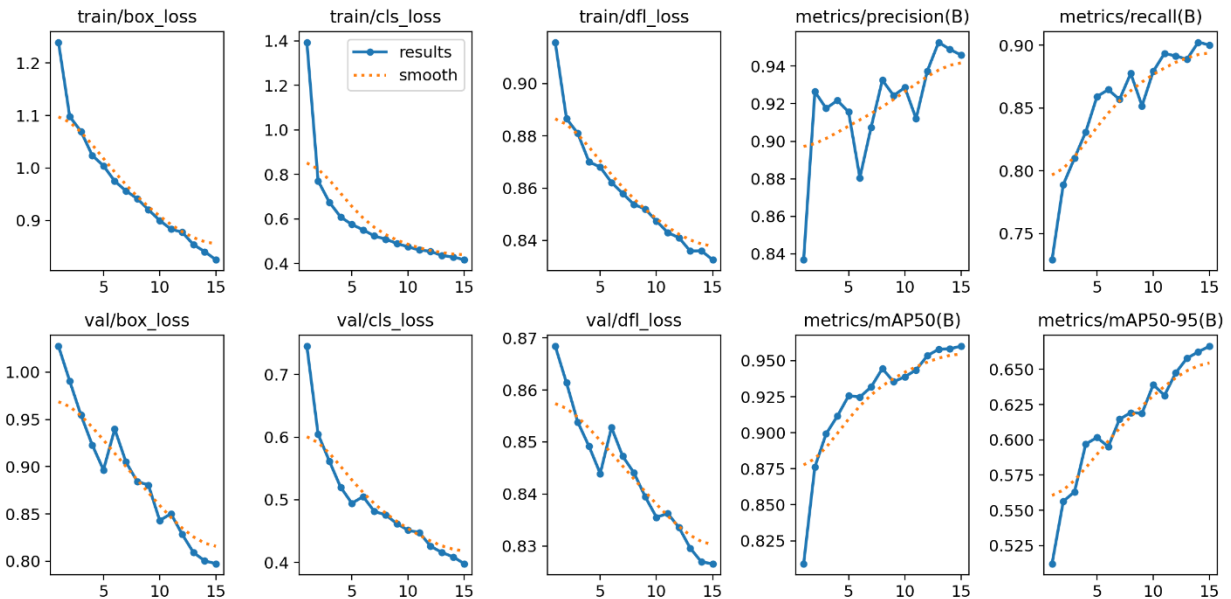


Imagem 14 – Métricas Gerais Modelo de Eventos v1

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.

- Precisão e Recall:

- A precisão atinge cerca de **0.94**, demonstrando baixa taxa de falsos positivos.
- O recall cresce até **0.9**, mostrando que a maioria dos objetos foi corretamente detectada.

- mAP:

- O **mAP@50** chega a aproximadamente **0.96**, destacando bom desempenho em detecções mais amplas.
- O **mAP@50-95** alcança cerca de **0.66**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

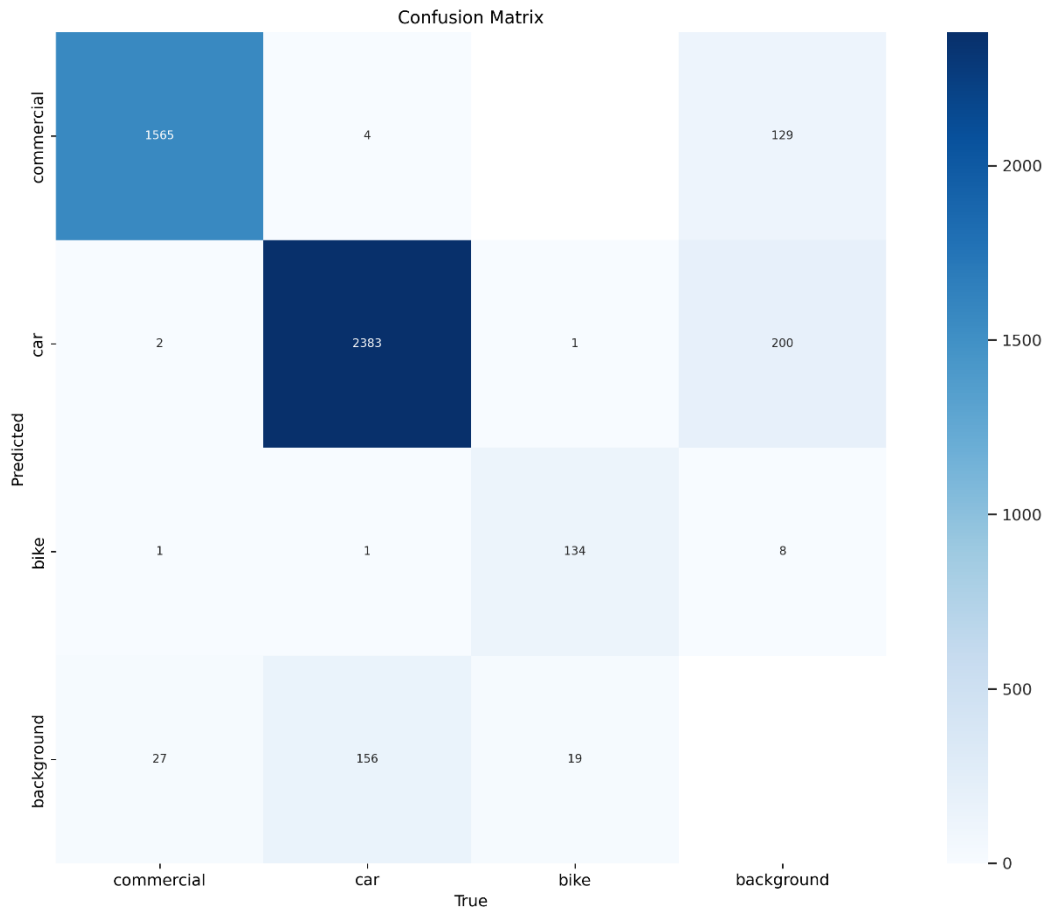


Imagem 15 – Matriz da Confusão Modelo de Eventos v1

Resultados:

- Classe "Commercial":
 - **TP:** 1565 detecções corretas.
 - **FN:** 133 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 2383 detecções corretas.
 - **FN:** 203 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 134 detecções corretas.
 - **FN:** 10 casos não detectados corretamente como " Bike".

- Classe "Background":
 - **FP:** 202 casos foram incorretamente classificados entre os veículos.
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para as classes "Commercial" e "Car", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para as classes "Bike" e "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados



Imagem 16 – Resultados Iniciais Modelo de Eventos v1

4.7 Modelo de Objetos

Descrição do Modelo: Modelo utilizado para realizar detecções de objetos nas rodovias.

Origem dos Dados: Câmeras posicionadas ao longo da Fernão Dias e Via Paulista, BR-101-SC, KM-203-500N, KM-157-500N; SP-255, KM-219-929S, KM-179-488S; SP-330, KM-243-500S, KM-317-000S, KM-314-035N; SP-334, KM-395-240N. Somado a datasets públicos contendo imagens de objetos, destroços, plásticos etc.

4.8 Modelo de Objetos Versão 1.0

Conjunto de Dados

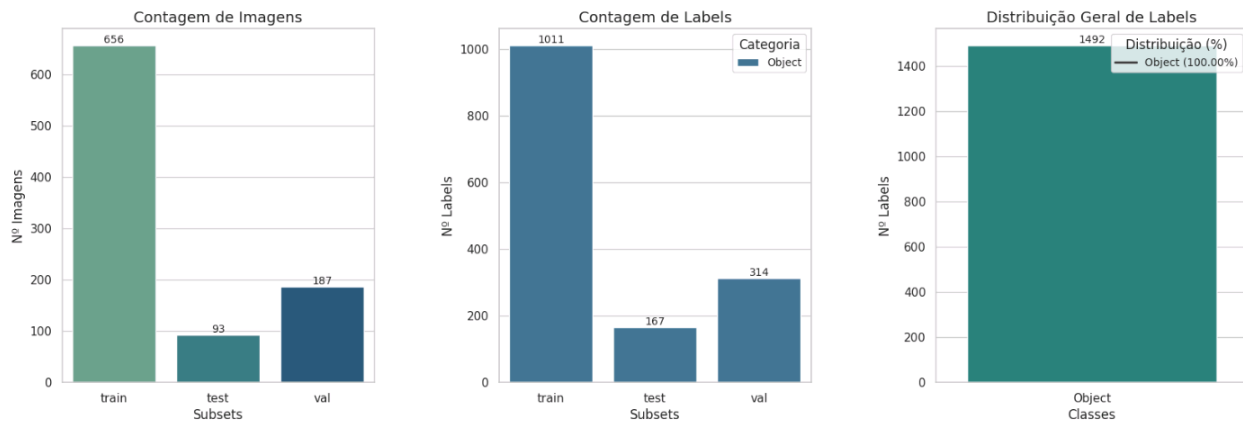


Imagem 17 – Distribuição Dataset Objetos v1

- Tamanho do Dataset: Total de 936 de imagens, e 1.492 de labels.
- Distribuição dos Dados:
 - Train: 656 imagens (Object: 1.011 labels)
 - Test: 93 imagens (Object: 167 labels)
 - Val: 187 imagens (Object: 314 labels)
- Data Augmentation: Nenhuma técnica foi aplicada neste dataset.

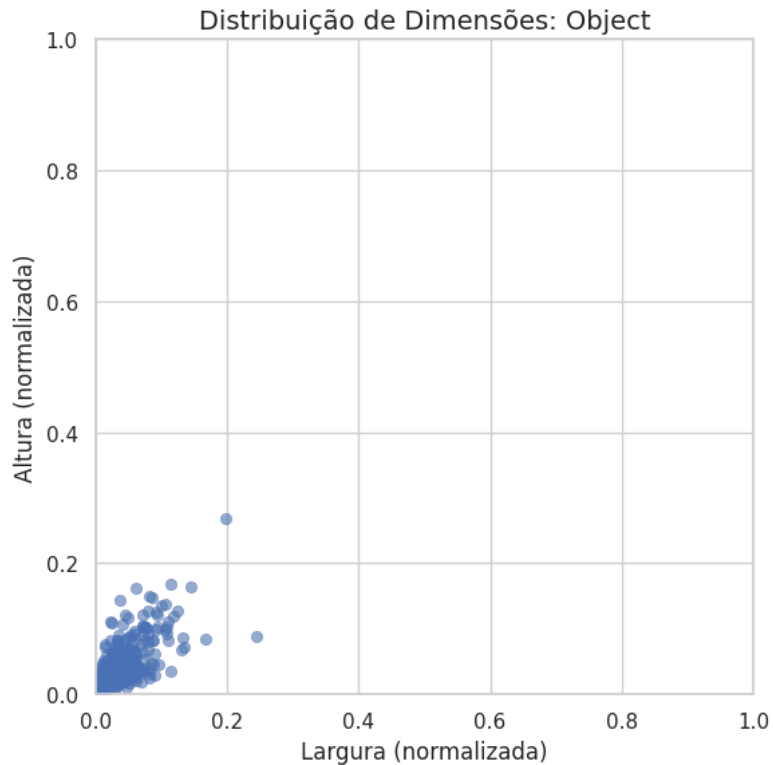


Imagem 18 – Distribuição Bouding Boxes Objetos v1

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'Object': As bounding boxes da classe "Object" estão altamente concentradas em dimensões pequenas, especialmente na faixa de 0.0 a 0.2 para largura e altura.
- Há alguns pontos dispersos para valores um pouco maiores de altura e largura (até cerca de 0.25), mas a maioria das detecções está agrupada em dimensões menores.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 4, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

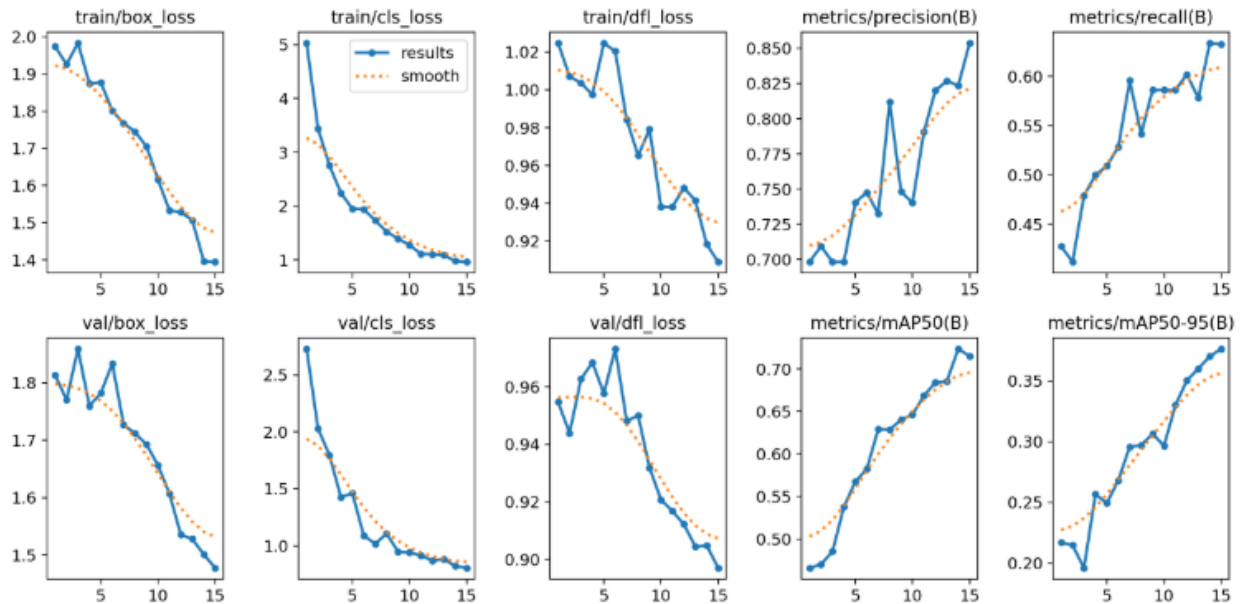


Imagem 19 – Métricas Gerais Detector de Objetos v1

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando uma boa convergência do modelo e redução do erro ao longo das épocas.

- Precisão e Recall:

- A precisão atinge aproximadamente **0.85**, demonstrando um bom controle de falsos positivos.
- O recall cresce até **0.60**, indicando que o modelo consegue detectar corretamente cerca de 60% dos objetos presentes nas imagens.

- mAP:

- O **mAP@50** atinge **0.70**, mostrando um desempenho satisfatório para detecções mais amplas.
- O **mAP@50-95** alcança aproximadamente **0.35**, sugerindo que ainda há margem para melhorias quando se trata de detecções mais rigorosas.

Matriz da confusão:

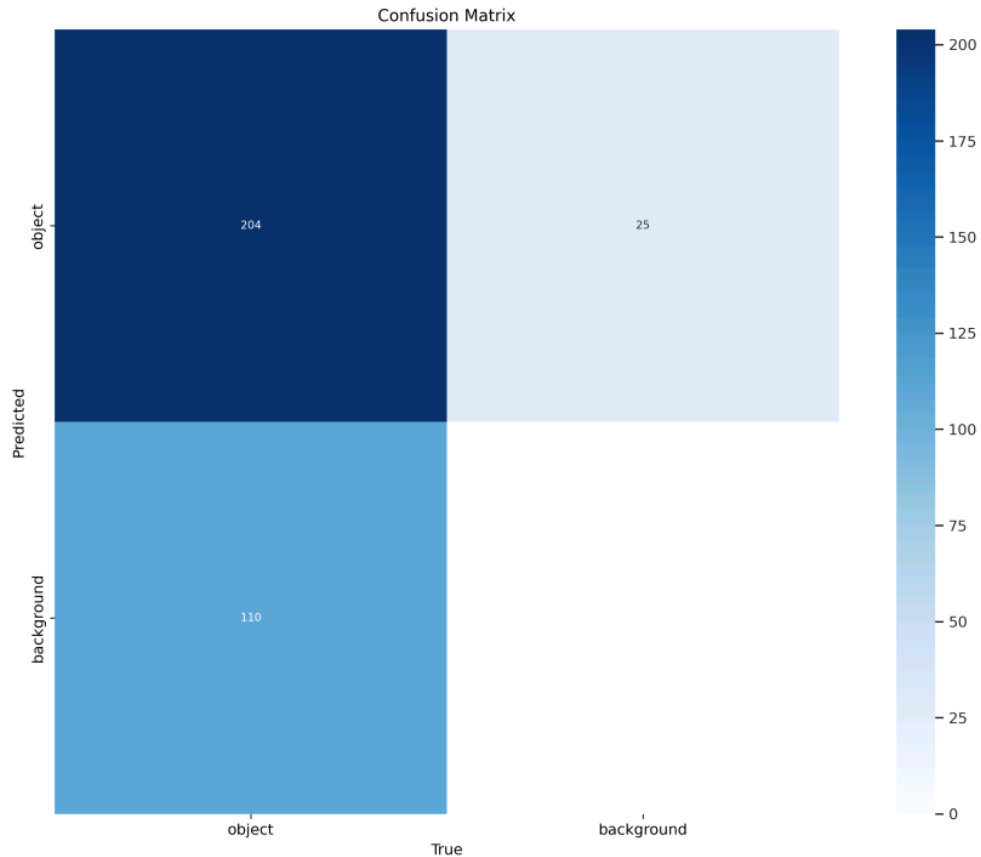


Imagem 20 – Matriz da Confusão Detector de Objetos v1

Resultados:

- Classe "Object":
 - **TP:** 204 detecções corretas.
 - **FN:** 110 casos não detectados corretamente como "Object".
- Classe "Background":
 - **FP:** 25 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade moderada** (recall) para a classe "Object", conseguindo identificar corretamente 204 casos, mas falhando em 110.

- A **precisão também é moderada**, já que ainda há falsos positivos (25), o que pode indicar que o modelo às vezes classifica erroneamente objetos como fundo.
- O **desempenho geral** do modelo é **razoável**, mas pode ser melhorado ajustando hiperparâmetros, refinando a estratégia de balanceamento de classes e aumentando a diversidade do dataset para reduzir falsos positivos e falsos negativos.

Visualização dos Resultados



Imagem 21 – Resultados Iniciais Modelo de Objetos v1

4.9 Modelo de Incêndio

Descrição do Modelo: Modelo utilizado para realizar detecções de incêndio com maior proximidade da rodovia e câmera.

Origem dos Dados: Câmeras posicionadas ao longo da Fernão Dias e Via Paulista, KM-857-000N, KM-857-000S, KM-073-000N, KM-077-630S, KM-086-000N, KM-483-000S, KM-060-000N, KM-517-110S, KM-056-190N. Somado a datasets públicos contendo imagens de incêndios, fogos, fumaças etc.

4.10 Modelo de Incêndio Versão 1.0

Conjunto de Dados

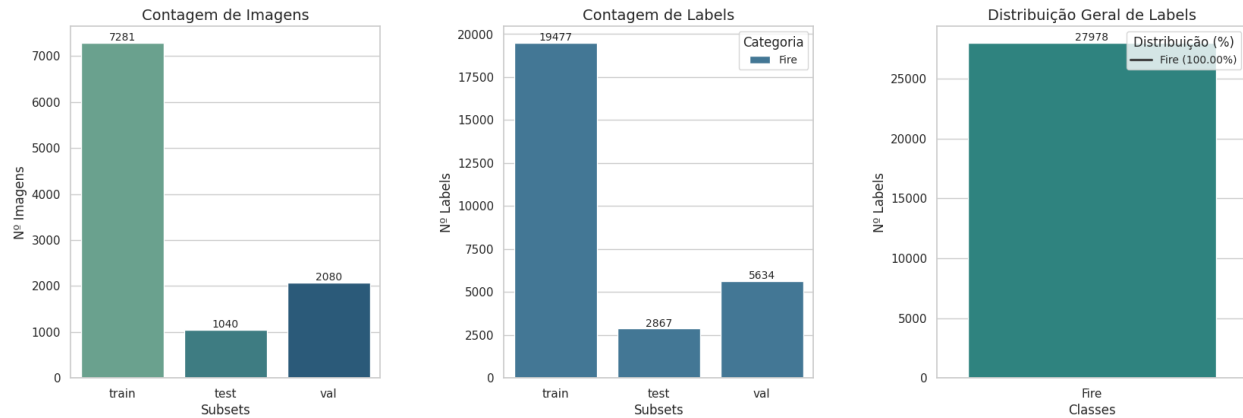


Imagem 22 – Distribuição Dataset Incêndio v1

- Tamanho do Dataset: Total de 10.401 de imagens, e 27.978 de labels.
- Distribuição dos Dados:
 - Train: 7.281 imagens (Fire: 19.477 labels)
 - Test: 1.040 imagens (Fire: 2.867 labels)
 - Val: 2.080 imagens (Fire: 5.634 labels)
- Data Augmentation: Nenhuma técnica foi aplicada neste dataset.

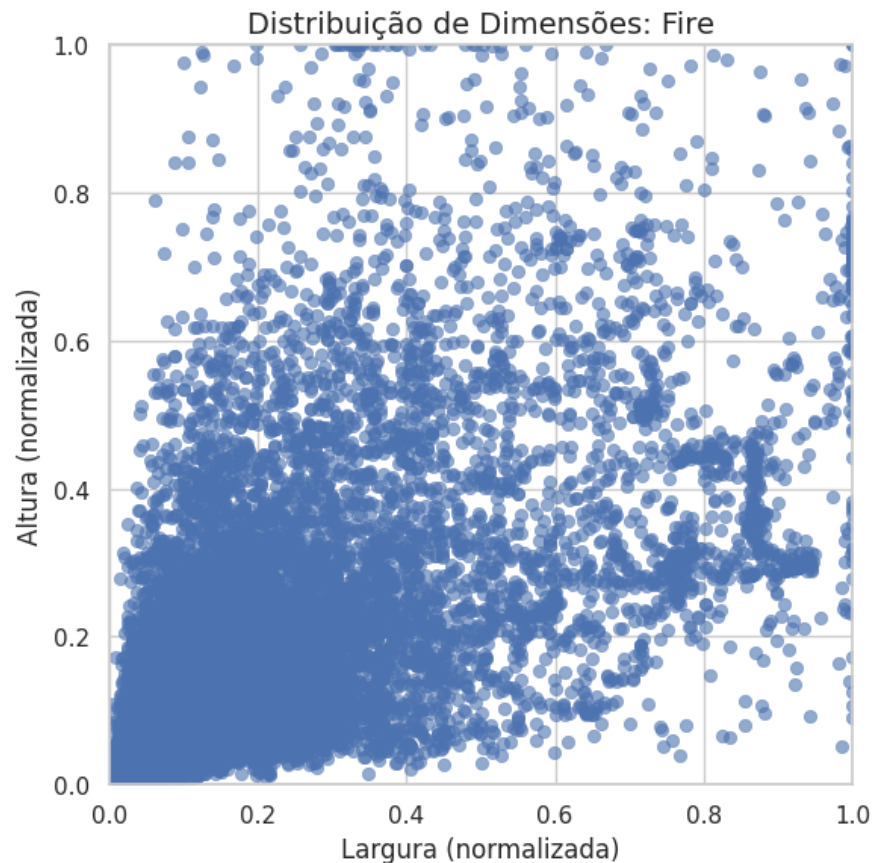


Imagem 23 – Distribuição Bouding Boxes Incêndio v1

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'Fire': As bounding boxes para a classe "Fire" estão concentradas em dimensões maiores, especialmente na faixa de 0.0 a 0.6 para largura e altura.
- Há dispersões para valores maiores de altura (até 0.6), sugerindo a presença de algumas imagens com objetos maiores.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 20, Batch: 8, Image Size: 640.
- Hardware:

- Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

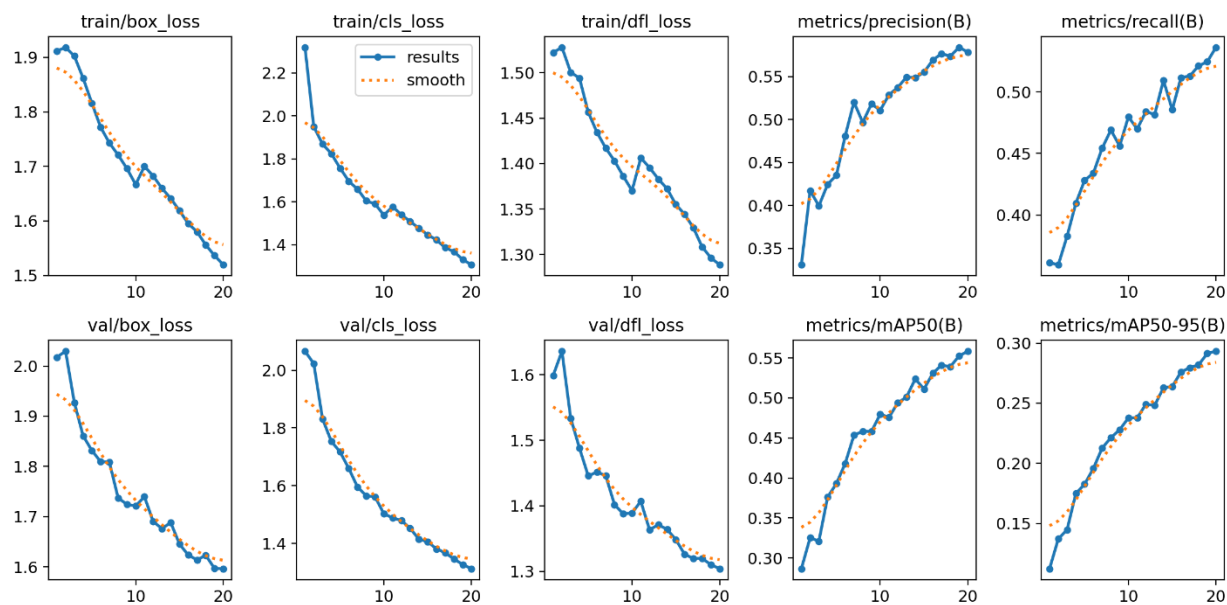


Imagem 24 – Métricas Gerais Detector de Incêndios v1

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.55**, demonstrando uma taxa moderada de falsos positivos.
 - O recall cresce até **0.50**, mostrando que a metade dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.55**, destacando um desempenho mediano em detecções mais amplas.

- O **mAP@50-95** alcança cerca de **0.30**, sugerindo margem para melhorias em detecções mais rigorosas.

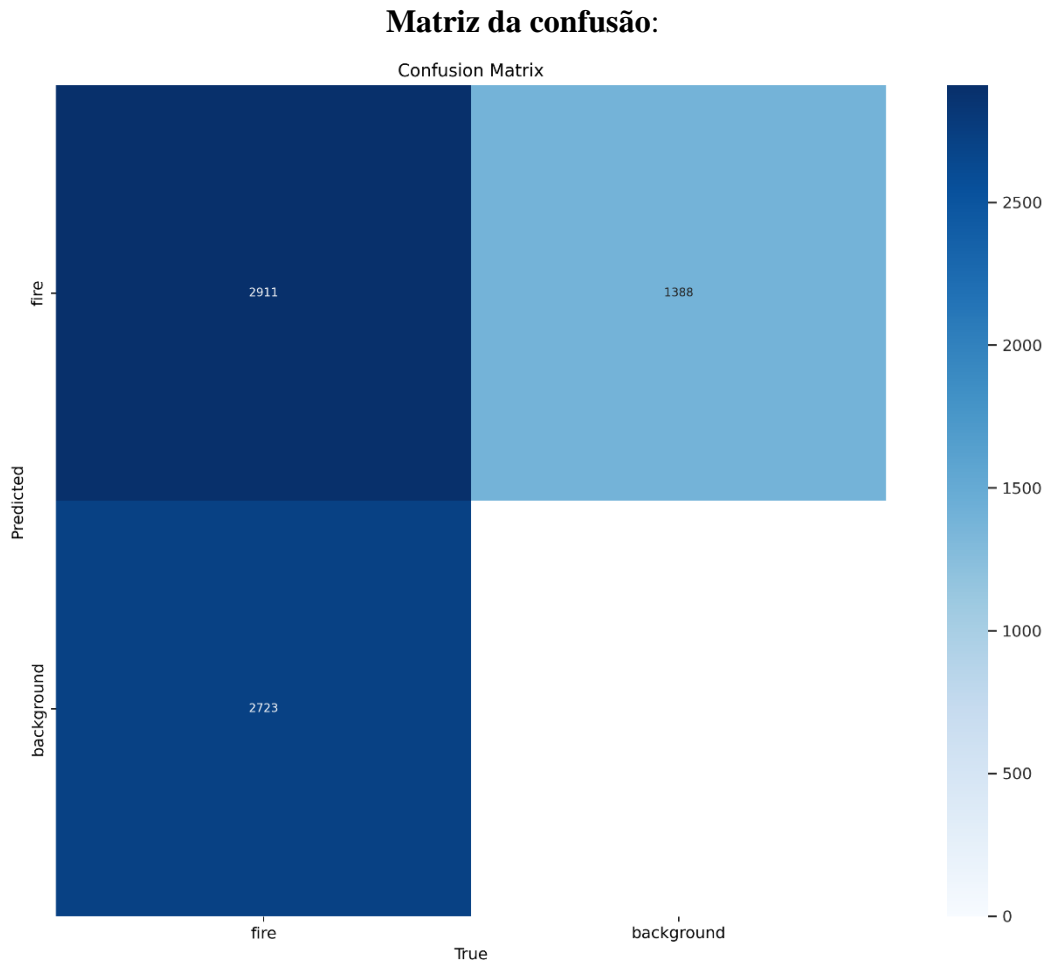


Imagem 25 – Matriz da Confusão Detector de Incêndios v1

Resultados:

- Classe "Fire":
 - **TP:** 2911 detecções corretas.
 - **FN:** 1388 casos não detectados corretamente como "Fire".
- Classe "Background":
 - **FP:** 2723 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade mediana (recall)** para a classe "Fire", o que significa que ele consegue identificar corretamente boa parte dos casos dentro dessa classe, com **2911 detecções corretas e 1388 detecções incorretas**.
- A precisão é mediana, o que sugere um número alto de falsos positivos (**FP**), especialmente na classe "Background", onde **2723 erros de classificação** ocorreram.
- O desempenho geral do modelo é moderado, com um recall de aproximadamente 50% para a classe "Fire". Para melhorar a precisão e reduzir falsos positivos na classe "Background", recomenda-se a aplicação de técnicas como balanceamento de classes, uso de datasets mais diversificados, ou ajustes nos thresholds de detecção.

Visualização dos Resultados

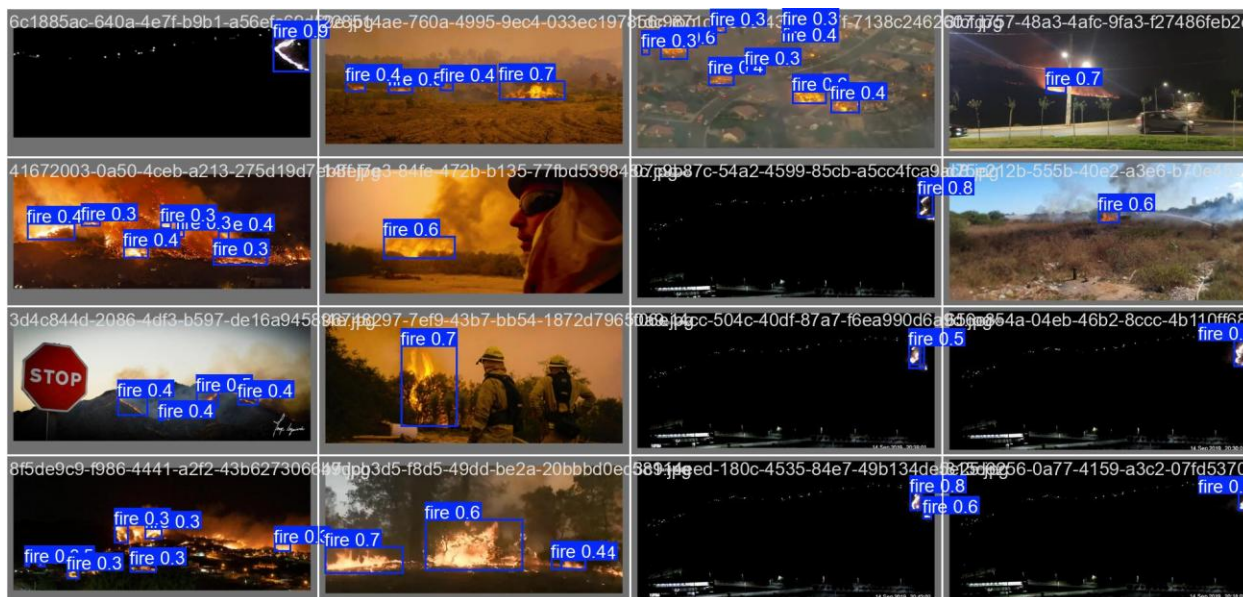


Imagem 26 – Resultados Iniciais Modelo de Incêndios v1

4.11 Modelo de Pessoas

Descrição do Modelo: Modelo utilizado para realizar detecções de ciclistas e pedestres nas rodovias e câmeras.

Origem dos Dados: Câmeras posicionadas ao longo da Fernão Dias e Via Paulista, KM-071-100S, KM-802-620N, KM-919-150N, KM-041-100S, KM-683-700N, KM-907-100N, KM-729-800N.

4.12 Modelo de Pessoas Versão 1.0

Conjunto de Dados

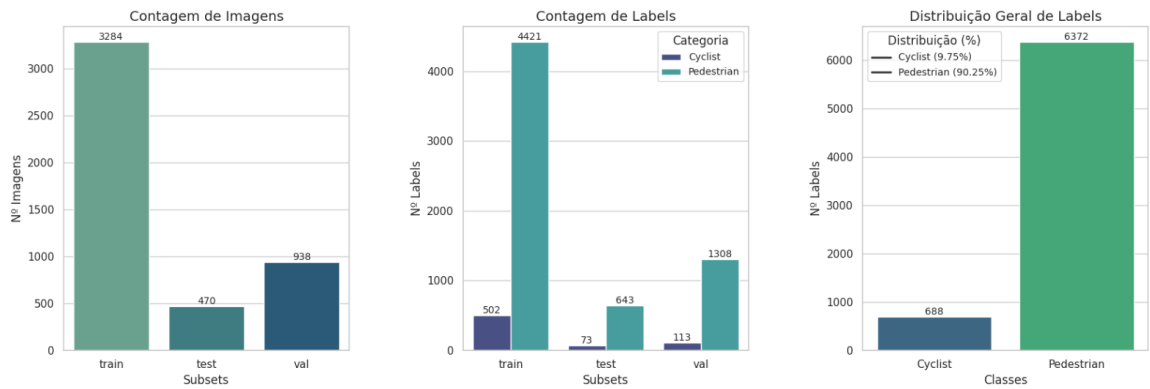


Imagem 27 – Distribuição Dataset Pessoas v1

- Tamanho do Dataset: Total de 4.692 de imagens, e 7.060 de labels.
- Distribuição dos Dados:
 - Train: 3.284 imagens (Cyclist: 502 labels, Pedestrian: 4421 labels)
 - Test: 470 imagens (Cyclist: 73 labels, Pedestrian: 643 labels)
 - Val: 938 imagens (Cyclist: 113 labels, Pedestrian: 1308 labels)
- Data Augmentation: Horizontal Flip, RandomBrightnessContrast, HueSaturationValue e ShiftScaleRotate.

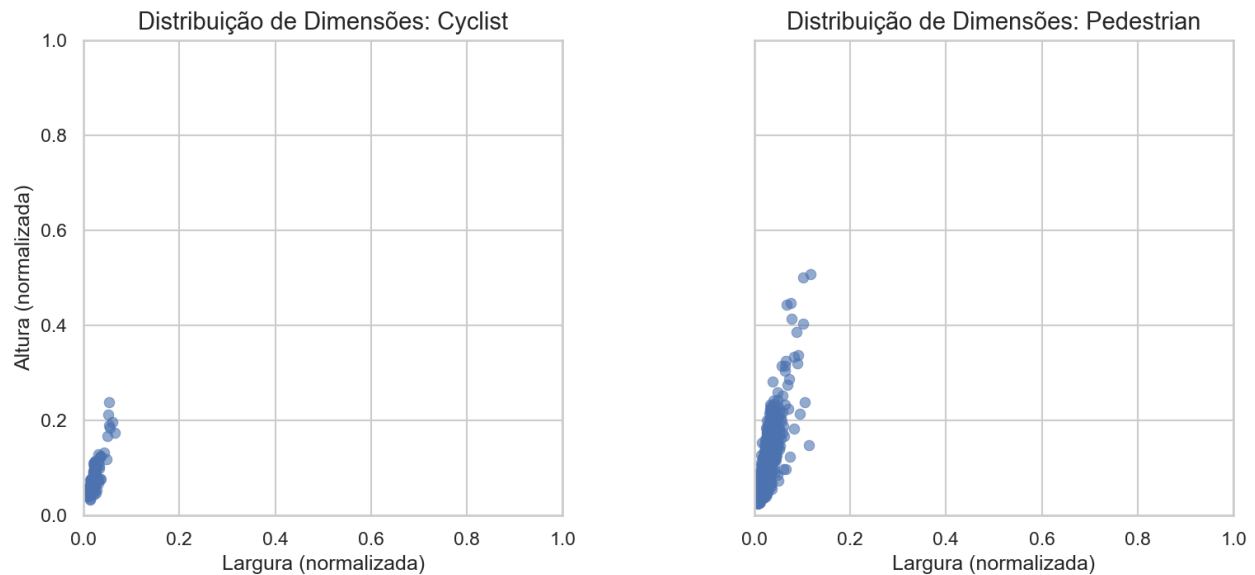


Imagem 28 – Distribuição Bounding Boxes Pessoas v1

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- As bounding boxes das classes "Pedestrian" e "Cyclist" possuem dimensões reduzidas, concentradas entre 0.0 e 0.2 para largura e altura. Isso sugere que os objetos detectados são pequenos e semelhantes, com a classe "Cyclist" apresentando menos amostras no dataset.
- Algumas dispersões chegam a 0.4, mas a maioria dos objetos mantém dimensões pequenas e padronizadas.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

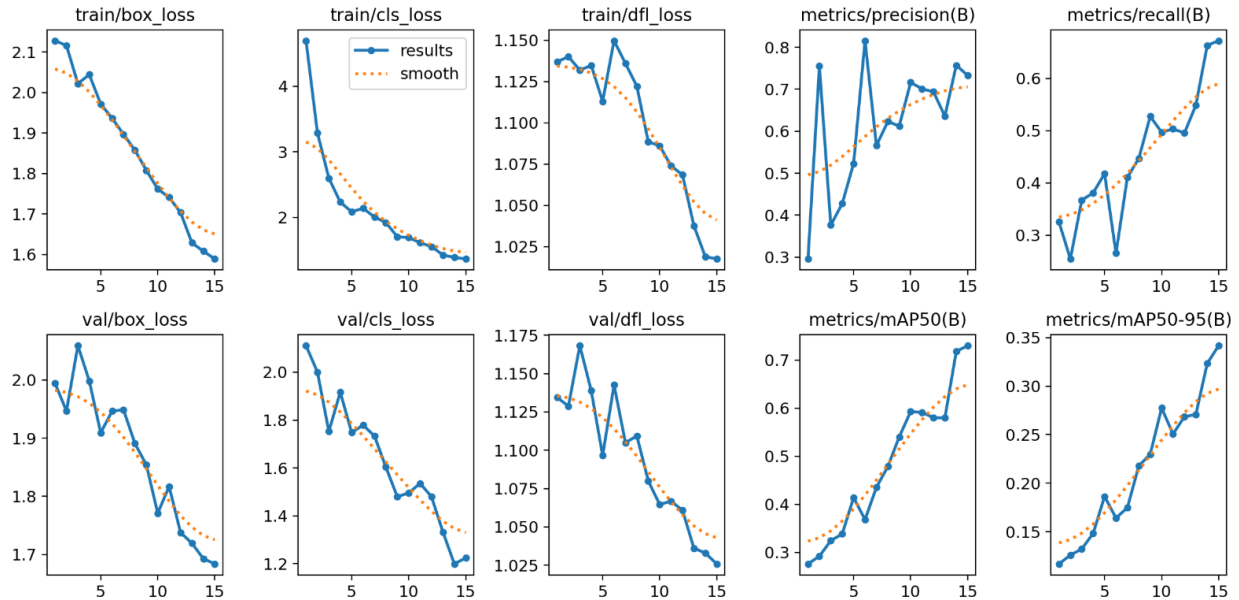


Imagem 29 – Métricas Gerais Modelo de Pessoas v1

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente ao longo das épocas tanto no treinamento quanto na validação. Isso indica uma boa convergência do modelo, com aprendizado progressivo e estabilização dos pesos.
- Precisão e Recall:
 - A precisão cresce até aproximadamente **0.75**, o que indica uma taxa moderada de falsos positivos.
 - O recall aumenta até **0.65**, mostrando que o modelo está conseguindo identificar a maioria dos objetos corretamente.
- mAP:
 - O **mAP@50** atinge cerca de **0.70**, destacando um desempenho razoável em detecções mais amplas.
 - O **mAP@50-95** alcança aproximadamente **0.35**, sugerindo que ainda há margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

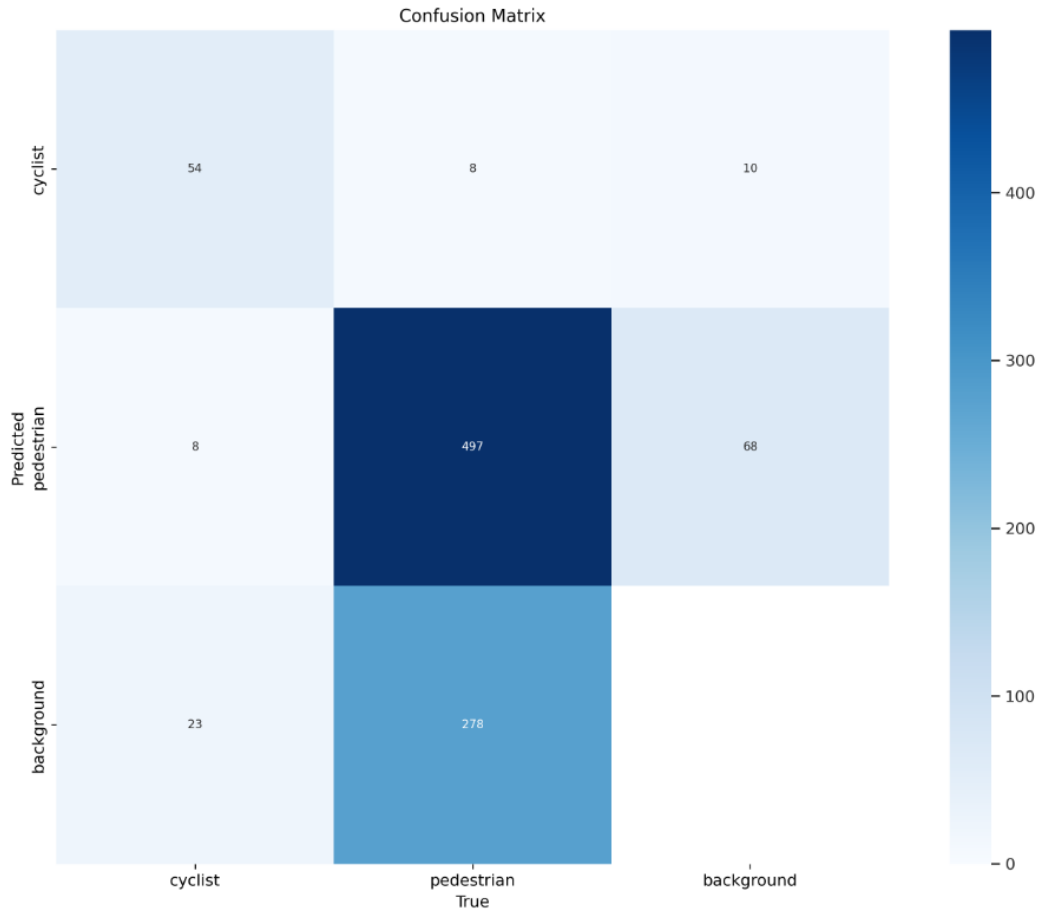


Imagem 30 – Matriz da Confusão Modelo de Pessoas v1

Resultados:

- Classe "Cyclist":
 - **TP:** 54 detecções corretas.
 - **FN:** 18 casos não detectados corretamente (8 classificados como "Pedestrian" e 10 como "Background").
- Classe "Pedestrian":
 - **TP:** 497 detecções corretas.
 - **FN:** 76 casos não detectados corretamente (8 classificados como "Cyclist" e 68 como "Background").
- Classe "Background":
 - **FP:** 23 amostras classificadas incorretamente como "Cyclist".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade moderada (recall de 65%)** para a classe "Pedestrian", o que indica que consegue identificar corretamente a maioria dos pedestres, mas **ainda apresenta alguns erros**.
- A **precisão está acima da média (75%)**, sugerindo que os falsos positivos não são excessivamente altos, mas ainda existem casos em que objetos são **erroneamente classificados**.
- O **desempenho geral** do modelo é **razoável**, com **mAP@50 de 70%** e **mAP@50-95 de 35%**, indicando que as detecções em limiares mais altos ainda podem ser aprimoradas. Será preciso um aumento de diversificação do dataset para reduzir ambiguidades entre classes e ajuste de hiperparâmetros e uso de técnicas como Focal Loss para lidar com o desequilíbrio entre classes.

Visualização dos Resultados



Imagem 31 – Resultados Iniciais Modelo de Pessoas v1

5. APRIMORAMENTO DA APLICAÇÃO DE DETECÇÃO DE EVENTOS

5.1 Modelo de Animais Versão 2.0

Conjunto de Dados (Sem modificações)

O conjunto de dados não sofreu modificações, foi realizado apenas um fine tuning do modelo com o mesmo dataset da versão anterior.

Fine-tuning

O modelo de Detecção de Animais foi inicialmente baseado em uma arquitetura pré-treinada e, posteriormente, passou por uma etapa de ajuste fino (fine-tuning). Essa segunda etapa teve como objetivo aprimorar a performance do modelo, tornando-o mais sensível, preciso e confiável nas detecções de animais em diferentes ambientes e condições visuais.

Ao invés de reiniciar o treinamento do zero, utilizamos um modelo base robusto (YOLOv8 Medium) e aplicamos um novo treinamento com parâmetros cuidadosamente configurados, especialmente voltados para a identificação de animais de diferentes tamanhos.

Abaixo, estão descritos os principais pontos da configuração aplicada e o que cada um representa.

Estratégias de Parada e Otimização

- **patience=10**: Se o desempenho não melhorar após 10 ciclos consecutivos, o treinamento é encerrado automaticamente (early stopping), evitando desperdício computacional.
- **optimizer='AdamW'**: Otimizador avançado que oferece maior estabilidade durante o treinamento, ideal para ajustes mais delicados.
- **lr=0.001**: Taxa de aprendizado inicial ajustada para equilibrar velocidade e estabilidade no ajuste dos pesos.
- **lrf=0.01**: Define a menor taxa de aprendizado ao final do ciclo de treinamento.
- **momentum=0.937**: Valor que ajuda a suavizar os passos do otimizador durante o treinamento.
- **weight_decay=0.0005**: Penalização aplicada a pesos excessivamente altos, ajudando a evitar sobreajuste.
- **warmup_epochs=3**: As 3 primeiras épocas são usadas para aquecer o modelo, iniciando com ajustes mais suaves.
- **label_smoothing=0.1**: Técnica para suavizar os rótulos de classes, reduzindo o impacto de erros esporádicos no treinamento.

Estratégia para Detecção de Objetos Pequenos

- **single_cls=True**: Indica que o modelo irá trabalhar com apenas uma classe (animais), o que reduz a complexidade e foca a aprendizagem em uma única tarefa.
- **iou=0.45**: Ajusta o limiar de Interseção sobre União (IoU), influenciando o critério de acerto entre as caixas preditas e as reais. Um valor mais baixo aumenta o *recall*.
- **conf=0.15**: Define uma confiança mínima mais baixa para detectar objetos pequenos ou parcialmente visíveis (como animais à distância).

Augmentações (Aumentos de Dados) Visuais

- **mosaic=1.0**: Junta quatro imagens em uma só, promovendo mais variações espaciais e melhorando a generalização.
- **mixup=0.2**: Mistura de imagens e rótulos que cria novos exemplos compostos, ajudando o modelo a lidar com sobreposição de objetos.
- **degrees=10**: Rotação leve das imagens, útil para detectar animais em diferentes ângulos.
- **scale=0.5**: Variação de escala para simular diferentes distâncias dos animais na imagem.
- **shear=2.0**: Aplica distorções inclinadas leves, aumentando a capacidade do modelo de identificar animais em posições não convencionais.
- **augment=True**: Ativa as estratégias de aumento de dados durante o treinamento.
- **dropout=0.1**: Técnica que “desativa” partes da rede durante o treinamento para melhorar a capacidade de generalização do modelo.

Configurações de Infraestrutura

- **model = YOLO('yolov8m.pt')**: Utiliza a arquitetura YOLOv8 Medium, que oferece bom equilíbrio entre velocidade e precisão, sendo adequada para cenários mais complexos como a detecção de animais.
- **data='data_animais.yaml'**: Define o conjunto de dados específico com as imagens e anotações da tarefa de detecção de animais.
- **imgsz=960**: Tamanho das imagens definido para aumentar a visibilidade de detalhes e facilitar a detecção de animais pequenos.
- **epochs=45**: O treinamento é realizado ao longo de 50 ciclos, garantindo tempo suficiente para convergência.

- **batch=8**: Define o número de imagens processadas simultaneamente, ajustado para melhor aproveitamento da GPU.
- **device=0**: O treinamento foi executado utilizando a GPU principal disponível.
- **name='Animais'**: Define o nome da pasta onde os resultados e registros deste experimento são salvos.

Considerações Finais da etapa de fine-tuning

Essa etapa de ajuste fino foi fundamental para transformar um modelo base em um modelo altamente especializado na tarefa de detecção de animais. O processo foi cuidadosamente ajustado para melhorar a capacidade de identificar animais de diferentes portes, ângulos e distâncias, preservando a robustez e precisão necessárias para aplicações reais. Estratégias específicas foram adotadas para lidar com a dificuldade adicional que objetos pequenos e parcialmente visíveis representam nesse tipo de tarefa.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 45, Batch: 8, Image Size: 960.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

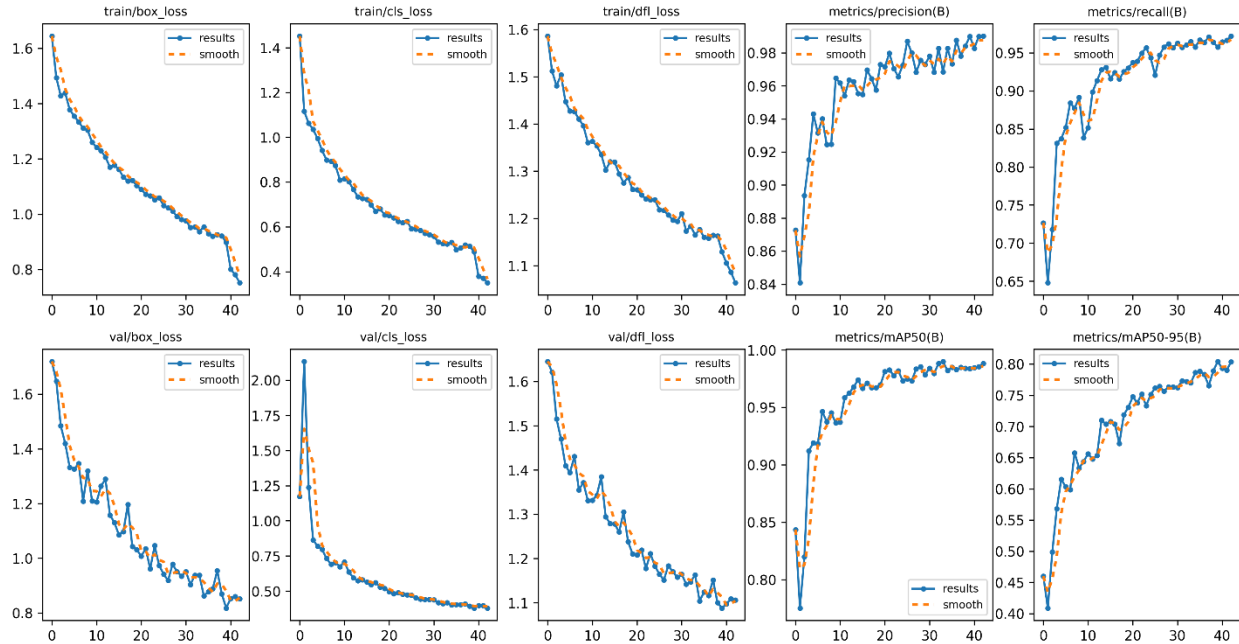


Imagem 32 – Métricas Gerais Modelo de Animais v2

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.98**, demonstrando uma taxa moderada de falsos positivos.
 - O recall cresce até **0.95**, mostrando que a maioria dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.98**, destacando um bom desempenho em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.80**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

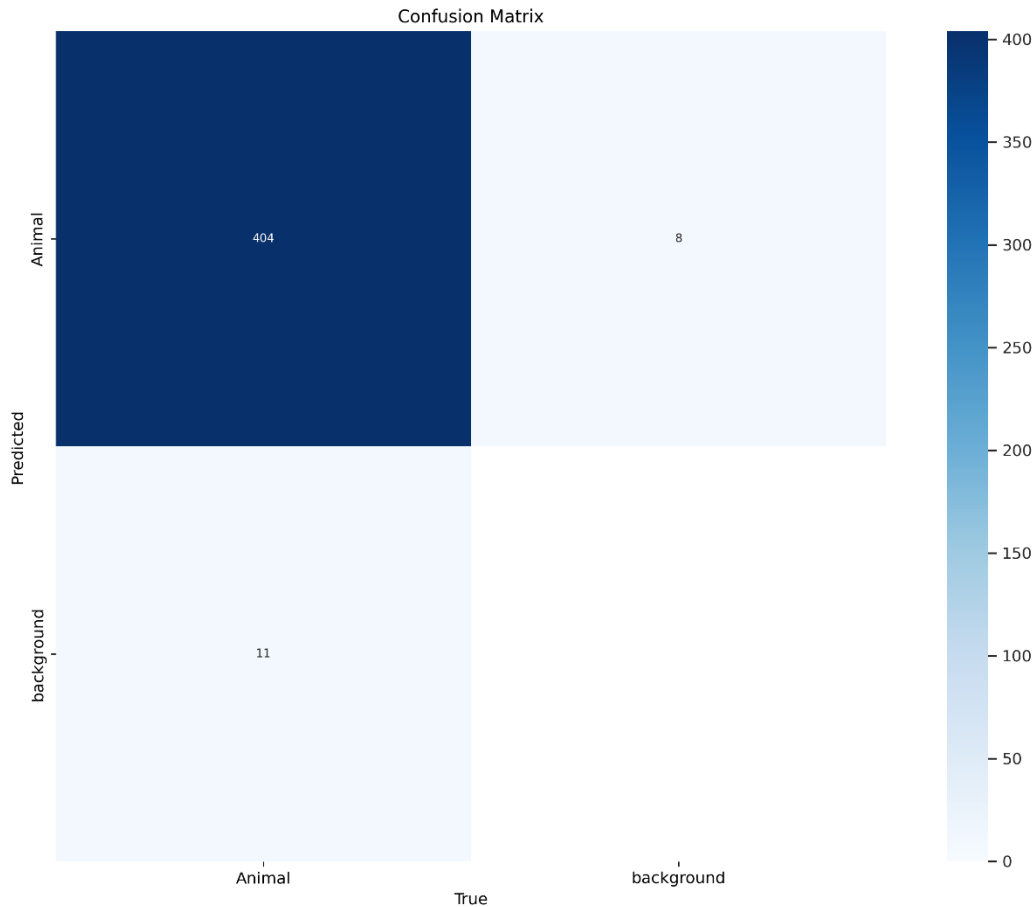


Imagem 33 – Matriz da Confusão Modelo de Animais v2

Resultados:

- Classe "Animal":
 - **TP:** 404 detecções corretas.
 - **FN:** 8 casos não detectados corretamente como "Animal".
- Classe "Background":
 - **FP:** 11 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a classe "Animal", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessa classe, com **404 detecções corretas**.
- A precisão é ligeiramente inferior, o que sugere um número baixo de falsos positivos (**FP**), especialmente na classe "Background", onde **11 erros de classificação** ocorreram.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na redução de falsos positivos (**FP**), especialmente para a classe "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados



Imagem 34 – Resultados Iniciais Modelo de Animais v2

5.2 Modelo de Eventos Versão 2.0

Conjunto de Dados

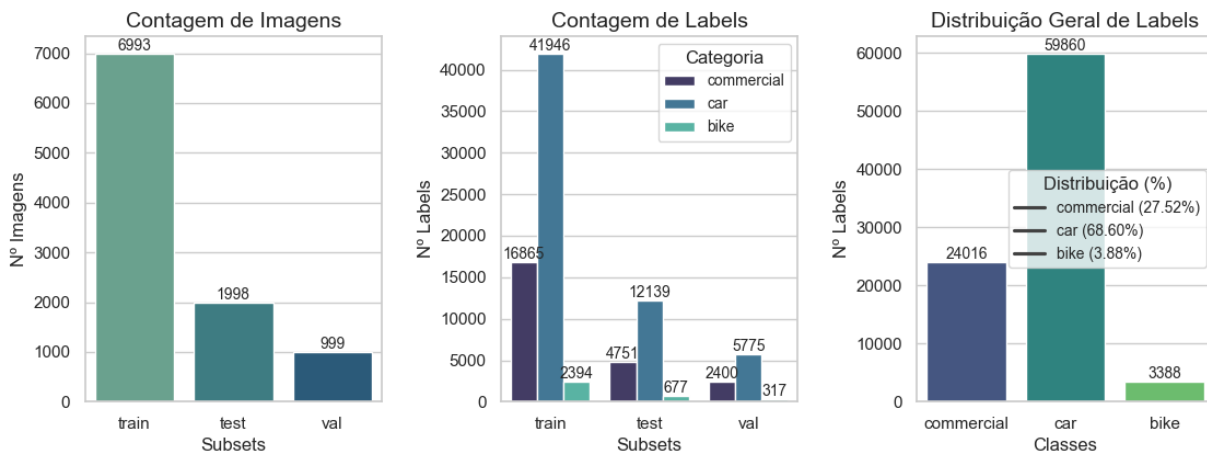


Imagem 35 – Distribuição Dataset Eventos v2

- Tamanho do Dataset: Total de 9.990 de imagens, e 87.264 de labels.
- Distribuição dos Dados:
 - Train: 6.993 imagens (commercial: 16.865 labels, car: 41.946 labels, bike: 2.394 labels)
 - Test: 1.998 imagens (commercial: 4.751 labels, car: 12.139 labels, bike: 677 labels)
 - Val: 999 imagens (commercial: 2.400 labels, car: 5.775 labels, bike: 317 labels)
- Data Augmentation: Horizontal Flip.

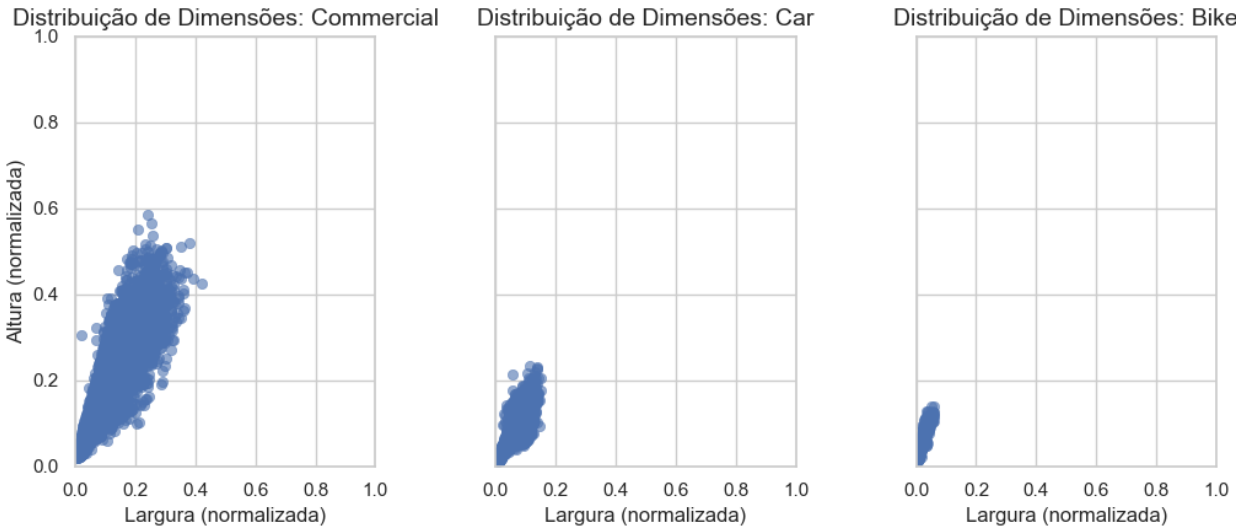


Imagem 36 – Distribuição Bouding Boxes Modelo de Eventos v2

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas e largas, sugerindo que os objetos dessa classe ocupam uma área de eixo X e Y maior em relação as outras classes (pois estamos rotulando o veículo inteiro, não apenas a cabine).
- Distribuição da Classe 'car': As bounding boxes para a classe car possuem uma ampla variação de tamanhos, porém quando submetido sobre uma visão panorâmica, é relativamente difícil distinguir padrões entre suas alturas e larguras.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas e centradas, nem tão grandes quanto os comerciais, nem tão dinâmicas quanto os carros.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 25, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

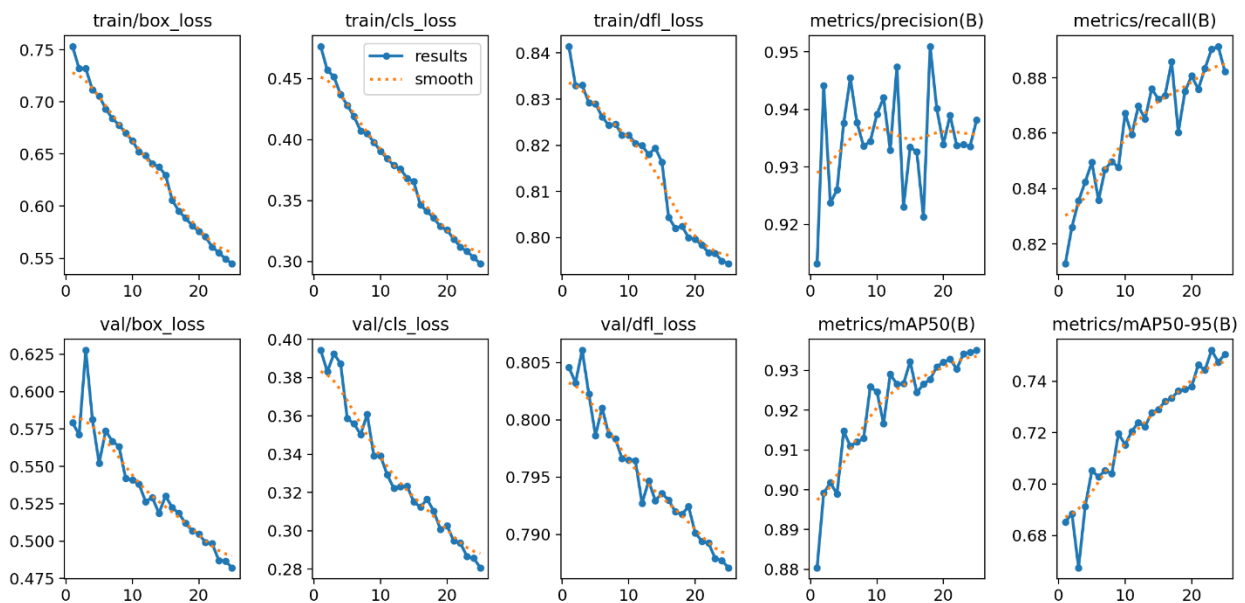


Imagem 37 – Métricas Gerais Modelo de Eventos v2

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.98**, demonstrando baixa taxa de falsos positivos.
 - O recall cresce até **0.89**, mostrando que a maioria dos objetos foi corretamente detectada.

- mAP:
 - O **mAP@50** chega a aproximadamente **0.94**, destacando bom desempenho em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.76**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

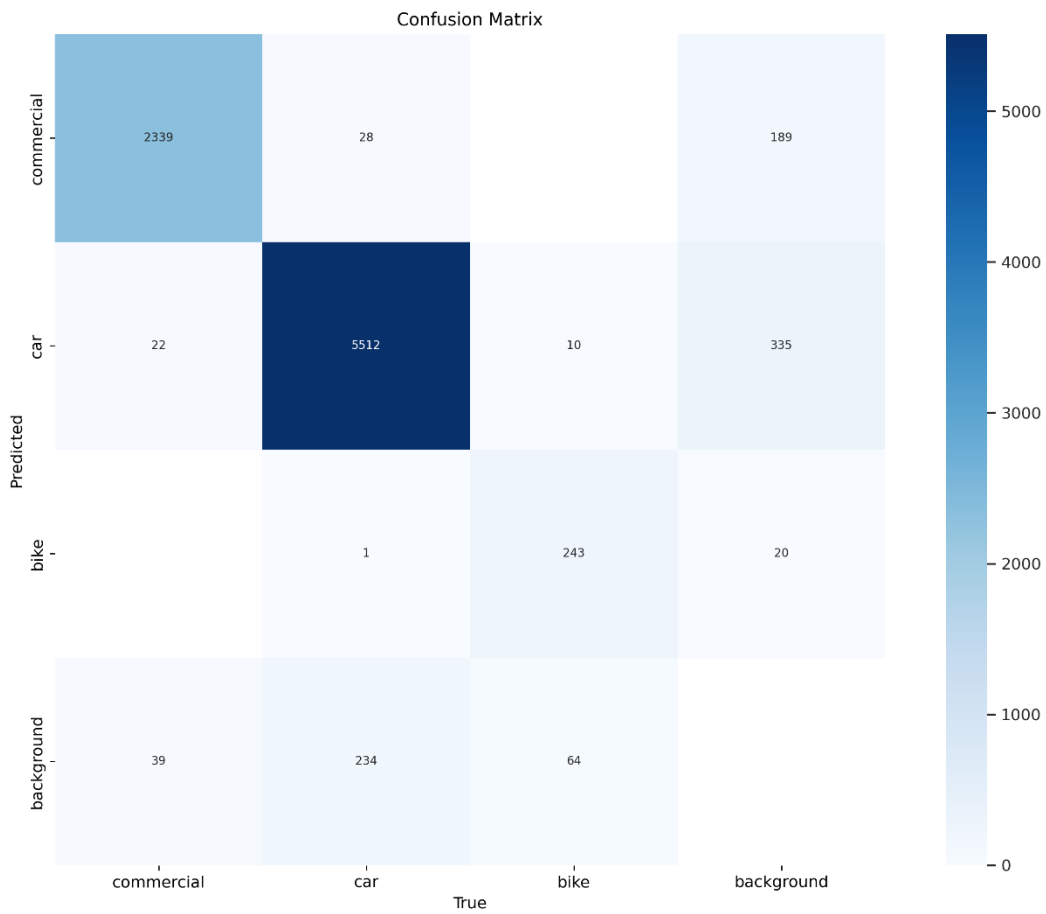


Imagem 38 – Matriz da Confusão Modelo de Eventos v2

Resultados:

- Classe "Commercial":
 - **TP:** 2339 detecções corretas.
 - **FN:** 217 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 5512 detecções corretas.
 - **FN:** 367 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 243 detecções corretas.
 - **FN:** 21 casos não detectados corretamente como " Bike".
- Classe "Background":
 - **FP:** 544 casos foram incorretamente classificados entre os veículos.
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para as classes "Commercial" e "Car", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos. Acreditasse que quanto mais longe um veículo estiver da câmera, maior será a área de falsos positivos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para as classes "Bike" e "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados



Imagem 39 – Resultados Iniciais Modelo de Eventos v2

5.3 Modelo de Eventos Versão 3.0

Conjunto de Dados

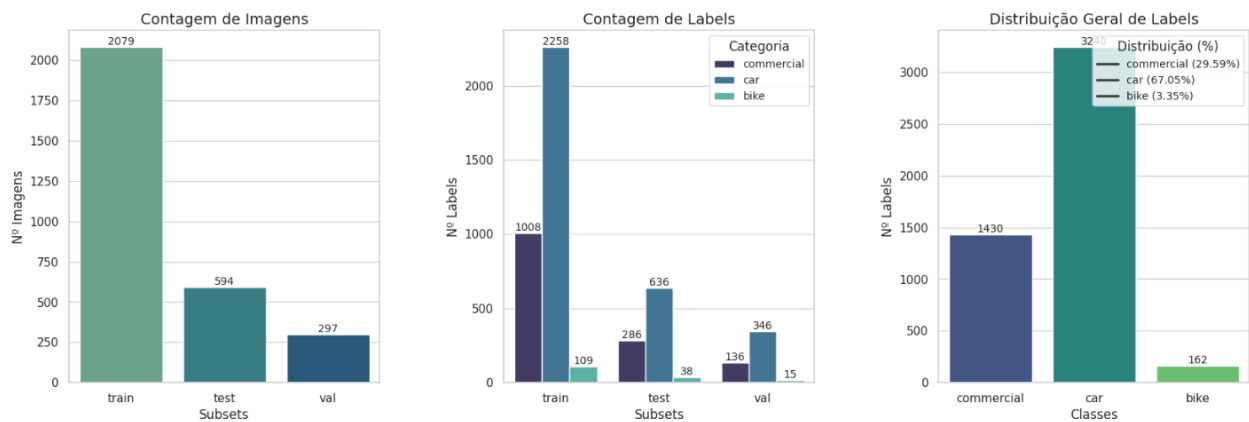


Imagem 40 – Distribuição Dataset Eventos v3

- Tamanho do Dataset: Total de 2.970 de imagens, e 4.832 de labels.
- Distribuição dos Dados:

- Train: 2.079 imagens (commercial: 1.008 labels, car: 2.258 labels, bike: 109 labels)
- Test: 594 imagens (commercial: 286 labels, car: 636 labels, bike: 38 labels)
- Val: 297 imagens (commercial: 136 labels, car: 346 labels, bike: 15 labels)
- Data Augmentation: Horizontal Flip.

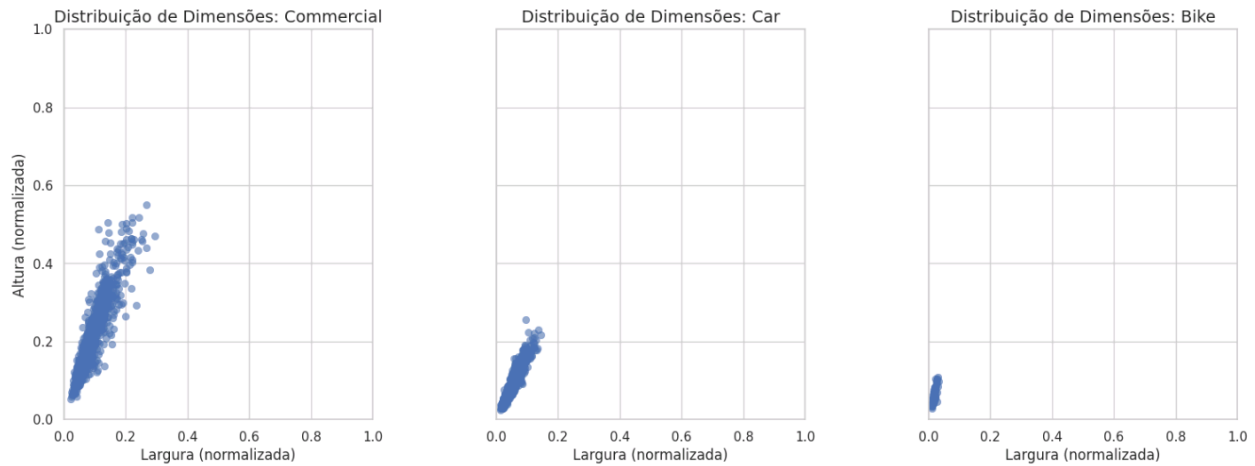


Imagem 41 – Distribuição Bouding Boxes Modelo de Eventos v3

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas e largas, sugerindo que os objetos dessa classe ocupam uma área de eixo X e Y maior em relação as outras classes (pois estamos rotulando o veículo inteiro, não apenas a cabine).
- Distribuição da Classe 'car': As bounding boxes para a classe car possuem uma ampla variação de tamanhos, porém quando submetido sobre uma visão panorâmica, é relativamente difícil distinguir padrões entre suas alturas e larguras.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas e centradas, nem tão grandes quanto os comerciais, nem tão dinâmicas quanto os carros.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 25, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

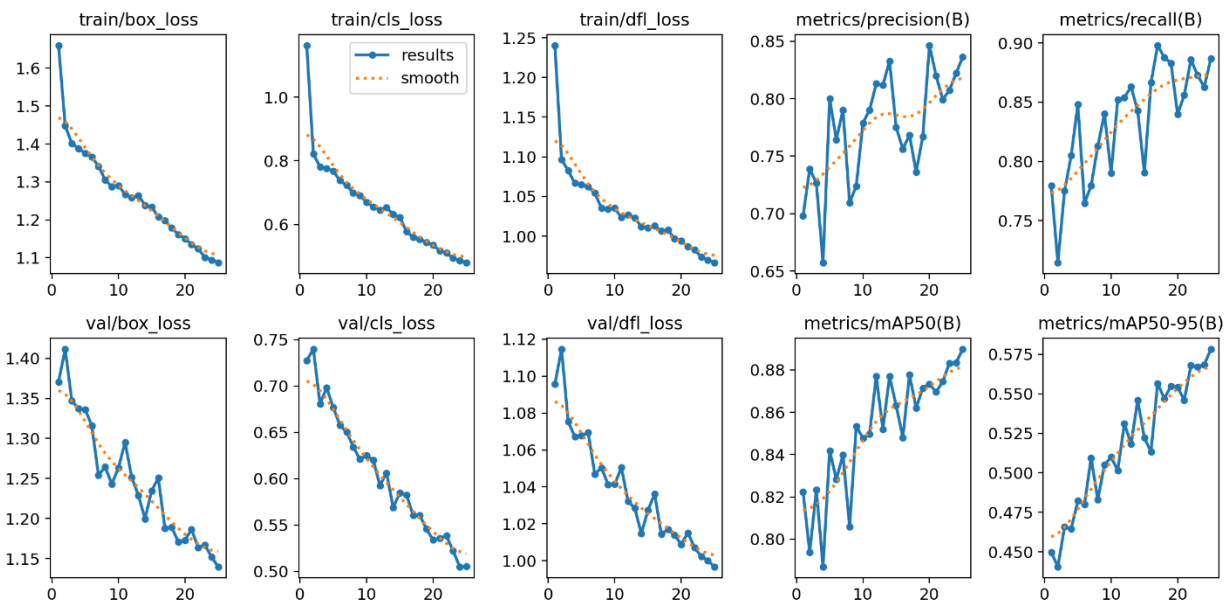


Imagem 42 – Métricas Gerais Modelo de Eventos v3

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.85**, demonstrando baixa taxa de falsos positivos.
 - O recall cresce até **0.90**, mostrando que a maioria dos objetos foi corretamente detectada.
- mAP:

- O **mAP@50** chega a aproximadamente **0.89**, destacando bom desempenho em detecções mais amplas.
- O **mAP@50-95** alcança cerca de **0.57**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

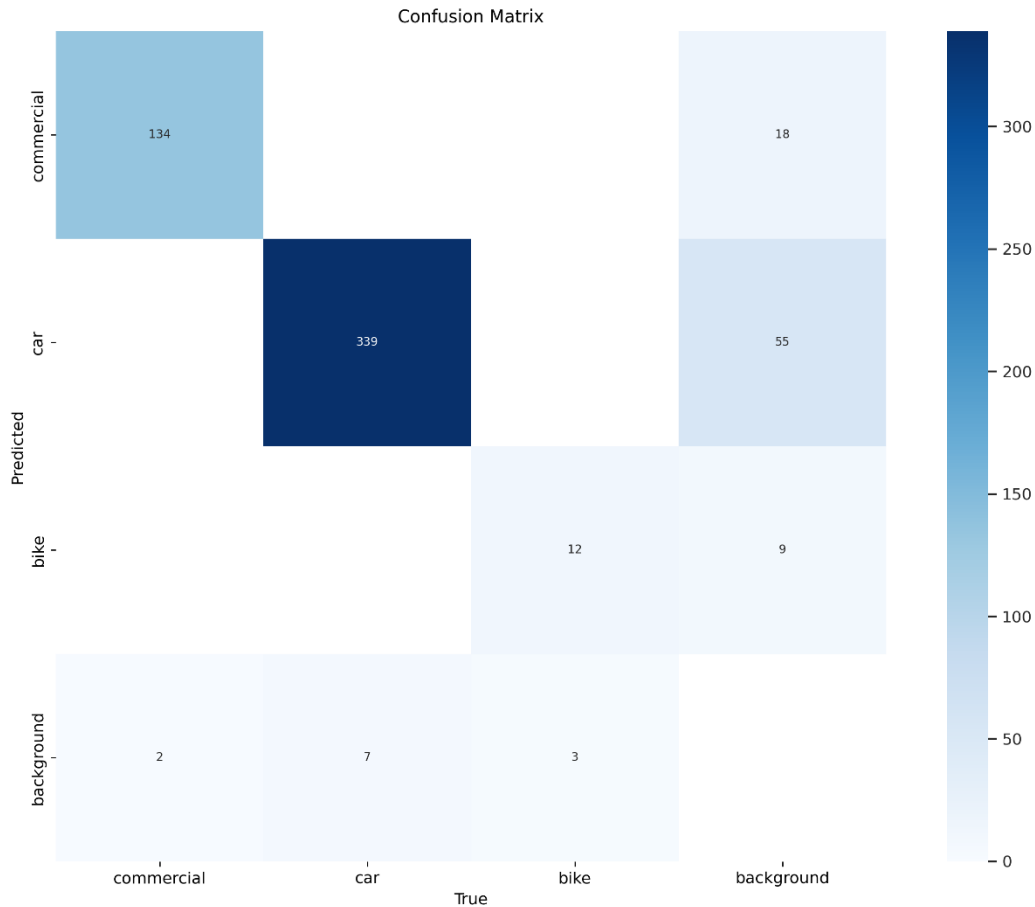


Imagem 43 – Matriz da Confusão Modelo de Eventos v3

Resultados:

- Classe "Commercial":
 - **TP:** 134 detecções corretas.
 - **FN:** 18 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 339 detecções corretas.

- **FN:** 55 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 12 detecções corretas.
 - **FN:** 9 casos não detectados corretamente como " Bike".
- Classe "Background":
 - **FP:** 12 casos foram incorretamente classificados entre os veículos.
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para as classes "Commercial" e "Car", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos. Acreditasse que quanto mais longe um veículo estiver da câmera, maior será a área de falsos positivos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para as classes "Bike" e "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados

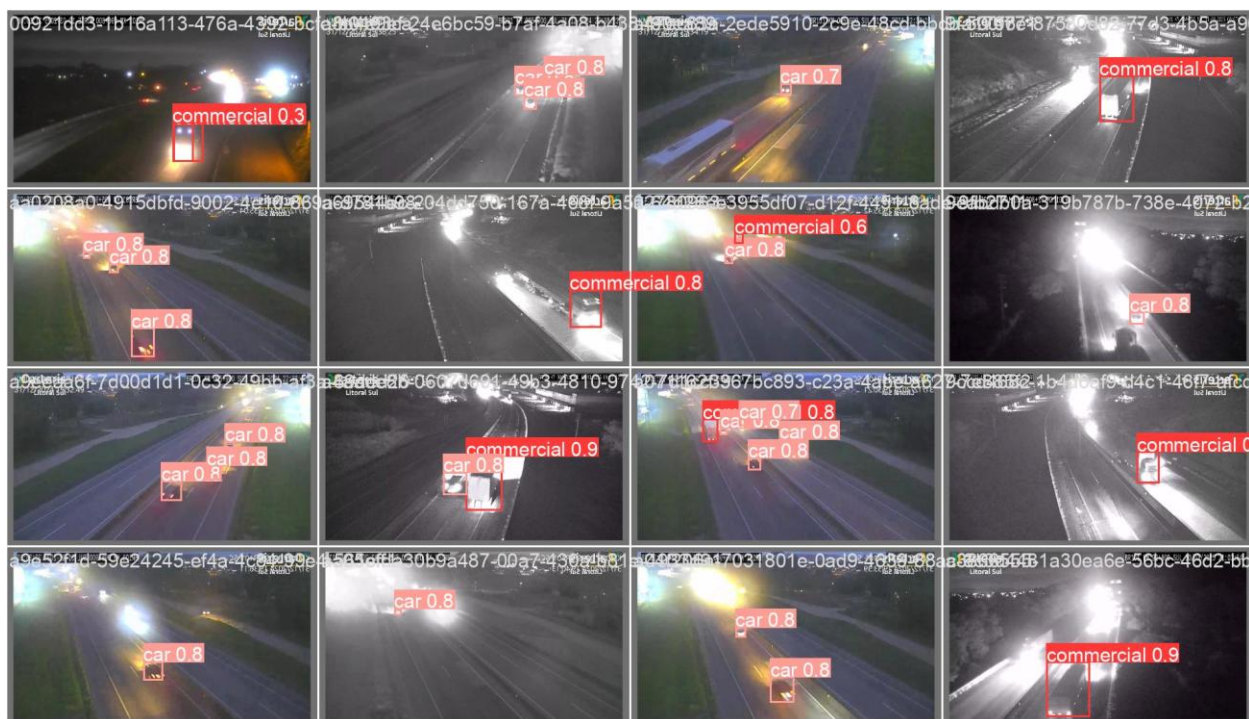


Imagem 44 – Resultados Iniciais Modelo de Eventos v3

5.4 Modelo de Incêndio Versão 2.0 Conjunto de Dados (Sem modificações)

O conjunto de dados não sofreu modificações, foi realizado apenas um fine tuning do modelo com o mesmo dataset da versão anterior.

Fine-tuning

O modelo de Detecção de Incêndio foi inicialmente treinado com uma base de dados específica e, posteriormente, passou por uma etapa de ajuste fino (fine-tuning). Essa segunda etapa teve como objetivo aprimorar a performance do modelo, tornando-o mais sensível, preciso e confiável nas detecções em cenários reais.

Ao invés de reiniciar o treinamento do zero, utilizamos o modelo já treinado anteriormente como base, e aplicamos um novo treinamento com parâmetros cuidadosamente configurados.

Abaixo, estão descritos os principais pontos da configuração aplicada e o que cada um representa.

Estratégias de Parada e Otimização

- **patience=7:** Se o desempenho não melhorar após 7 ciclos, o treinamento é encerrado automaticamente (early stopping).

- **optimizer='AdamW'**: Otimizador que ajuda a fazer ajustes mais finos e estáveis nos pesos do modelo.
- **lr0=0.0003**: Taxa de aprendizado inicial. Um valor baixo permite mudanças sutis, evitando que o modelo perca o que já aprendeu.
- **lrf=0.01**: Fator que define o valor mínimo da taxa de aprendizado ao longo do tempo.
- **weight_decay=0.01**: Evita que o modelo fique superajustado penalizando pesos muito altos.
- **warmup_epochs=2.0**: Durante as primeiras 2 épocas, o modelo treina com cautela, aumentando gradualmente o ritmo.
- **warmup_bias_lr=0.1**: Aplicado aos termos de viés da rede no início do treinamento, também de forma gradual.

Augmentações (Aumentos de Dados) Visuais

- **hsv_h / hsv_s / hsv_v**: Pequenas variações de matiz, saturação e brilho aplicadas às imagens para aumentar a diversidade sem perder a fidelidade visual.
- **translate=0.03**: Leves translações (movimentações horizontais/verticais) da imagem.
- **scale=0.2**: Variação no tamanho dos objetos na imagem, para simular diferentes distâncias.
- **fliplr=0.2**: Aplicação de espelhamento horizontal em 20% das imagens.
- **mosaic / mixup / copy_paste / shear / flipud = 0.0**: Técnicas mais agressivas de aumento foram desativadas para preservar as características originais das imagens e evitar ruído no ajuste fino.

Pesos das Funções de Perda (Loss)

- **box=7.5**: Aumenta a importância da precisão na localização dos incêndios na imagem (bounding boxes).
- **cls=0.5**: Peso menor para a classificação de categorias, pois o foco principal é em um tipo de objeto (incêndio).
- **dfl=1.5**: Controla o impacto da regressão de distância entre caixas preditas e reais.

Configurações de Infraestrutura

- **pretrained=False**: O modelo NÃO usou pesos genéricos (ex: COCO), mas sim o modelo treinado anteriormente na base de incêndios.

- **cache=True:** Armazena as imagens em RAM para agilizar o carregamento e acelerar o treinamento.
- **workers=4:** Define 4 subprocessos para carregar os dados em paralelo, otimizando a performance.

Considerações Finais da etapa de fine-tuning

Essa etapa de ajuste fino foi essencial para transformar um modelo bom em um modelo altamente adaptado ao seu caso real de uso para detecção de incêndio em ambientes específicos. O processo foi cuidadosamente planejado para preservar o conhecimento adquirido, evitar sobreajuste e refinar os detalhes mais críticos, como localização precisa e robustez visual.

Resultados

- Métricas Gerais:

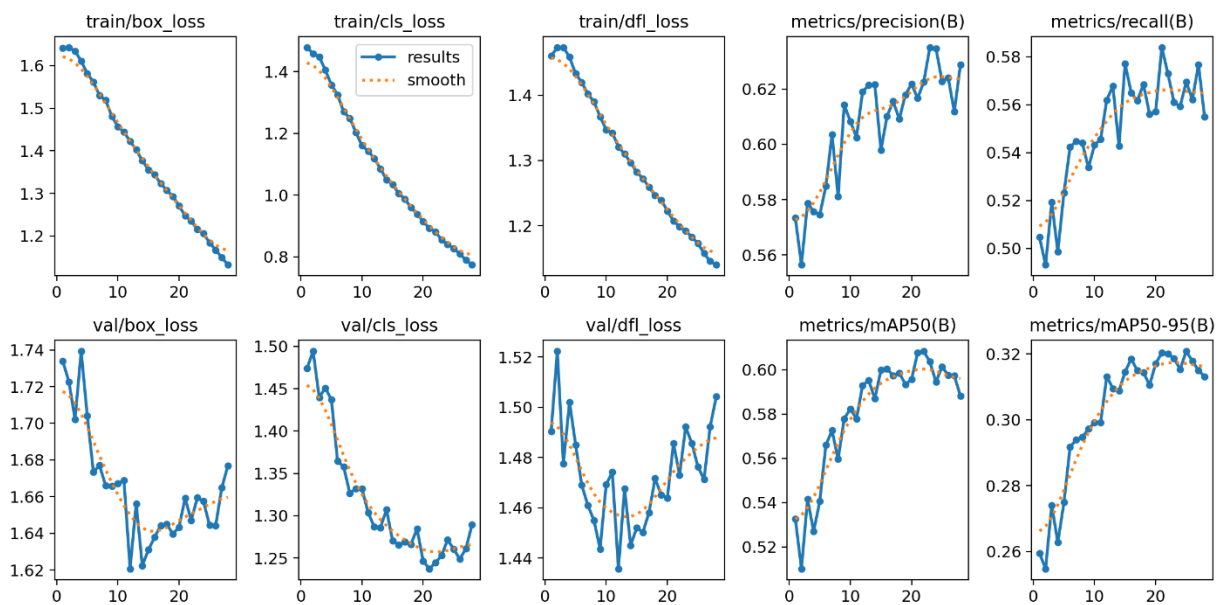


Imagem 45 – Métricas Gerais Modelo de Incêndios v2

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **df_l_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:

- A precisão atinge cerca de **0.63**, demonstrando uma taxa moderada de falsos positivos.
- O recall cresce até **0.58**, mostrando que a metade dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.60**, destacando um desempenho mediano em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.32**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

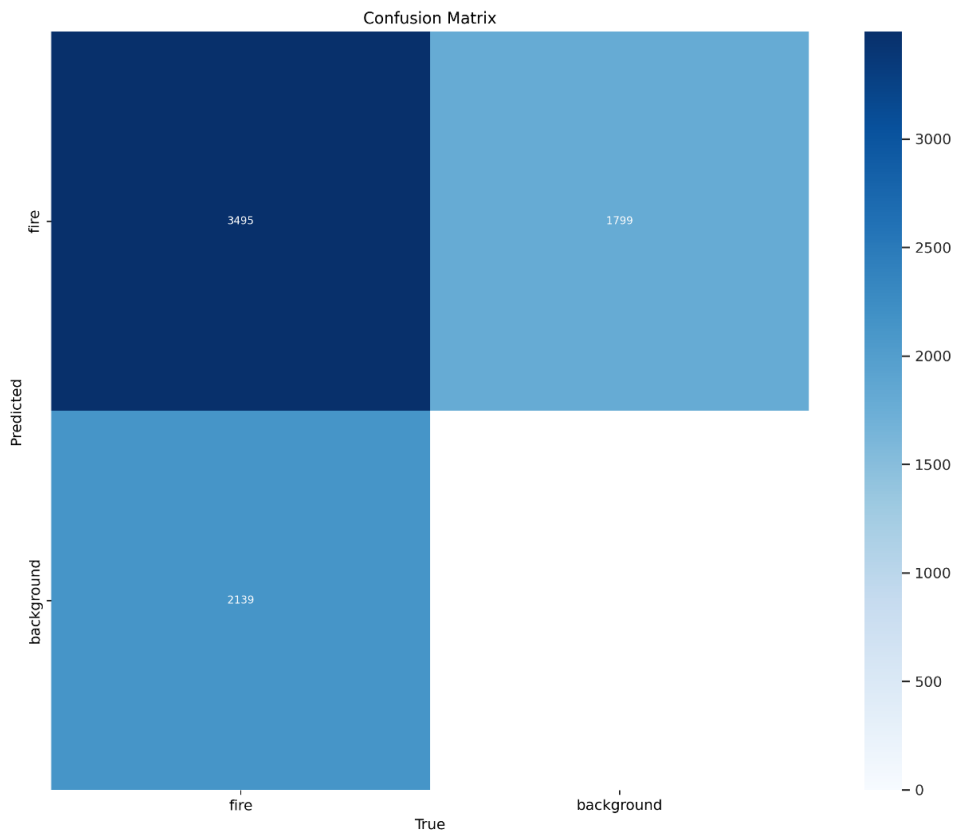


Imagem 46 – Matriz da Confusão Modelo de Incêndios v2

Resultados:

- Classe "Fire":
 - **TP:** 3.495 detecções corretas.

- **FN:** 1.799 casos não detectados corretamente como "Fire".
- Classe "Background":
 - **FP:** 2.139 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade mediana (recall)** para a classe "Fire", o que significa que ele consegue identificar corretamente boa parte dos casos dentro dessa classe, com **3.495 detecções corretas e 1.799 detecções incorretas**.
- A precisão é mediana, o que sugere um número alto de falsos positivos (**FP**), especialmente na classe "Background", onde **2.139 erros de classificação** ocorreram.
- O desempenho geral do modelo é moderado, com um recall de aproximadamente 60% para a classe "Fire". Para melhorar a precisão e reduzir falsos positivos na classe "Background", recomenda-se a aplicação de técnicas como balanceamento de classes, uso de datasets mais diversificados, ou ajustes nos thresholds de detecção.

Visualização dos Resultados



Imagem 47 – Resultados Iniciais Modelo de Incêndios v2

5.5 Modelo de Pessoas Versão 2.0

Conjunto de Dados

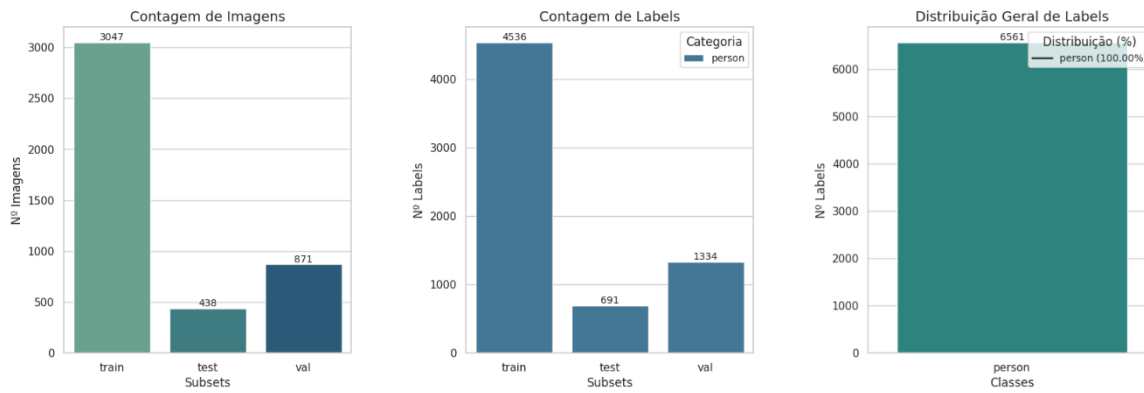


Imagem 48 – Distribuição Dataset Pessoas v2

- Tamanho do Dataset: Total de 4.356 de imagens, e 6.561 de labels.
- Distribuição dos Dados:
 - Train: 3.047 imagens (Person: 4.536 labels)
 - Test: 438 imagens (Person: 691 labels)
 - Val: 871 imagens (Person: 1.334 labels)

- Data Augmentation: Horizontal Flip, RandomBrightnessContrast, HueSaturationValue e ShiftScaleRotate.

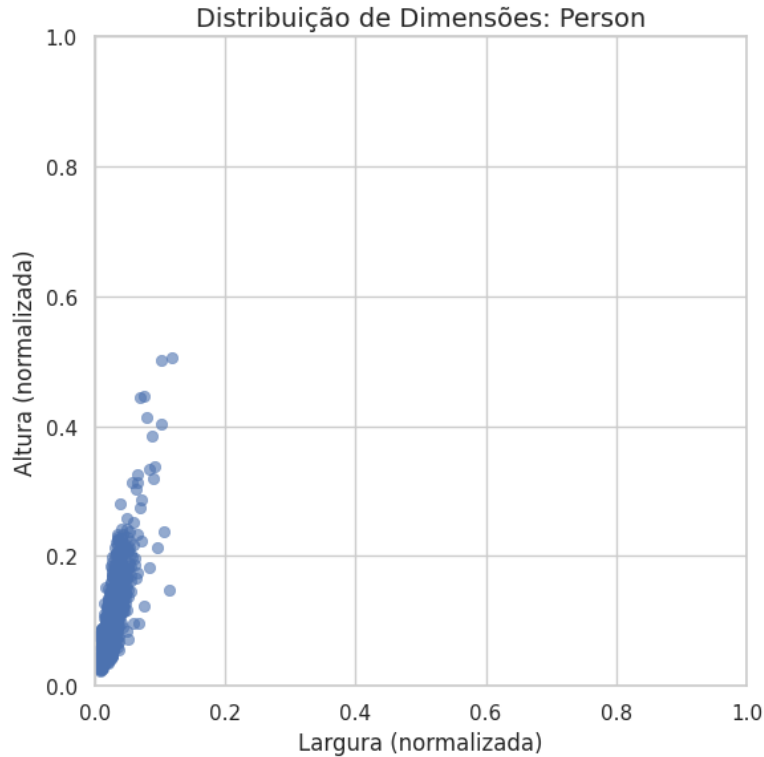


Imagem 49 – Distribuição Bounding Boxes Pessoas v2

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- As bounding boxes para a classe "Person" estão concentradas em dimensões menores, especialmente na faixa de 0.0 a 0.2 para largura e 0.0 a 0.4 para altura.
- A distribuição das alturas apresenta uma leve dispersão até 0.6, sugerindo a presença de algumas imagens com objetos maiores.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.

- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

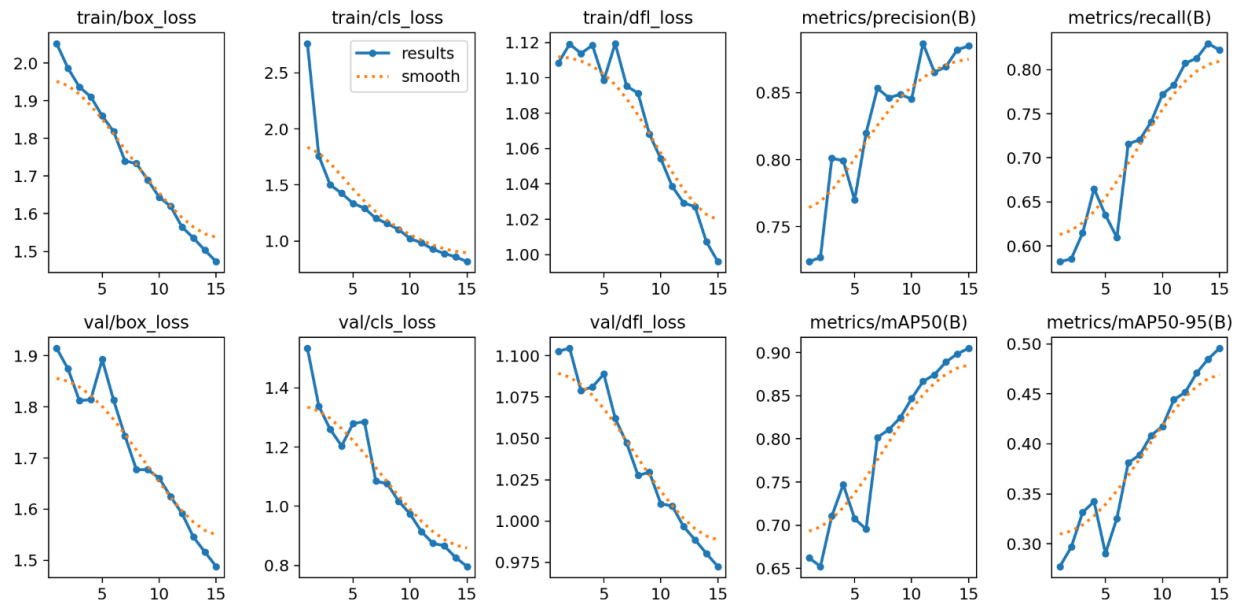


Imagem 50 – Métricas Gerais Modelo de Pessoas v2

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente ao longo das épocas, tanto no treinamento quanto na validação. Isso indica uma boa convergência do modelo, com aprendizado progressivo e estabilização dos pesos.
- Precisão e Recall:
 - Precisão cresce até aproximadamente **0.88**, o que indica uma taxa relativamente baixa de falsos positivos.
 - Recall aumenta até **0.82**, mostrando que o modelo está conseguindo identificar a maioria dos objetos corretamente.
- mAP:
 - **mAP@50** atinge cerca de **0.90**, destacando um desempenho sólido em detecções menos rigorosas.

- **mAP@50-95** alcança aproximadamente **0.50**, sugerindo que o modelo ainda pode melhorar em detecções com limiares mais elevados.

Matriz da confusão:

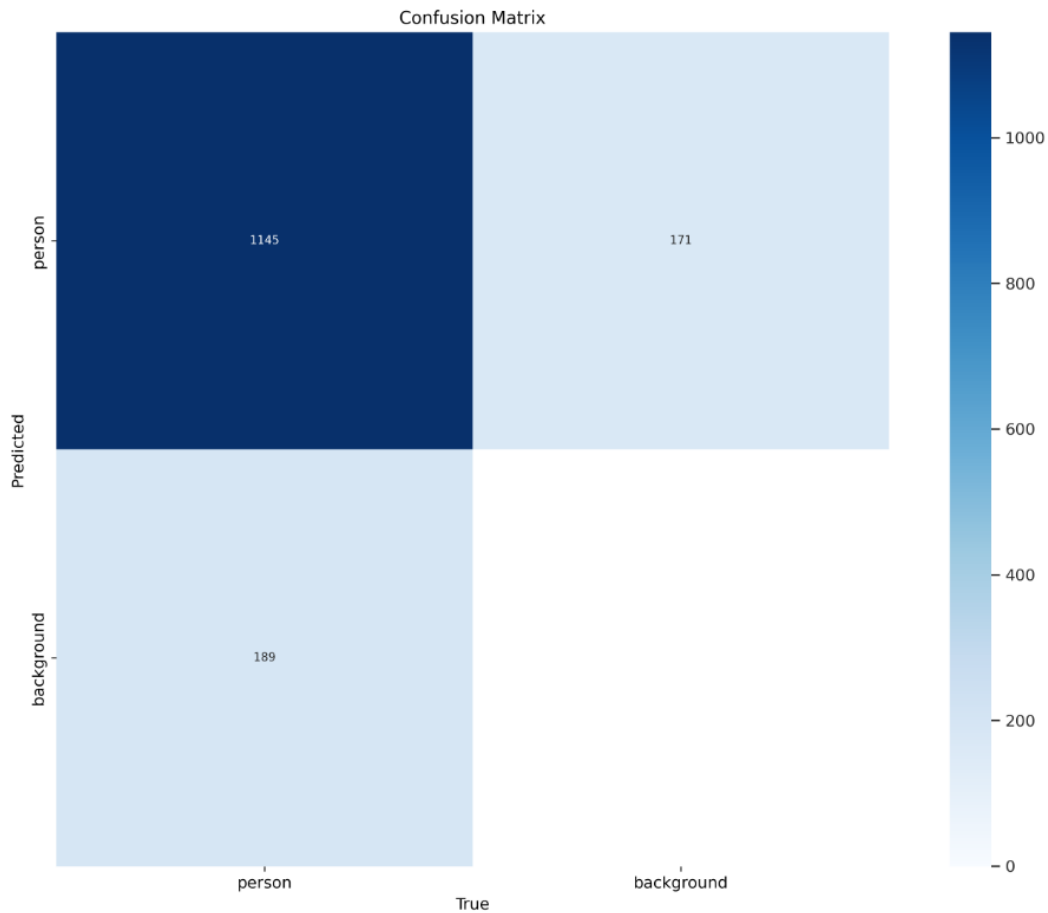


Imagem 51 – Matriz da Confusão Modelo de Pessoas v2

Resultados:

- Classe "Person":
 - **TP:** 1145 detecções corretas.
 - **FN:** 189 casos não detectados corretamente (classificados como "Background").
- Classe "Background":
 - **FP:** 171 amostras classificadas incorretamente como "Person".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade alta (recall de 82%)** para a classe "Person", indicando que consegue identificar corretamente a maioria das pessoas.
- A **precisão está acima da média (88%)**, sugerindo que os falsos positivos são relativamente baixos.
- O **desempenho geral do modelo é muito bom**, com **mAP@50 de 90%** e **mAP@50-95 de 50%**, indicando que as detecções em limiares mais altos são razoavelmente boas, mas ainda podem ser aprimoradas.

Visualização dos Resultados



Imagem 52 – Resultados Iniciais Modelo de Pessoas v2

5.6 Modelo de Pessoas Versão 3.0 Conjunto de Dados (Sem modificações)

O conjunto de dados não sofreu modificações, foi realizado apenas um fine tuning do modelo com o mesmo dataset da versão anterior.

Fine-tuning

O modelo de Detecção de Pessoas foi inicialmente baseado em uma arquitetura pré-treinada e, posteriormente, passou por uma etapa de ajuste fino (fine-tuning). Essa segunda etapa teve como objetivo aprimorar a performance do modelo, tornando-o mais sensível, preciso e confiável nas detecções de pessoas em diferentes ambientes e cenários visuais.

Ao invés de reiniciar o treinamento do zero, utilizamos uma versão avançada da arquitetura YOLO (YOLOv8 Large), que oferece maior capacidade de detecção, e aplicamos um novo treinamento com parâmetros cuidadosamente configurados para otimizar o uso da GPU e garantir maior precisão.

Abaixo, estão descritos os principais pontos da configuração aplicada e o que cada um representa.

Estratégias de Parada e Otimização

- **patience=10**: Caso o desempenho do modelo não melhore por 10 ciclos consecutivos, o treinamento é encerrado automaticamente (early stopping), evitando desperdício de tempo e recursos.
- **optimizer='AdamW'**: Otimizador que garante maior estabilidade e controle durante o ajuste dos pesos do modelo.
- **lr0=0.001**: Taxa de aprendizado inicial moderada, definida para promover atualizações consistentes dos pesos sem causar instabilidade.
- **lrf=0.0001**: Valor final da taxa de aprendizado, mais baixo, para permitir um refinamento preciso nas últimas etapas do treinamento.
- **warmup_epochs=3**: Primeiras 3 épocas com ritmo de treinamento mais suave, ajudando a estabilizar o processo de ajuste.

Estratégias de Detecção e Precisão

- **iou=0.4**: Define o limiar de Interseção sobre União para considerar uma detecção correta. Um valor mais conservador, que favorece o aumento do recall.

- **conf=0.25**: Define a confiança mínima necessária para considerar uma detecção válida. Um valor intermediário, equilibrando sensibilidade e precisão.

Augmentações (Aumentos de Dados) Visuais

- **mosaic=1.0**: Técnica que junta várias imagens em uma só, promovendo maior variedade e auxiliando na detecção em múltiplos contextos.
- **scale=0.5**: Permite ao modelo aprender com objetos (pessoas) de diferentes tamanhos, simulando distâncias variadas.
- **perspective=0.001**: Pequena distorção de perspectiva aplicada para melhorar a robustez do modelo em ambientes com diferentes ângulos de captura.

Configurações de Infraestrutura

- **model = YOLO('yolov8l.pt')**: Utilização da arquitetura YOLOv8 Large, voltada para tarefas onde é necessário maior precisão, como é o caso da detecção de pessoas.
- **data='data_pessoas.yaml'**: Conjunto de dados específico contendo imagens e anotações para a detecção de pessoas.
- **imgsz=960**: Tamanho de imagem reduzido em comparação ao máximo, visando economia de memória GPU sem perder qualidade nas detecções.
- **epochs=50**: Número de ciclos de treinamento definidos para permitir convergência adequada do modelo.
- **batch=8**: Número de imagens processadas simultaneamente, ajustado para não exceder a memória disponível na GPU.
- **device='cuda:0'**: O treinamento foi executado utilizando a GPU principal da máquina, garantindo velocidade e eficiência computacional.
- **name='pessoas_v4'**: Nome do experimento, usado para organização dos resultados no ambiente de treino.

Considerações Finais da etapa de fine-tuning

Essa etapa de ajuste fino foi essencial para transformar um modelo base em um modelo especializado e confiável para o seu caso de uso: detecção de pessoas em imagens reais. A escolha

dos hiperparâmetros e augmentações foi feita com foco na precisão, controle de consumo de memória e robustez em diferentes situações visuais. O modelo resultante está mais bem preparado para identificar pessoas com maior confiança, mesmo em cenários complexos ou com pessoas parcialmente visíveis.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

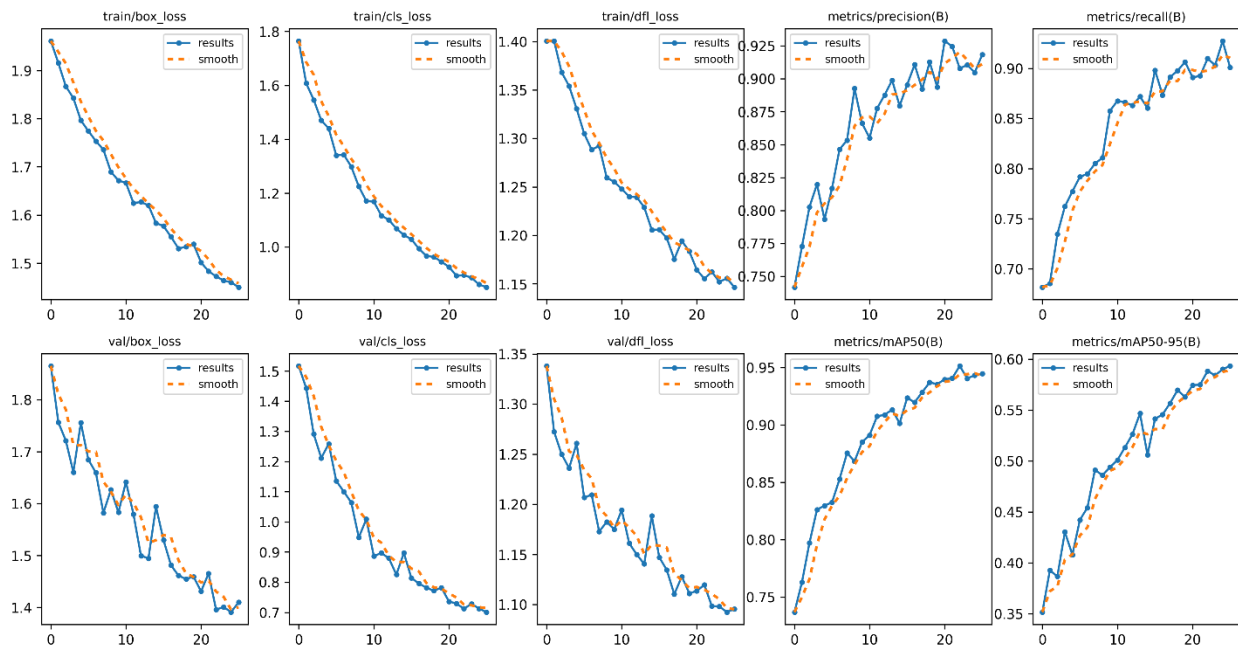


Imagem 53 – Métricas Gerais Modelo de Pessoas v3

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente ao longo das épocas, tanto no treinamento quanto na validação. Isso indica uma boa convergência do modelo, com aprendizado progressivo e estabilização dos pesos.
- Precisão e Recall:
 - Precisão cresce até aproximadamente **0.92**, o que indica uma taxa relativamente baixa de falsos positivos.
 - Recall aumenta até **0.92**, mostrando que o modelo está conseguindo identificar a maioria dos objetos corretamente.
- mAP:
 - **mAP@50** atinge cerca de **0.95**, destacando um desempenho sólido em detecções menos rigorosas.
 - **mAP@50-95** alcança aproximadamente **0.59**, sugerindo que o modelo ainda pode melhorar em detecções com limiares mais elevados.

Matriz da confusão:

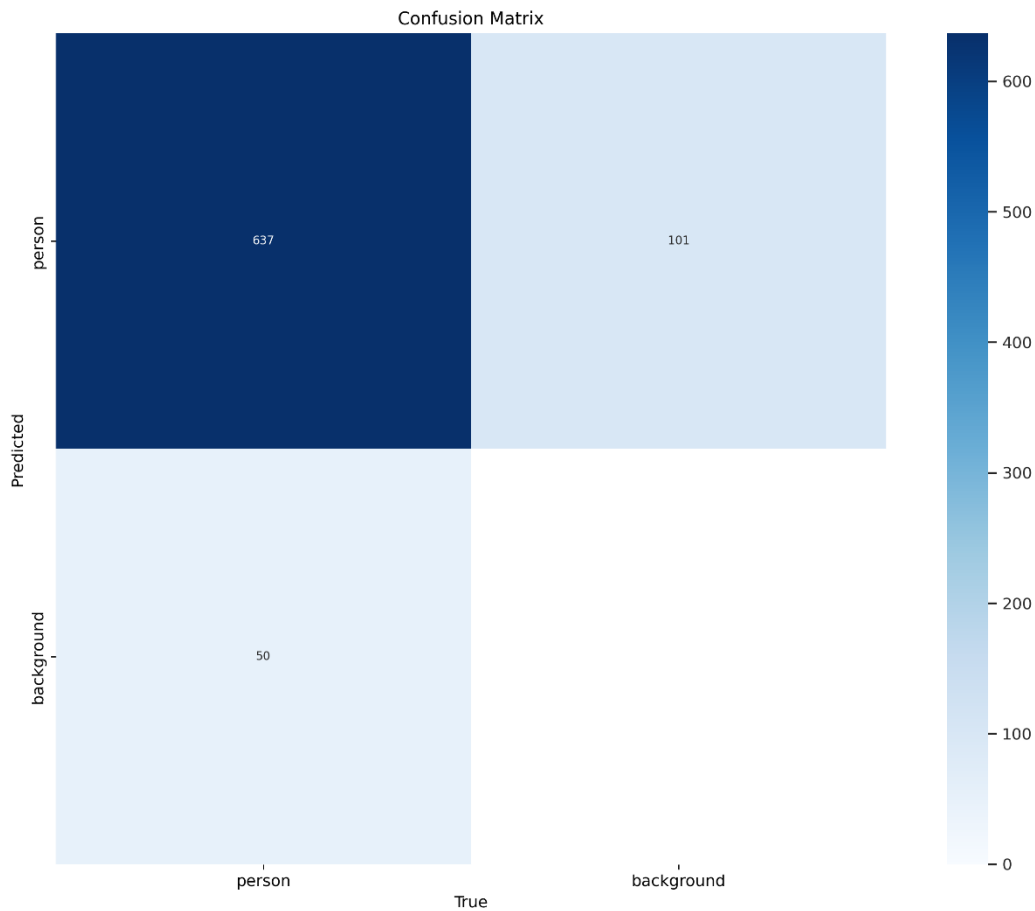


Imagem 54 – Matriz da Confusão Modelo de Pessoas v3

Resultados:

- Classe " Person":
 - **TP:** 637 detecções corretas.
 - **FN:** 101 casos não detectados corretamente (classificados como "Background").
- Classe "Background":
 - **FP:** 50 amostras classificadas incorretamente como "Person".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade alta (recall de 92%)** para a classe "Person", indicando que consegue identificar corretamente a maioria das pessoas.

- A **precisão** está **acima da média (92%)**, sugerindo que os falsos positivos são relativamente baixos.
- O **desempenho geral** do modelo é **muito bom**, com **mAP@50 de 95%** e **mAP@50-95 de 59%**, indicando que as detecções em limiares mais altos são razoavelmente boas, mas ainda podem ser aprimoradas.

Visualização dos Resultados



Imagem 55 – Resultados Iniciais Modelo de Pessoas v3

5.7 Modelo de Objetos Versão 2.0 Conjunto de Dados (Sem modificações)

O conjunto de dados não sofreu modificações, foi realizado apenas um fine tuning do modelo com o mesmo dataset da versão anterior.

Fine-tuning

O modelo de Detecção de Objetos foi inicialmente baseado em uma arquitetura pré-treinada e, posteriormente, passou por uma etapa de ajuste fino (fine-tuning). Essa segunda etapa teve como

objetivo aprimorar a performance do modelo, tornando-o mais sensível, preciso e confiável nas detecções em cenários reais.

Ao invés de reiniciar o treinamento do zero, utilizamos um modelo base já consolidado (YOLOv8 Medium) e aplicamos um novo treinamento com parâmetros cuidadosamente configurados.

Abaixo, estão descritos os principais pontos da configuração aplicada e o que cada um representa.

Estratégias de Parada e Otimização

- **patience=10**: Se o desempenho não melhorar após 10 ciclos consecutivos, o treinamento é encerrado automaticamente (early stopping), evitando sobreajuste e otimizando o uso de recursos computacionais.
- **epochs=50**: Define um limite de até 50 ciclos de treinamento, permitindo que o modelo tenha tempo para convergir.
- **batch=8**: Define o número de imagens processadas por vez. Esse valor foi escolhido para otimizar o uso da GPU (14GB), garantindo estabilidade e boa performance.
- **device=0**: Força o treinamento a ser executado na GPU principal, garantindo maior velocidade no processamento.

Augmentações (Aumentos de Dados) Visuais

- **hsv_h / hsv_s / hsv_v**: Variações nas cores (matiz, saturação e brilho) que ajudam o modelo a se adaptar a diferentes condições de iluminação.
- **translate=0.1**: Simula pequenos deslocamentos dos objetos na imagem, o que torna o modelo mais robusto a variações de posicionamento.
- **scale=0.5**: Permite variação no tamanho dos objetos, auxiliando na detecção de objetos pequenos ou grandes.
- **fliplr=0.5**: Espelhamento horizontal aplicado em metade das imagens, útil para detectar objetos em diferentes orientações.
- **mosaic=1.0**: Técnica avançada que junta quatro imagens em uma só, promovendo variedade nas combinações visuais e aumentando a robustez do modelo.
- **mixup=0.2**: Mistura de duas imagens com seus rótulos, o que ajuda a melhorar a generalização em cenários mais variados.
- **degrees / shear / perspective / flipud = 0.0**: Esses ajustes foram desativados para evitar distorções que pudessem comprometer a qualidade das detecções de objetos reais.

Configurações de Infraestrutura

- **model = YOLO('yolov8m.pt')**: Utilização da versão Medium do YOLOv8 como ponto de partida. Essa versão tem maior capacidade que modelos menores (como o Nano), sendo mais indicada para tarefas que exigem maior precisão e detecção de objetos pequenos.
- **data='data_objetos.yaml'**: Define o conjunto de dados com as classes e imagens específicas para o problema de detecção de objetos.
- **imgsz=960**: Utilização de imagens maiores para garantir que objetos pequenos sejam detectados com mais facilidade.
- **save=True / save_period=1**: Salva os resultados de cada época, permitindo rastrear a evolução do modelo ao longo do tempo.
- **project='runs/detect' / name='objetos_v2' / exist_ok=True**: Define a organização das pastas de saída do experimento para facilitar o controle e versionamento.
- **val=True**: Garante que a validação seja executada durante o treinamento para monitorar o desempenho.

Considerações Finais da etapa de fine-tuning

Essa etapa de ajuste fino foi essencial para transformar um modelo base genérico em um modelo especializado para o seu caso real de uso para detecção de objetos em imagens específicas do projeto. O processo foi cuidadosamente configurado para garantir boa capacidade de generalização, evitar sobreajuste e refinar as características visuais mais importantes para o contexto de aplicação.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 4, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

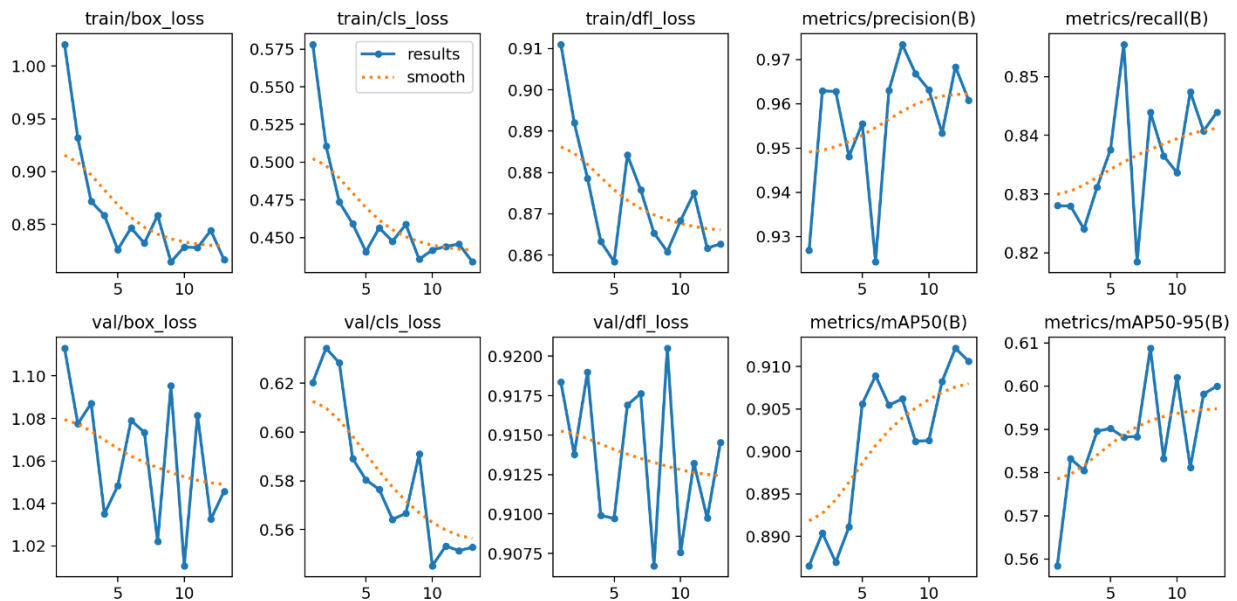


Imagem 56 – Métricas Gerais Modelo de Objetos v2

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando uma boa convergência do modelo e redução do erro ao longo das épocas.

- Precisão e Recall:

- A precisão atinge aproximadamente **0.98**, demonstrando um bom controle de falsos positivos.
- O recall cresce até **0.86**, indicando que o modelo consegue detectar corretamente cerca de 60% dos objetos presentes nas imagens.

- mAP:

- O **mAP@50** atinge **0.91**, mostrando um desempenho satisfatório para detecções mais amplas.
- O **mAP@50-95** alcança aproximadamente **0.61**, sugerindo que ainda há margem para melhorias quando se trata de detecções mais rigorosas.

Matriz da confusão:

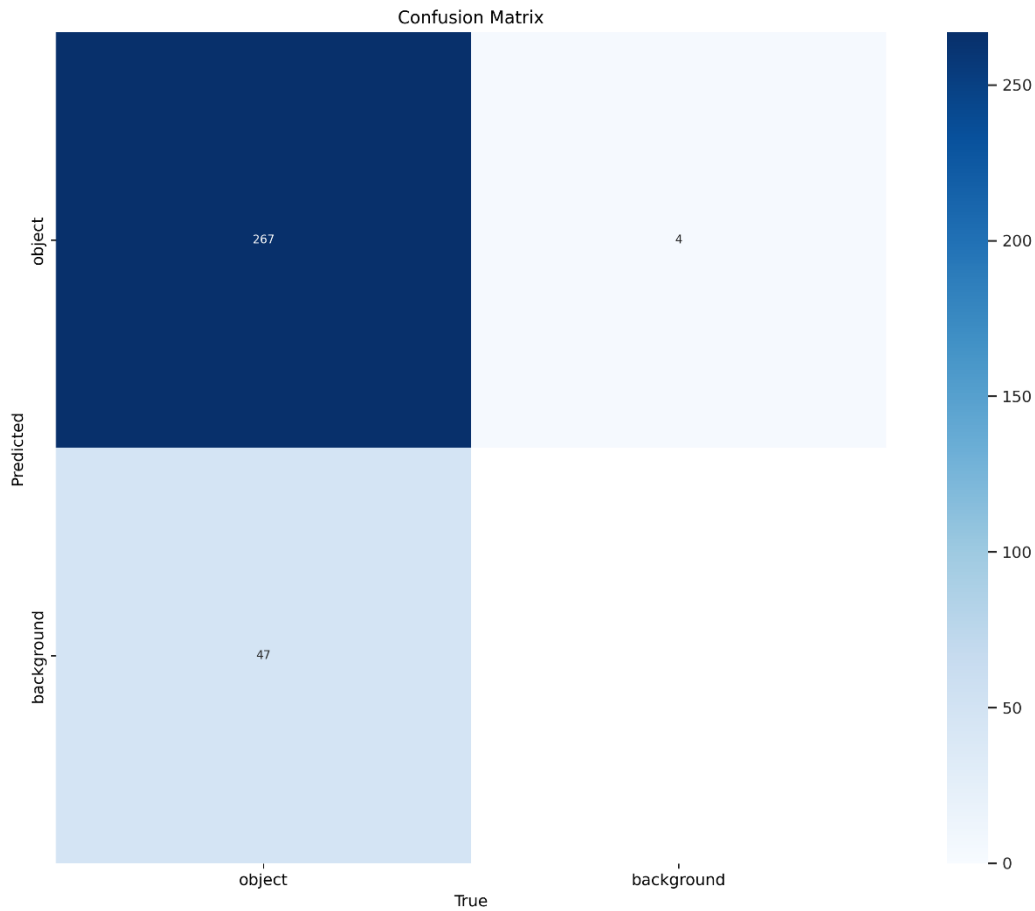


Imagem 57 – Matriz da Confusão Modelo de Objetos v2

Resultados:

- Classe "Object":
 - **TP:** 267 detecções corretas.
 - **FN:** 4 casos não detectados corretamente como "Object".
- Classe "Background":
 - **FP:** 47 casos foram incorretamente classificados como "Background".
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **sensibilidade alta** (recall) para a classe "Object", conseguindo identificar corretamente 267 casos, falhando somente em 4.
- A **precisão também é moderada**, já que ainda há falsos positivos (47), o que pode indicar que o modelo às vezes classifica erroneamente objetos como fundo.
- O **desempenho geral** do modelo é **razoável**, mas pode ser melhorado ajustando hiperparâmetros, refinando a estratégia de balanceamento de classes e aumentando a diversidade do dataset para reduzir falsos positivos e falsos negativos

Visualização dos Resultados

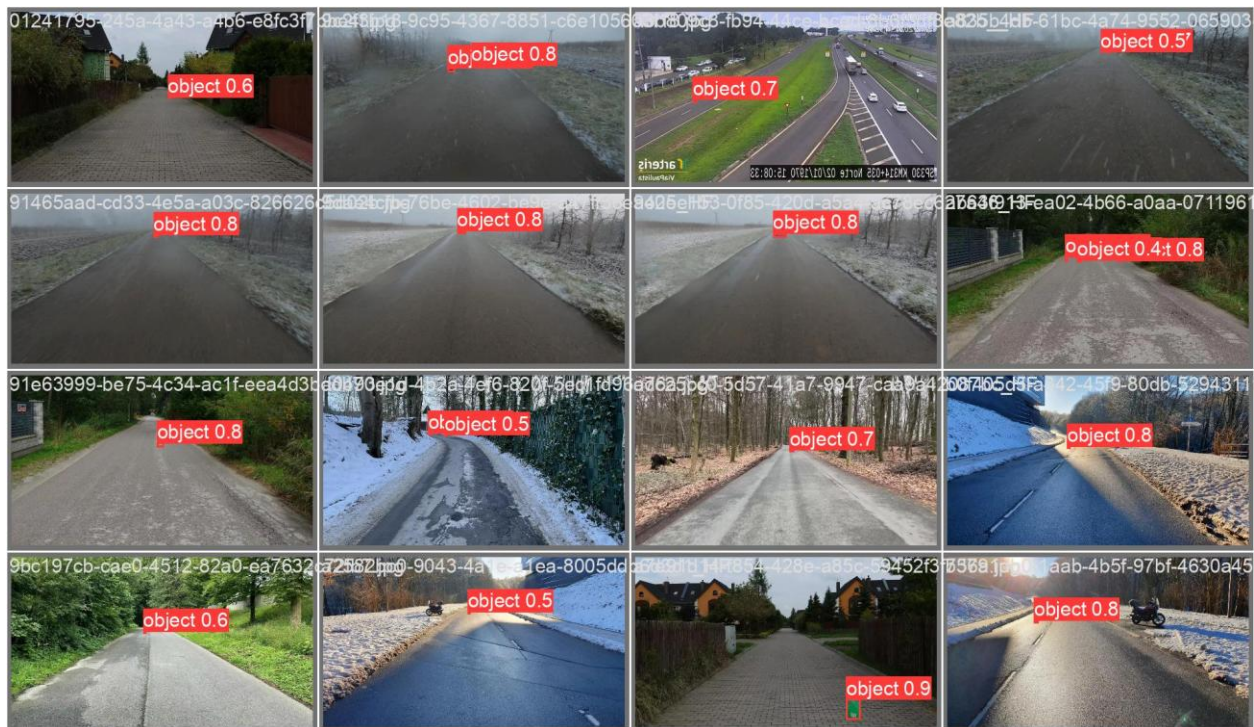


Imagem 58 – Resultados Iniciais Modelo de Objetos v2

5.8 Conclusão dos modelos

Os resultados obtidos com o treinamento dos modelos indicam avanços significativos no desenvolvimento de soluções automatizadas para cenários críticos. Cada modelo demonstrou capacidades relevantes no reconhecimento de objetos em seus respectivos domínios, com desempenhos variados em função das condições dos dados e das particularidades do cenário.

Apesar das conquistas, desafios como falsos positivos, falsos negativos e limitações de generalização ainda precisam ser abordados. Esses pontos oferecem oportunidades claras para aprimoramentos futuros, seja através da expansão e refinamento dos conjuntos de dados, seja por

meio da experimentação com arquiteturas alternativas ou ajustes nos hiperparâmetros de treinamento.

Os resultados apresentados neste documento formam a base para decisões estratégicas sobre a aplicação prática dos modelos e indicam caminhos promissores para o avanço de sistemas inteligentes de monitoramento. Assim, o trabalho reforça o compromisso com a inovação e a eficiência na integração de tecnologias de visão computacional em sistemas reais.

6. CONFIGURAÇÃO EDGE COMPUTING EM SERVIDORES TEMPORÁRIOS

Com o objetivo de validar a viabilidade e o desempenho de soluções em arquitetura descentralizada, foi realizada a configuração de um ambiente de **Edge Computing** utilizando **servidores temporários**. A iniciativa visa testar aplicações críticas em cenários próximos da borda, reduzindo latência e aumentando a resiliência dos serviços em ambientes com conectividade limitada.

A configuração contemplou os seguintes passos:

- Provisionamento de servidores temporários com recursos compatíveis para execução local de aplicações (processador, memória e armazenamento otimizados);
- Instalação de sistema operacional leve e customização do ambiente para operação autônoma e eficiente;
- Implantação de contêineres e serviços por meio de ferramentas leves de orquestração, como K3s, quando necessário;
- Estabelecimento de rotas seguras de comunicação entre os dispositivos de borda e o backend central na nuvem ou data center;
- Configuração de monitoramento local, com coleta de métricas e logs para análise de desempenho e estabilidade;
- Simulações de desconexão e reconexão para validação da operação offline e sincronização posterior dos dados.

Durante os testes, o ambiente mostrou-se adequado para processar dados localmente com baixo tempo de resposta, atendendo aos requisitos de aplicações sensíveis à latência, como análise em tempo real, visão computacional e automações locais.

A infraestrutura temporária permanecerá em operação por um período controlado, permitindo ajustes e coleta de dados adicionais para embasar futuras decisões sobre a adoção definitiva de arquitetura de borda em ambientes produtivos.

7. APRIMORAMENTO DA APLICAÇÃO DE CONTAGEM DE VEÍCULOS

7.1 Modelo de Contagem Versão 2.0

Conjunto de Dados

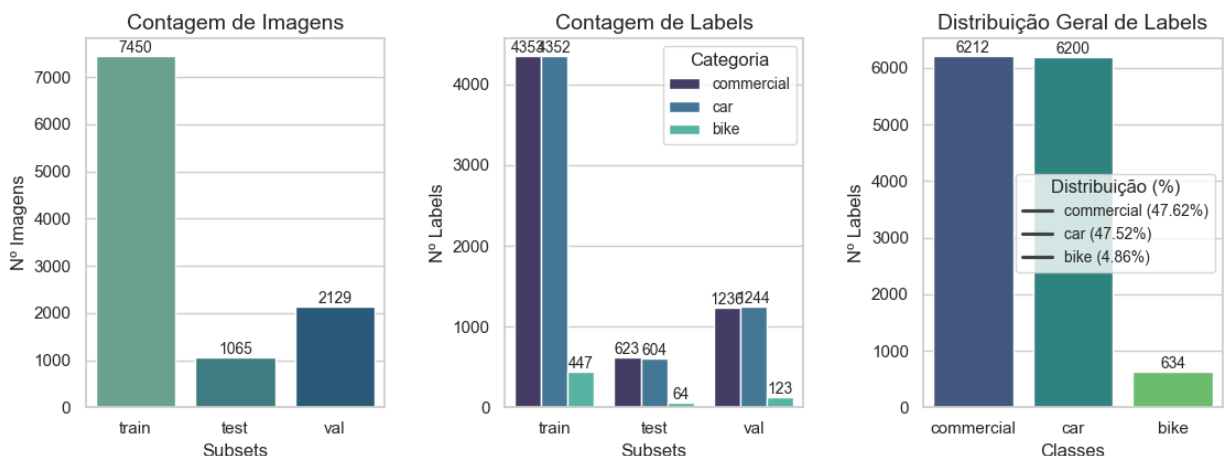


Imagem 59 – Distribuição Dataset Contagem v2

- Tamanho do Dataset: Total de 10.644 de imagens, e 13.046 de labels.
- Distribuição dos Dados:
 - Train: 7.450 imagens (commercial: 4.353 labels, car: 4.352 labels, bike: 447 labels)
 - Test: 1.065 imagens (commercial: 623 labels, car: 604 labels, bike: 64 labels)
 - Val: 2.129 imagens (commercial: 1.236 labels, car: 1.244 labels, bike: 123 labels)
- Data Augmentation: Nenhum.

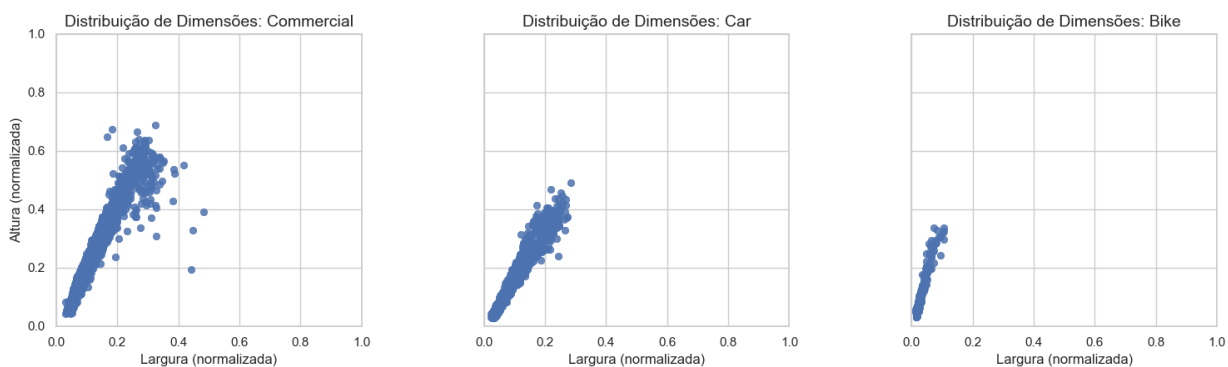


Imagem 60 – Distribuição Bouding Boxes Modelo de Contagem v2

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas, sugerindo que os objetos dessa classe ocupam uma área de eixo Y maior em relação as outras classes.
- Distribuição da Classe 'car': As bounding boxes para a classe car são mais largas, com uma ampla variação de tamanhos, indicando que os carros ocupam uma área significativa nas imagens, variando em tamanho e proporção. Carros se assemelham com comerciais pois o ângulo de visão da câmera é direto e linear.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas, nem tão largas quanto os carros, nem tão altas quanto as comerciais.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

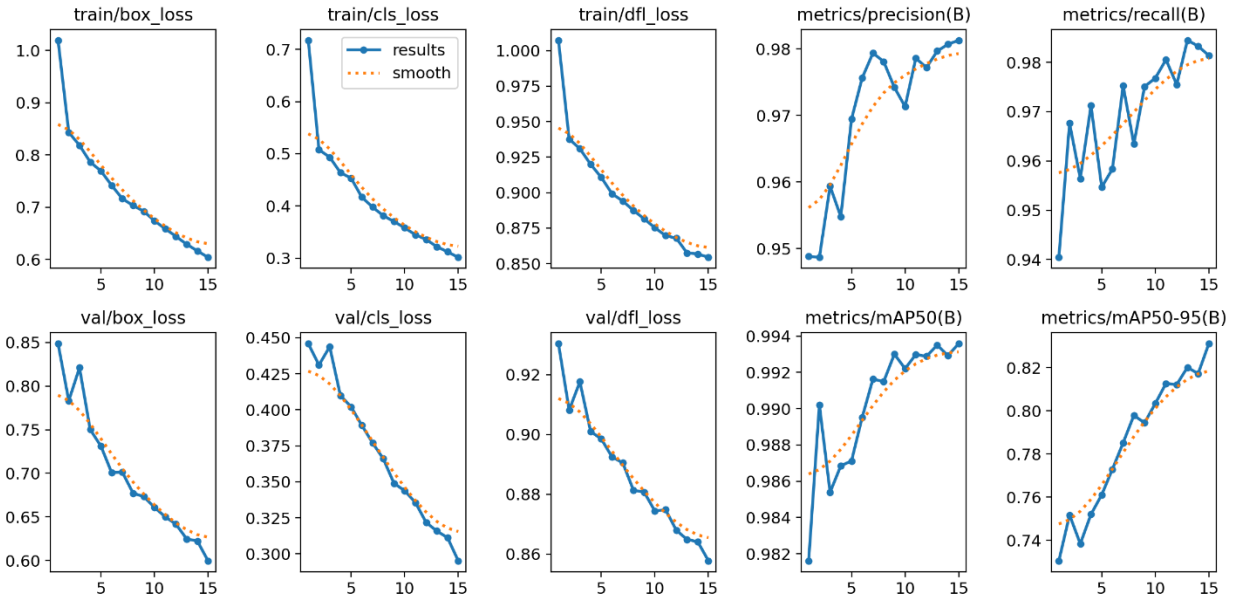


Imagem 61 – Métricas Gerais Modelo de Contagem v2

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.98**, demonstrando baixa taxa de falsos positivos.
 - O recall cresce até **0.99**, mostrando que a maioria dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.99**, destacando bom desempenho em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.83**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

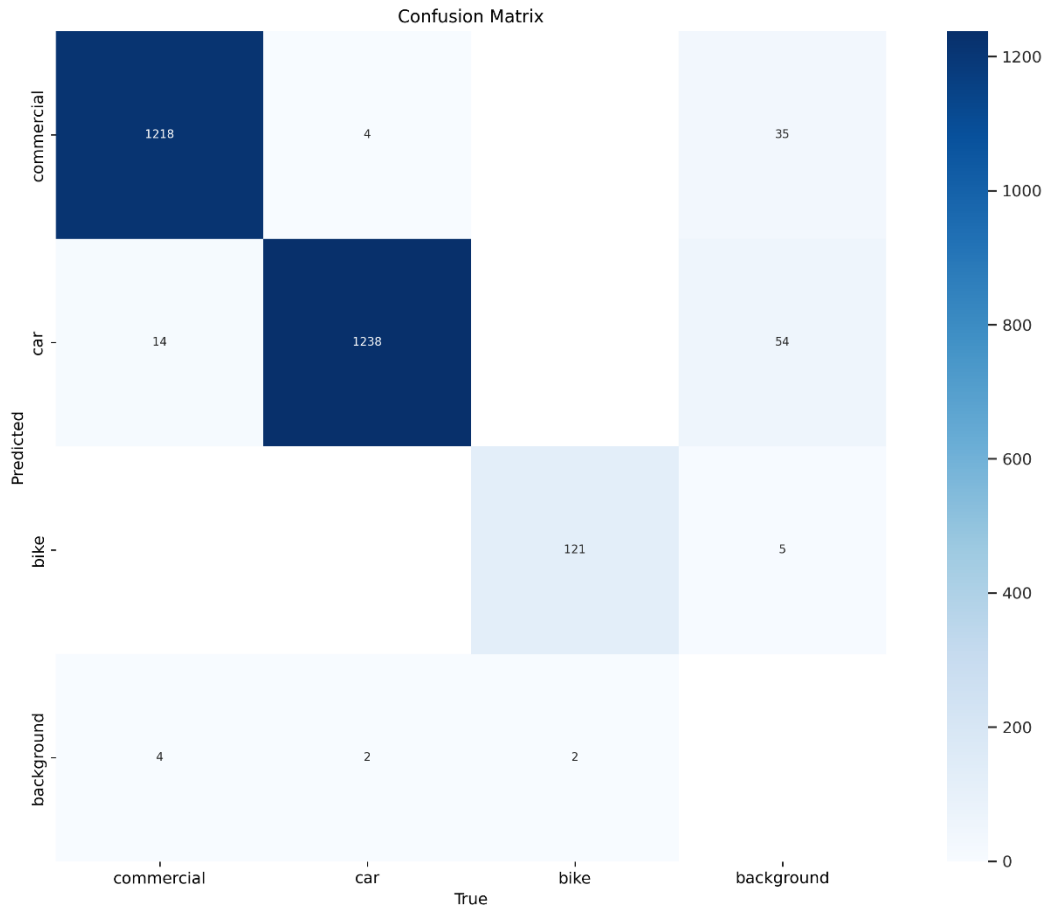


Imagem 62 – Matriz da Confusão Modelo de Contagem v2

Resultados:

- Classe "Commercial":
 - **TP:** 1218 detecções corretas.
 - **FN:** 39 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 1238 detecções corretas.
 - **FN:** 68 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 121 detecções corretas.
 - **FN:** 5 casos não detectados corretamente como " Bike".
- Classe "Background":

- **FP:** 8 casos foram incorretamente classificados entre os veículos.
- **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para a classe "Commercial", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**.

Visualização dos Resultados

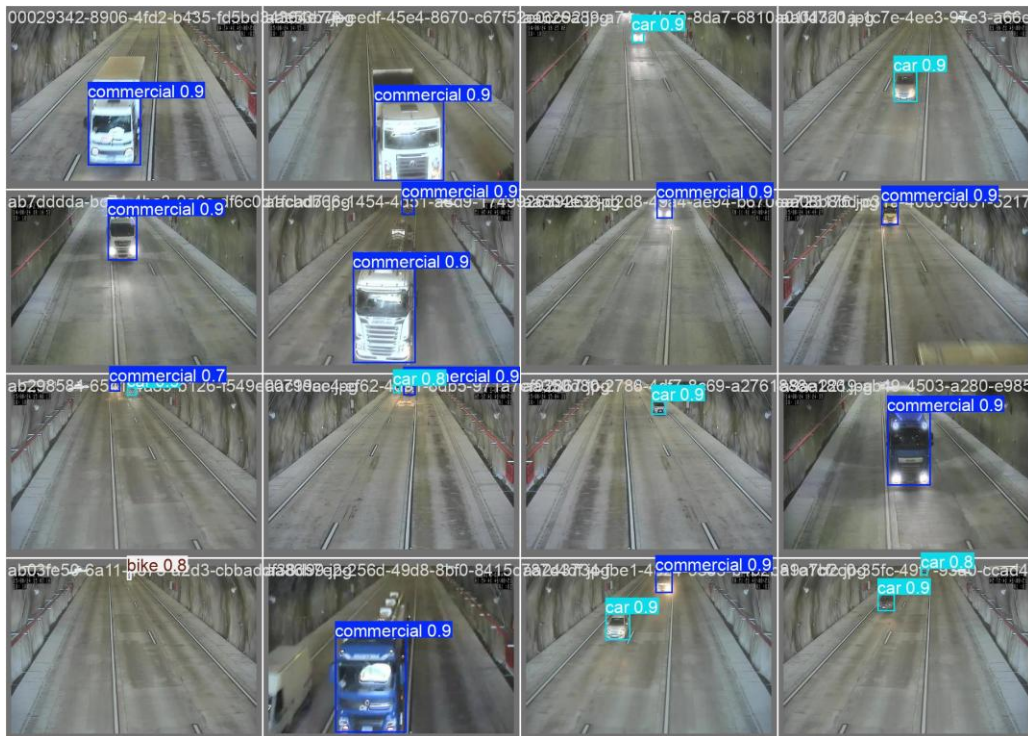


Imagem 63 – Resultados Iniciais Modelo de Contagem v2

7.2 Modelo de Contagem Versão 3.0

Conjunto de Dados

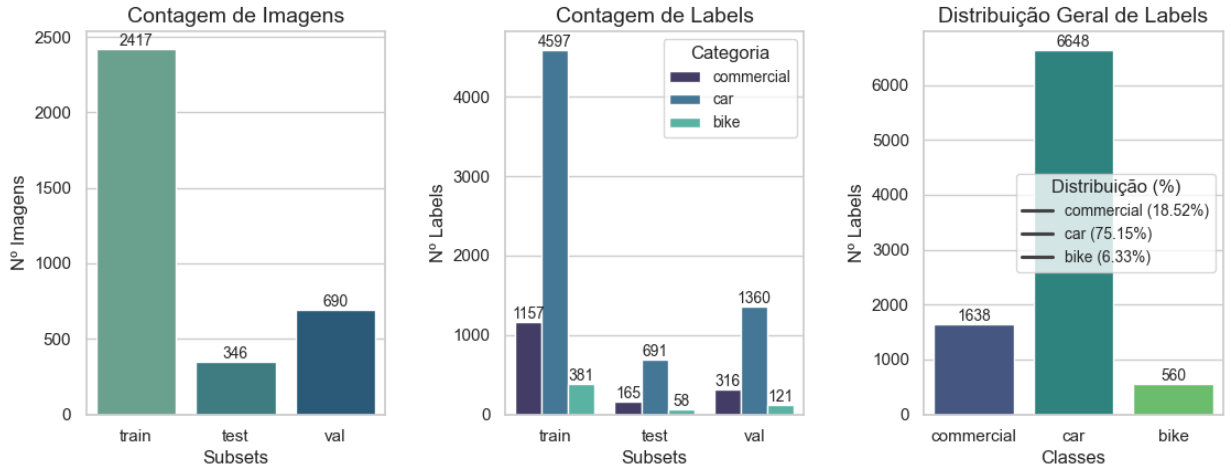
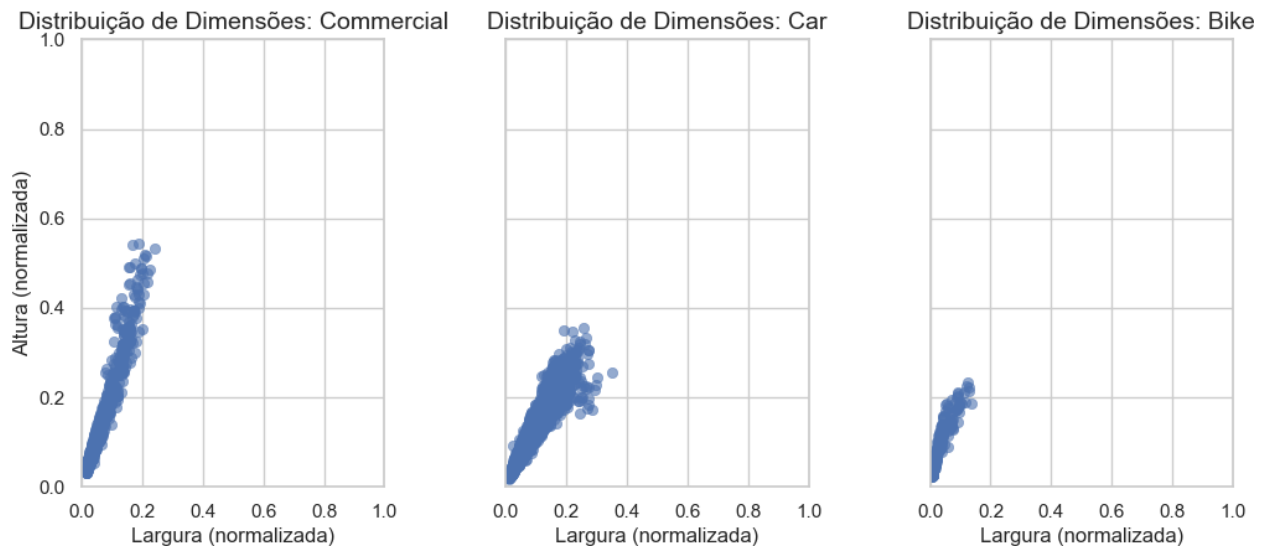


Imagem 64 – Distribuição Dataset Contagem v3

- Tamanho do Dataset: Total de 3.453 de imagens, e 8.846 de labels.
- Distribuição dos Dados:
 - Train: 2.417 imagens (commercial: 1.157 labels, car: 4.597 labels, bike: 165 labels)
 - Test: 346 imagens (commercial: 381 labels, car: 691 labels, bike: 58 labels)
 - Val: 690 imagens (commercial: 316 labels, car: 1.360 labels, bike: 121 labels)
- Data Augmentation: Horizontal Flip.



Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas, sugerindo que os objetos dessa classe ocupam uma área de eixo Y maior em relação as outras classes.
- Distribuição da Classe 'car': As bounding boxes para a classe car são mais largas, com uma ampla variação de tamanhos, indicando que os carros ocupam uma área significativa nas imagens, variando em tamanho e proporção.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas, nem tão largas quanto os carros, nem tão altas quanto as comerciais.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 20, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

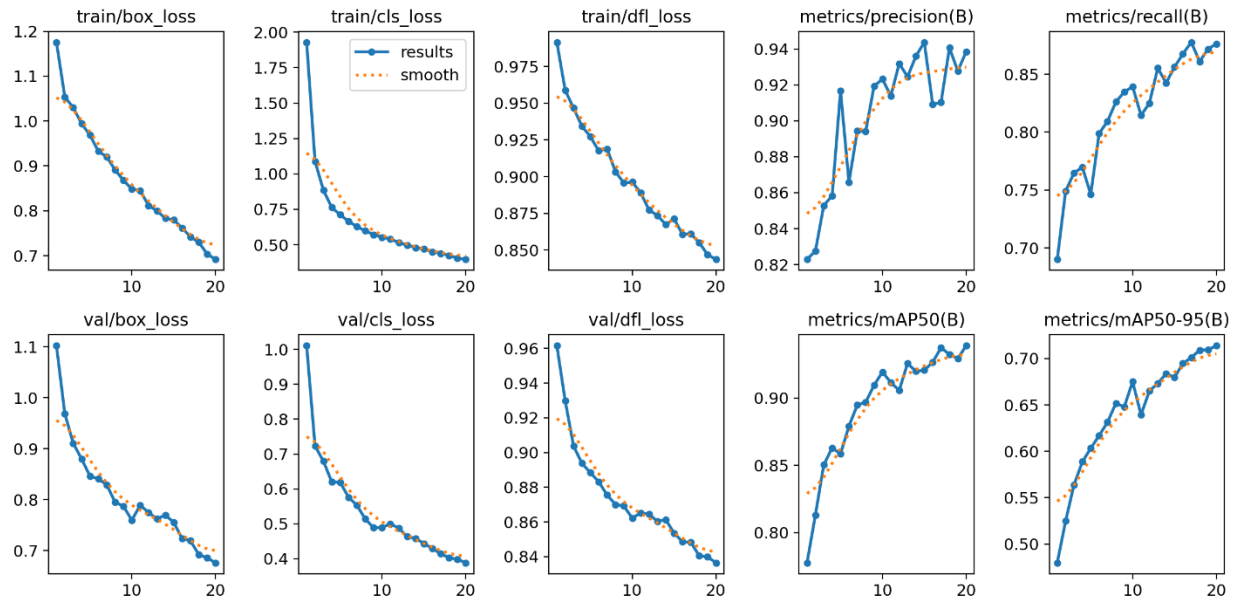


Imagem 66 – Métricas Gerais Modelo de Contagem v3

- Perdas de Treinamento e Validação:
 - As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.
- Precisão e Recall:
 - A precisão atinge cerca de **0.95**, demonstrando baixa taxa de falsos positivos.
 - O recall cresce até **0.88**, mostrando que a maioria dos objetos foi corretamente detectada.
- mAP:
 - O **mAP@50** chega a aproximadamente **0.95**, destacando bom desempenho em detecções mais amplas.
 - O **mAP@50-95** alcança cerca de **0.72**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

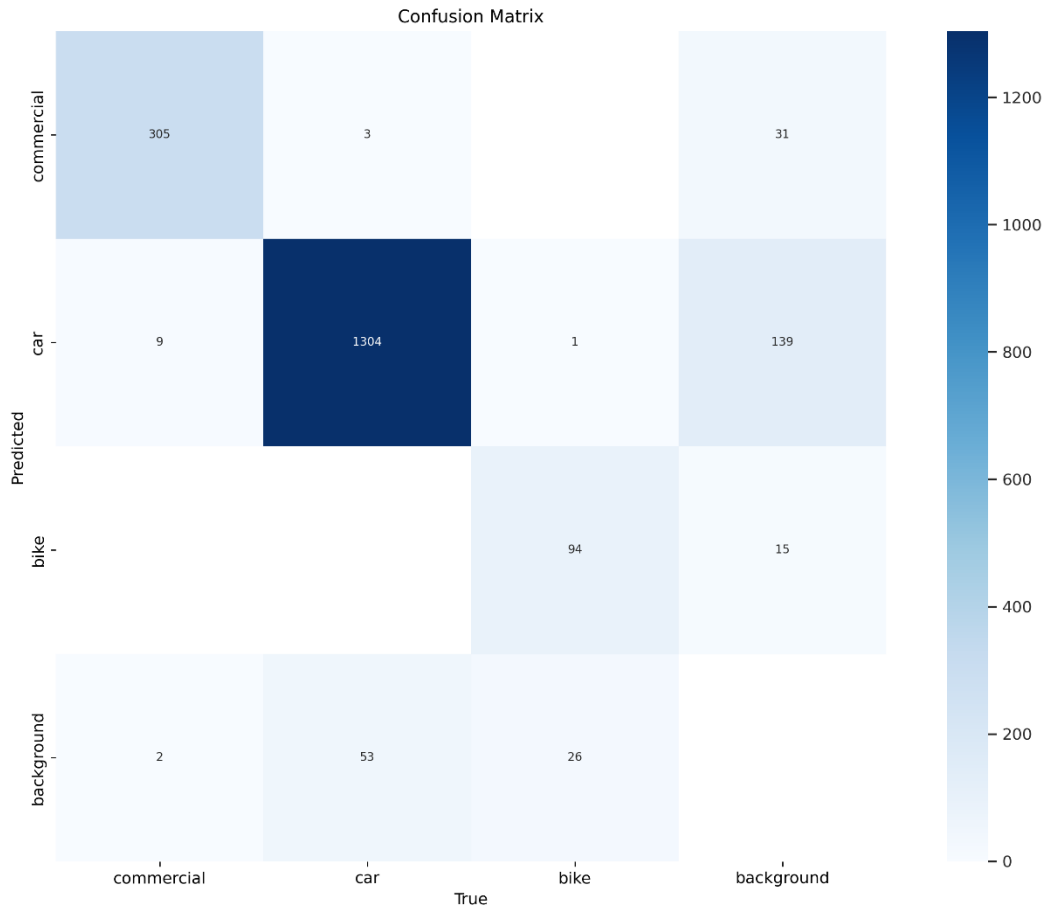


Imagem 67 – Matriz da Confusão Modelo de Contagem v3

Resultados:

- Classe "Commercial":
 - **TP:** 305 detecções corretas.
 - **FN:** 34 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 1304 detecções corretas.
 - **FN:** 149 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 94 detecções corretas.
 - **FN:** 15 casos não detectados corretamente como " Bike".
- Classe "Background":

- **FP:** 81 casos foram incorretamente classificados entre os veículos.
- **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para a classe "Commercial", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para a classe "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados

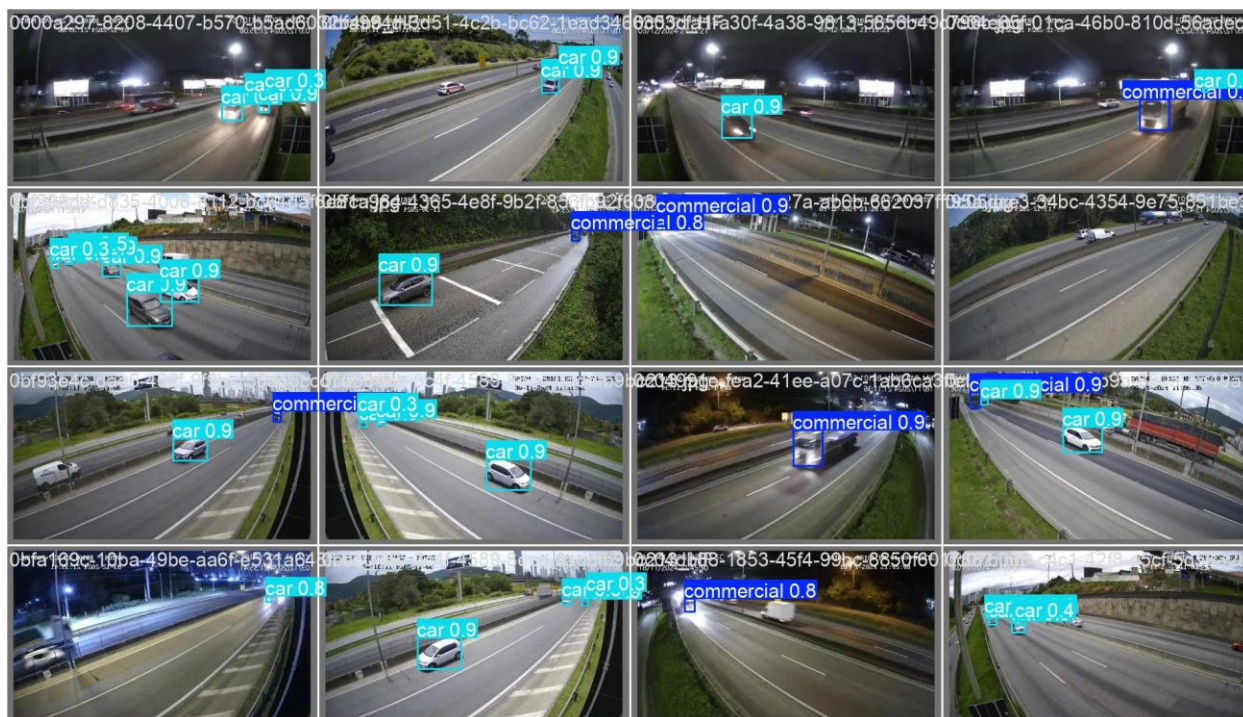


Imagem 68 – Resultados Iniciais Modelo de Contagem v3

8. CONFIGURAÇÃO KUBERNETES EDGE EM SERVIDORES TEMPORÁRIOS

Foi realizada a configuração de um ambiente **Kubernetes Edge** utilizando servidores temporários, com o objetivo de validar a execução distribuída de aplicações em infraestrutura de borda (edge computing).

O processo envolveu a instalação e configuração de nós leves (edge nodes) com recursos limitados, preparados para operar em condições de conectividade instável e com menor capacidade de processamento. A orquestração foi feita a partir de um **cluster centralizado**, com gerenciamento híbrido entre o plano de controle (control plane) hospedado na nuvem e os nós edge alocados nos servidores locais.

As principais etapas incluíram:

- Provisão dos servidores temporários com sistema operacional Linux otimizado;
- Instalação de componentes essenciais como kubelet, kube-proxy e container runtime;
- Registro seguro dos nós edge no cluster Kubernetes central por meio de certificados e tunelamento;
- Configuração de políticas de tolerância e afinidade para agendamento de workloads apenas nos nós edge;
- Implementação de monitoramento básico com Prometheus e coleta de logs via Fluentd, adaptados ao ambiente descentralizado.

O ambiente foi utilizado para testes de baixa latência e operação offline/resiliente, com validação do comportamento das aplicações em casos de perda de conectividade parcial. A configuração demonstrou viabilidade para cenários de computação na borda, especialmente em contextos em que o processamento local é necessário para reduzir carga na rede e tempo de resposta.

Este ambiente será mantido temporariamente para observação, e as lições aprendidas serão utilizadas para definir diretrizes de futuras implantações permanentes.

9. AQUISIÇÃO DE EQUIPAMENTOS E PREPARAÇÃO DE INFRAESTRUTURA NA RODOVIA PARA PROCESSAMENTO DO MODELO DE CONTAGEM DE VEÍCULOS

Em dezembro de 2024 foi contratada a prestação de serviço de instalação e montagem das câmeras que seriam utilizadas para captura de imagens e processamento do modelo de contagem de veículos. Essas câmeras já constavam em estoque da concessionária.

Em janeiro de 2025 foi efetuada a reposição destas câmeras e demais equipamentos, onde foram adquiridas 8 câmeras 6 MP HIKVISION, 7 SWITCH INDUSTRIAL POE PLANET, 20 SEALTUBO INDELFLEX e 1 CABO LAN 4P.

10. AQUISIÇÃO DOS SERVIDORES PARA SUSTENTAÇÃO DOS MODELOS

Este tópico tem como finalidade registrar, de forma técnica e detalhada, todas as etapas envolvidas na instalação, configuração e entrega em produção do servidor Lenovo ThinkAgile MX450 Edge IS, adquirido pela Arteris S/A conforme proposta comercial BRDLS005909120. O projeto contempla a implementação de um servidor de alto desempenho, com capacidade para suportar cargas de trabalho intensivas, virtualização e inteligência artificial, integrando componentes de última geração e licenciamento especializado.

11. CONFIGURAÇÃO E PREPARO DE INFRAESTRUTURA DE TI ARTERIS

Instalação Física

A instalação física do servidor ThinkAgile MX450 Edge IS foi realizada com precisão e seguindo as diretrizes técnicas da Lenovo. O equipamento foi fixado em trilhos deslizantes e conectado à rede elétrica com redundância N+N, utilizando fontes de alimentação hot-swap de 1100W. As conexões de rede foram estabelecidas por meio de adaptadores Mellanox ConnectX-6 Lx, garantindo conectividade de 10/25GbE.

Os módulos de memória TruDDR4 3200MHz foram instalados conforme especificado, totalizando 128GB. Os SSDs NVMe PCIe 4.0 foram alocados nos slots apropriados, com configuração de armazenamento em RAID 1 para os discos M.2 SATA, e armazenamento principal em NVMe U.3 de 1.92TB. As GPUs NVIDIA A2 foram integradas ao servidor, com suporte físico e térmico adequado, e conectadas aos transceivers SFP+ para comunicação de alta velocidade.

Configuração Lógica e Software

Concluída a instalação física, iniciou-se a configuração lógica do servidor. O sistema operacional Windows Server 2022 Datacenter foi instalado, com ativação da licença principal e das licenças adicionais conforme especificado na proposta. A pilha de software Microsoft Azure Stack HCI versão 23H2 foi configurada para operação em modo de máximo desempenho, habilitando recursos de virtualização, gerenciamento de clusters e integração com serviços híbridos.

O Lenovo XClarity Controller (XCC2 Platinum) foi ativado, permitindo o gerenciamento remoto e monitoramento contínuo do servidor. O Lenovo XClarity Pro foi instalado e configurado para o endpoint, com suporte estendido de 3 e 5 anos conforme licenciamento contratado.

Adicionalmente, foram ativadas as licenças NVIDIA AI Enterprise Subscription para cada GPU A2, com validade de 5 anos, conforme informado pela Lenovo. Essas licenças são essenciais para habilitar os recursos de inteligência artificial e machine learning embarcados nas GPUs.

Testes de Validação

Após a configuração dos sistemas, foram realizados testes de validação para garantir a integridade e o desempenho do servidor. Os testes incluíram:

- Inicialização completa e verificação de POST.
- Validação da conectividade de rede em todas as interfaces.
- Testes de leitura e escrita nos SSDs NVMe.
- Benchmark de desempenho das GPUs NVIDIA A2.
- Verificação de redundância elétrica e térmica.
- Testes de RAID e recuperação de dados.
- Monitoramento via XClarity Controller.

Todos os testes foram documentados e aprovados, garantindo que o servidor estava apto para operação em ambiente de produção.

Entrega em Produção

Com todas as etapas concluídas, o servidor ThinkAgile MX450 Edge IS foi oficialmente entregue ao ambiente de produção da Arteris S/A. A documentação técnica foi disponibilizada, incluindo logs de instalação, registros de licenciamento, relatórios de testes e manuais de operação. O aceite formal foi realizado pelo cliente, com assinatura dos responsáveis técnicos.

12. APRIMORAMENTO DA APLICAÇÃO DE CONTAGEM DE VEÍCULOS

12.1 Modelo de Contagem Versão 4.0

Conjunto de Dados

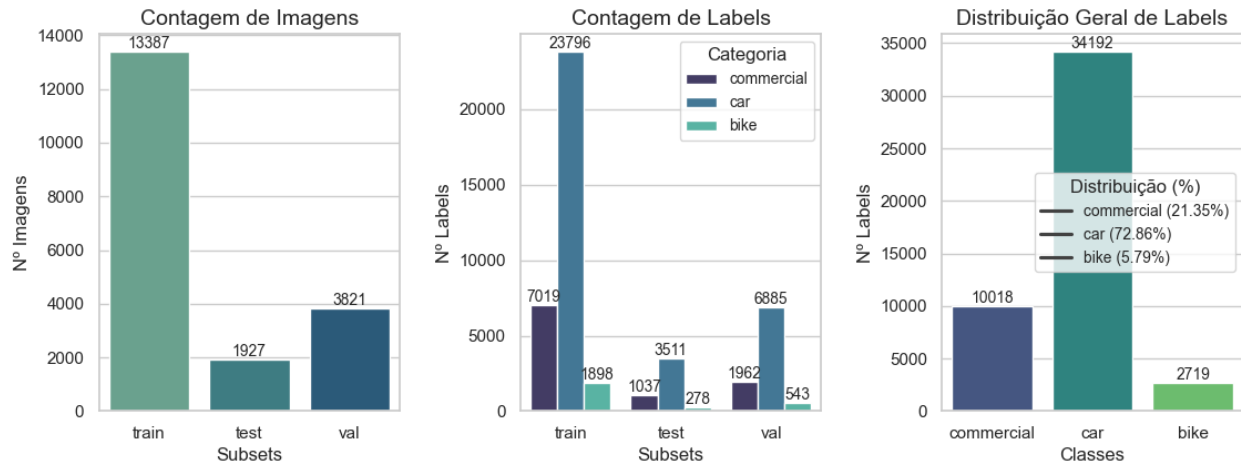


Imagem 69 – Distribuição Dataset Contagem v4

- Tamanho do Dataset: Total de 19.135 de imagens, e 46.929 de labels.
- Distribuição dos Dados:
 - Train: 13.387 imagens (commercial: 7.019 labels, car: 23.796 labels, bike: 1.898 labels)
 - Test: 1.927 imagens (commercial: 1.037 labels, car: 3.511 labels, bike: 278 labels)
 - Val: 3.821 imagens (commercial: 1.962 labels, car: 6.885 labels, bike: 543 labels)
- Data Augmentation: Horizontal Flip.

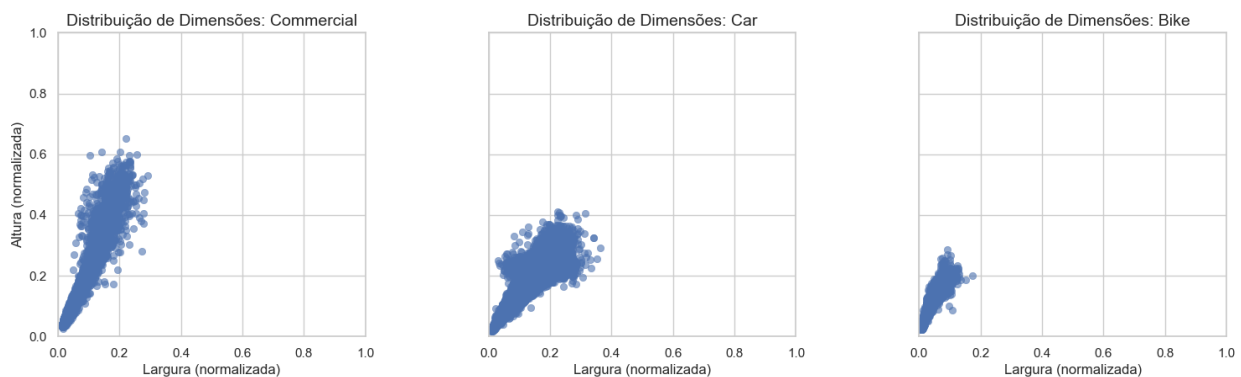


Imagem 70 – Distribuição Bouding Boxes Modelo de Contagem v4

Quanto mais próximas da formação de uma diagonal perfeita, melhor é a distribuição de objetos por imagens no dataset. Vale ressaltar alguns detalhes, como:

- Distribuição da Classe 'commercial': As bounding boxes para a classe commercial estão mais concentradas com dimensões relativamente altas, sugerindo que os objetos dessa classe ocupam uma área de eixo Y maior em relação as outras classes.
- Distribuição da Classe 'car': As bounding boxes para a classe car são mais largas, com uma ampla variação de tamanhos, indicando que os carros ocupam uma área significativa nas imagens, variando em tamanho e proporção.
- Distribuição da Classe 'bike': A classe bike apresenta uma distribuição de bounding boxes com dimensões pequenas, nem tão largas quanto os carros, nem tão altas quanto as comerciais.

Configuração do Treinamento

- Parâmetros do Treinamento:
 - Framework utilizado Ultralytics YOLOv8.
 - Epochs: 15, Batch: 8, Image Size: 640.
- Hardware:
 - Azure Standard_NC8as_T4_v3 (8 cores, 56 GB RAM, 352 GB disk).

Resultados

- Métricas Gerais:

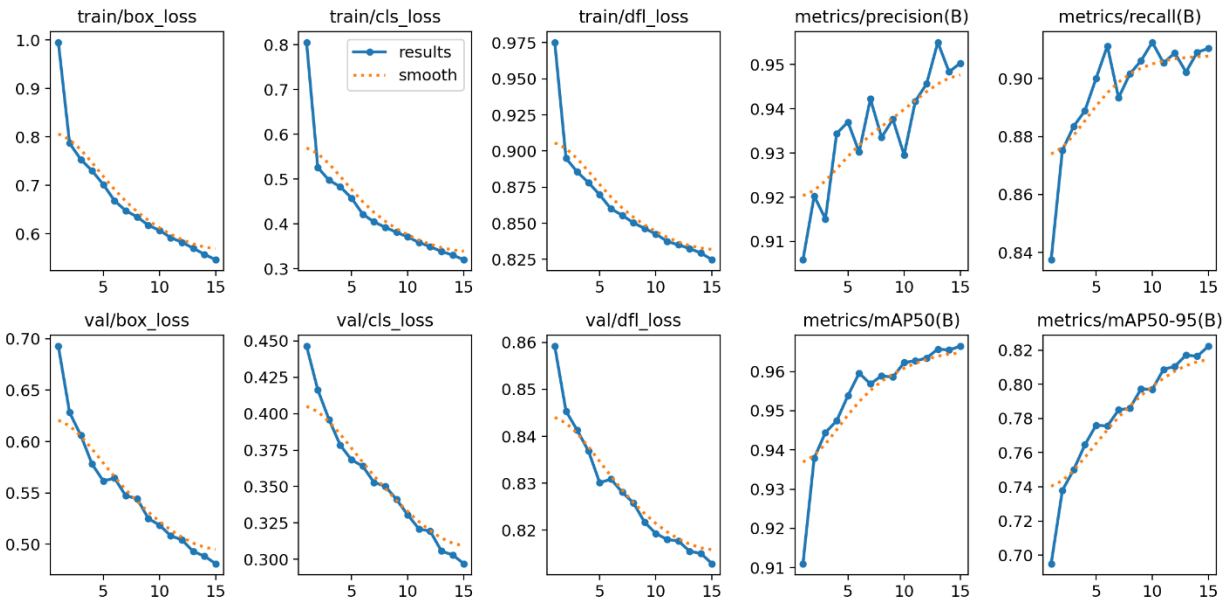


Imagem 71 – Métricas Gerais Modelo de Contagem v4

- Perdas de Treinamento e Validação:

- As perdas (**box_loss**, **cls_loss**, **dfl_loss**) diminuem consistentemente tanto no treinamento quanto na validação, indicando boa convergência do modelo.

- Precisão e Recall:

- A precisão atinge cerca de **0.96**, demonstrando baixa taxa de falsos positivos.
- O recall cresce até **0.91**, mostrando que a maioria dos objetos foi corretamente detectada.

- mAP:

- O **mAP@50** chega a aproximadamente **0.97**, destacando bom desempenho em detecções mais amplas.
- O **mAP@50-95** alcança cerca de **0.82**, sugerindo margem para melhorias em detecções mais rigorosas.

Matriz da confusão:

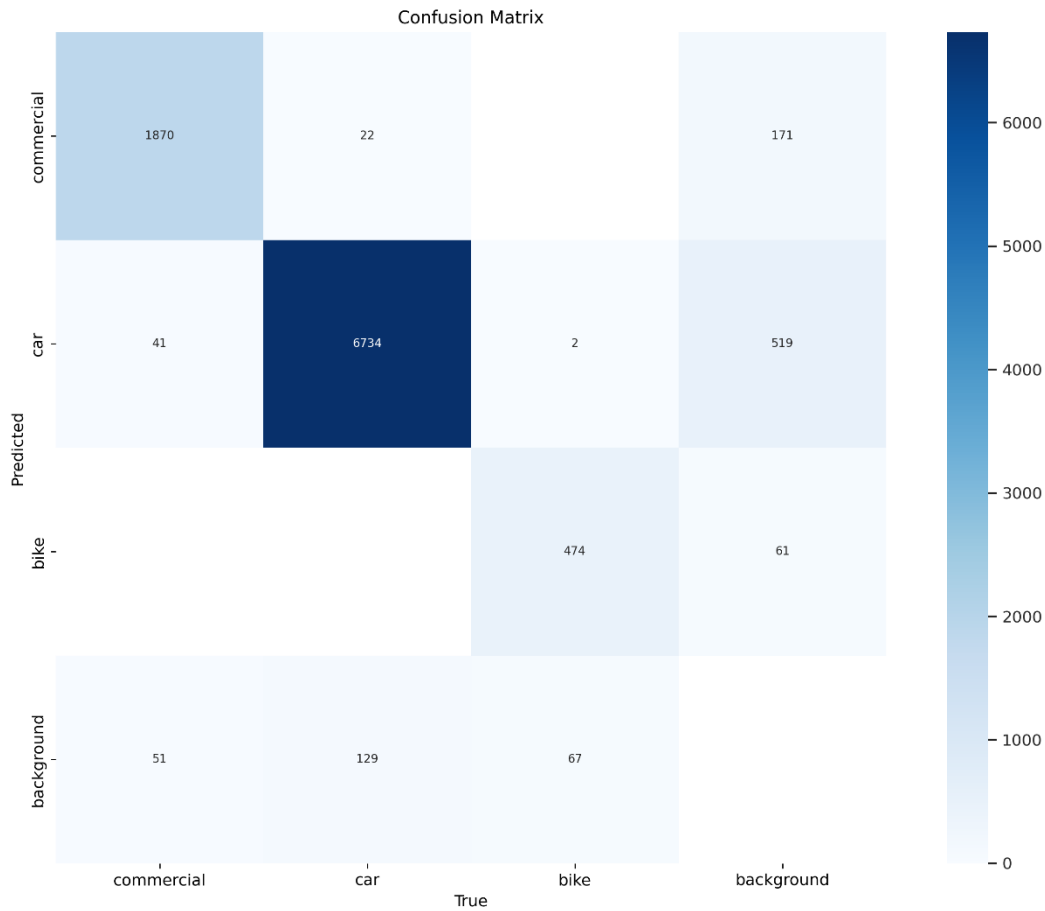


Imagem 72 – Matriz da Confusão Modelo de Contagem v4

Resultados:

- Classe "Commercial":
 - **TP:** 1.870 detecções corretas.
 - **FN:** 193 casos não detectados corretamente como " Commercial".
- Classe "Car":
 - **TP:** 6.734 detecções corretas.
 - **FN:** 562 casos não detectados corretamente como " Car".
- Classe "Bike":
 - **TP:** 474 detecções corretas.
 - **FN:** 61 casos não detectados corretamente como " Bike".

- Classe "Background":
 - **FP:** 247 casos foram incorretamente classificados entre os veículos.
 - **TN:** O restante das instâncias foi corretamente identificado como "Background".

Resumo Geral:

- O modelo exibe uma **alta sensibilidade (recall)** para a maioria das classes, especialmente para a classe "Commercial", o que significa que ele consegue identificar corretamente a maioria dos casos dentro dessas classes.
- A **precisão** é ligeiramente inferior, o que sugere um número moderado de **falsos positivos (FP)**, especialmente na classe "Background", onde há uma sobrecarga de erros de classificação entre veículos.
- Em geral, o desempenho do modelo é robusto, mas melhorias podem ser alcançadas com um foco específico na **redução de falsos positivos (FP)**, especialmente para a classe "Background", a fim de aumentar a precisão e a confiabilidade das classificações.

Visualização dos Resultados



Imagem 73 – Resultados Iniciais Modelo de Contagem v4

13. CRIAÇÃO ARQUITETURA INGESTÃO DE DADOS

13.1 Engenharia

Para esta solução, foram desenvolvidos fluxos que englobam a extração de dados capturados por câmeras nas rodovias, com o objetivo de realizar a contagem de veículos. Essas informações são armazenadas no Data Lake da Arteris e, em seguida, passam por etapas de processamento e tratamento no Azure Databricks. O resultado é disponibilizado em dashboards interativos desenvolvidos no Power BI.

O principal objetivo da solução é processar os dados provenientes diretamente dos arquivos JSON disponibilizados no Blob Umbrella, assegurando o armazenamento seguro e a aplicação consistente das regras de negócio. Isso garante maior precisão nos indicadores gerados, proporcionando insights mais confiáveis para a tomada de decisões estratégicas.

13.2 Data Lake Storage

O armazenamento no Data Lake foi realizado seguindo os padrões estabelecidos para os ambientes desenvolvimento e produção, com a estruturação dos dados em camadas: bronze, silver e gold. Essa abordagem garante a organização, escalabilidade e consistência no processamento e consumo dos dados.

Camada bronze:

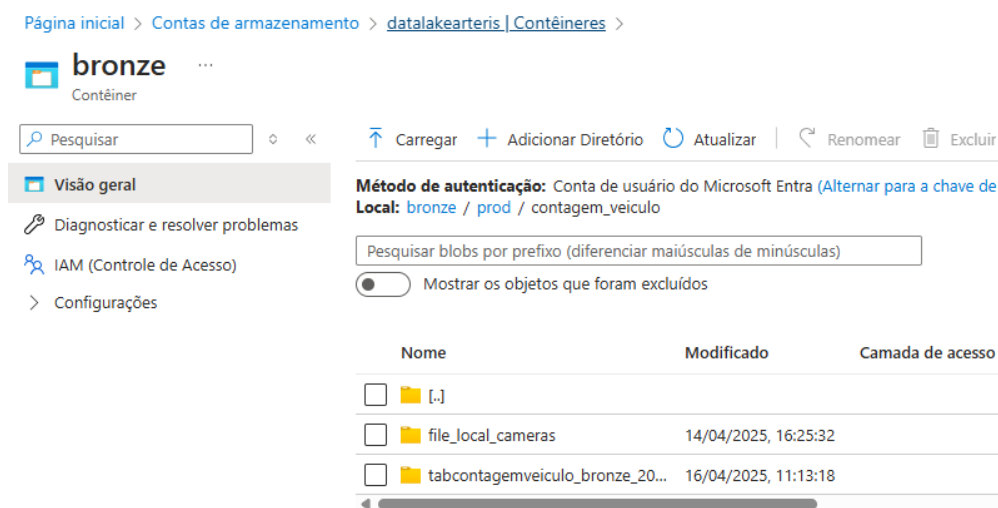


Imagem 74 – Armazenamento na Camada Bronze

Camada silver:

[Página inicial](#) > [Contas de armazenamento](#) > [datalakearteris | Contêineres](#) >

silver Contêiner

Pesquisar

Carregar + Adicionar Diretório Atualizar Renomear Excluir

Método de autenticação: Chave de acesso (Alternar para conta de usuário do Microsoft Local: silver / prod / operacao / tabcontagemveiculo)

Pesquisar blobs por prefixo (diferenciar maiúsculas de minúsculas)

Mostrar os objetos que foram excluídos

Nome	Modificado	Camada de acesso
<input type="checkbox"/> [.]		
<input type="checkbox"/> tabcontagemveiculo_silver	14/04/2025, 17:25:09	

Imagem 75 – Armazenamento na Camada Silver

Camada gold:

[Página inicial](#) > [Contas de armazenamento](#) > [datalakearteris | Contêineres](#) >

gold Contêiner

Pesquisar

Carregar + Adicionar Diretório Atualizar Renomear Excluir

Método de autenticação: Chave de acesso (Alternar para conta de usuário do Microsoft Entra Local: gold / prod / operacao / tabcontagemveiculo / obt_contagem_veiculo)

Pesquisar blobs por prefixo (diferenciar maiúsculas de minúsculas)

Nome
<input type="checkbox"/> [.]
<input type="checkbox"/> _delta_log
<input type="checkbox"/> part-00000-b49a63f8-d4d7-434d-84b0-a252c13f6dca-c000.snappy.parquet

Imagem 76 – Armazenamento na Camada Gold

13.3 Workspace Azure Databricks

O armazenamento dos notebooks no espaço de trabalho do Databricks foi realizado conforme os padrões definidos para os ambientes desenvolvimento e produção, alinhados à estruturação dos dados nas camadas: bronze, silver e gold. Essa organização facilita o gerenciamento, o desenvolvimento colaborativo e a separação clara das etapas de processamento de dados.

Camada bronze:

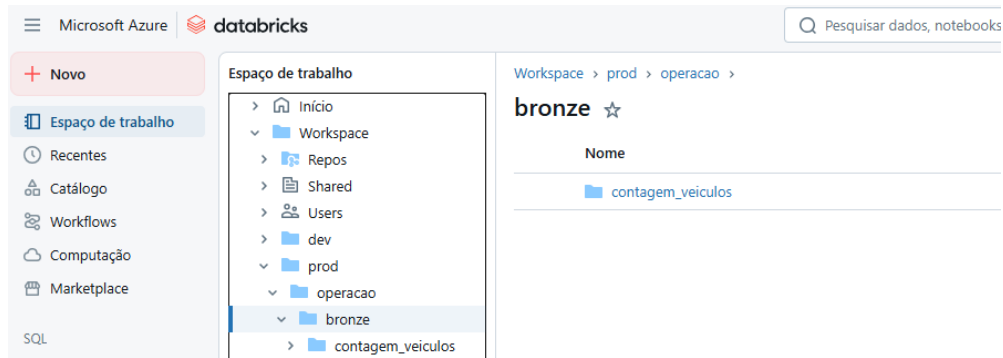


Imagem 77 – Notebooks da Camada Bronze

Camada silver:

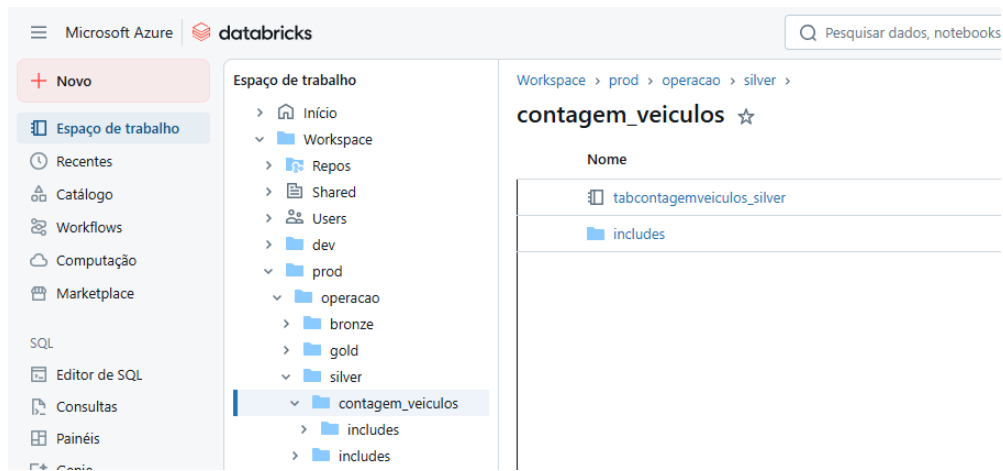


Imagem 78 – Notebooks da Camada Silver

Camada gold:

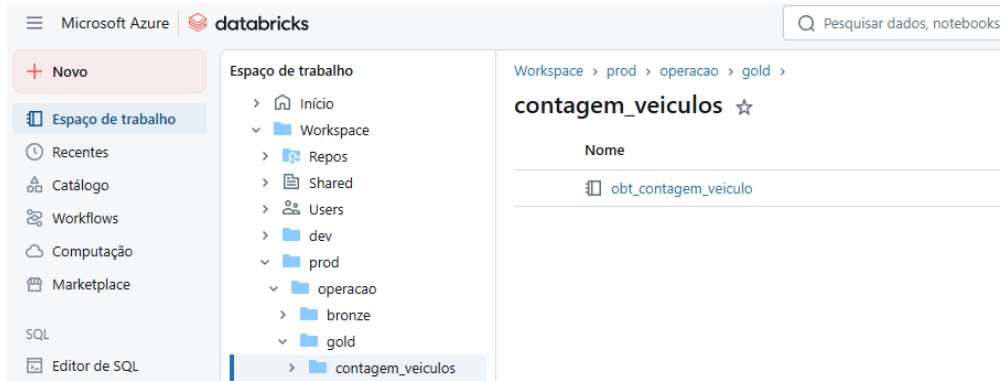


Imagem 79 – Notebooks da Camada Gold

13.4 Modelo de dados

O modelo de dados foi estruturado em uma arquitetura em camadas (Bronze, Silver e Gold), seguindo as boas práticas de organização e tratamento de dados em ambientes analíticos.

- **Camada Bronze:** Responsável pelo armazenamento dos dados brutos extraídos diretamente das câmeras, via arquivos JSON, com informações como horário do vídeo, contagem de veículos, velocidade média e categoria detectada.
- **Camada Silver:** Realiza o tratamento e enriquecimento dos dados, adicionando atributos como localização da câmera (latitude e longitude), status do equipamento e marcação temporal do processamento.
- **Camada Gold:** Consolida os dados prontos para consumo analítico, com estrutura otimizada para geração de indicadores, comparativos temporais e visualização em dashboards.

BRONZE		SILVER		GOLD	
<i>tabcontagemveiculo_bronze</i>		<i>tabcontagemveiculo_silver</i>		<i>obt_contagem_veiculo</i>	
CameraId	VARCHAR	sk_camera	VARCHAR	sk_camera	VARCHAR
VideoDate	DATETIME	dt_video	DATETIME	dt_video	DATETIME
VideoStartTime	TIME	ts_tempo_inicial_video	TIME	ts_tempo_inicial_video	TIME
VideoEndTime	TIME	ts_tempo_final_video	TIME	ts_tempo_final_video	TIME
VideoProcesstimeStamp	DATETIME	dt_processamento_video	DATETIME	dt_processamento_video	DATETIME
avgSpeed	INT	nb_media_velocidade	INT	nb_media_velocidade	INT
category	VARCHAR	ds_categoria	VARCHAR	ds_categoria	VARCHAR
lane	VARCHAR	nb_quantidade_veiculos	INT	nb_quantidade_veiculos	INT
vehicleCount	INT	ds_local_camera	VARCHAR	ds_local_camera	VARCHAR
Status	BIT	ds_latitude	VARCHAR	ds_latitude	VARCHAR
		ds_longitude	VARCHAR	ds_longitude	VARCHAR
		dt_updated_at_brt	DATETIME	dt_updated_at_brt	DATETIME
		dt_inicio_processo	DATETIME	dt_inicio_processo	DATETIME
		dt_fim_processo	DATETIME	dt_fim_processo	DATETIME
		st_status_camera	BIT	st_status_camera	BIT
				dt_inicial_video	DATETIME
				ds_periodo_dia	VARCHAR
				ts_diferenca_video	TIME

Imagem 80 – Transformações entre Camadas

13.5 Dicionário de dados

Abaixo está a descrição dos campos das tabelas, incluindo seus respectivos tipos de dados e explicação sobre cada um.

Dicionário da Tabela da Camada Bronze

Nome Campo	Tipo de dado	Descrição
CameraId	string	Identificador da câmera responsável pelo registro
VideoDate	datetime	Data da gravação do vídeo
VideoStartTime	time	Horário de início da gravação
VideoEndTime	time	Horário de final da gravação
VideoProcesstimeStamp	datetime	Data e hora em que o vídeo foi processado
avgSpeed	int	Velocidade média dos veículos
category	string	Categoria do veículo
lane	string	Número da faixa da rodovia
vehicleCount	int	Total de veículos detectados
Status	bit	Situação de disponibilidade da câmera

Dicionário da Tabela da Camada Silver

Nome Campo	Tipo de dado	Descrição
sk_camera	string	Identificador da câmera responsável pelo registro
dt_video	datetime	Data da gravação do vídeo
ts_tempo_inicial_video	time	Horário de início da gravação
ts_tempo_final_video	time	Horário de final da gravação
dt_processamento_video	datetime	Data e hora em que o vídeo foi processado
dt_updated_at_brt	datetime	Data de extração dos dados da API para o Lake

nb_media_velocidade	int	Velocidade média dos veículos
ds_faixa	string	Número da faixa da rodovia
ds_categoria	string	Categoria do veículo identificado
nb_quantidade_veiculos	int	Total de veículos detectados na gravação
ds_local_camera	string	Localização da instalação da câmera
ds_latITUDE	string	Latitude do local onde a câmera está instalada
ds_longitude	string	Longitude do local onde a câmera está instalada
dt_inicio_processo	datetime	Data e hora início da execução do processo
dt_fim_processo	datetime	Data e hora fim da execução do processo
st_status_camera	bit	Situação de disponibilidade da câmera

Dicionário da Tabela da Camada Gold

Nome Campo	Tipo de dado	Descrição
sk_camera	string	Identificador da câmera responsável pelo registro
dt_video	datetime	Data da gravação do vídeo
ts_tempo_inicial_video	time	Horário de início da gravação
ts_tempo_final_video	time	Horário de final da gravação
dt_processamento_video	datetime	Data e hora em que o vídeo foi processado
dt_updated_at_brt	datetime	Data de extração dos dados da API para o Lake
nb_media_velocidade	int	Velocidade média dos veículos
ds_faixa	string	Número da faixa da rodovia
ds_categoria	string	Categoria do veículo identificado
nb_quantidade_veiculos	int	Total de veículos detectados na gravação
ds_local_camera	string	Localização da instalação da câmera

ds_latitude	string	Latitude do local onde a câmera está instalada
ds_longitude	string	Longitude do local onde a câmera está instalada
dt_inicial_video	datetime	Data e hora da gravação do vídeo
ds_periodo_dia	string	Período de início da gravação
ts_diferenca_video	time	Tempo entre o início e final da gravação da câmera
dt_inicio_processo	datetime	Data e hora início da execução do processo
dt_fim_processo	datetime	Data e hora fim da execução do processo
st_status_camera	bit	Situação de disponibilidade da câmera

13.6 Fluxo de carga diária

O fluxo de tarefas inicia com a ingestão de dados brutos na tabela Bronze. Após o sucesso dessa etapa, os dados passam por um processo de tratamento e enriquecimento, sendo armazenados na tabela Silver. Por fim, os dados tratados são consolidados e organizados para consumo analítico na tabela Gold.

Descrição das Tarefas do Pipeline:

- Bronze: “source_to_bronze_contagem_veiculos” – Responsável pela ingestão dos dados brutos originados das câmeras.
- Silver: “bronze_to_silver_contagem_veiculos” – Realiza o tratamento, padronização e enriquecimento dos dados.
- Gold: “silver_to_gold_contagem_veiculos” – Consolida os dados finais prontos para análise e visualização.
- Data Quality: “pc_data_quality” – Executa verificações e validações para garantir a qualidade dos dados ao longo do pipeline.
- Expurgo: “expurgo_tabelas_deltas” – Responsável pela limpeza e manutenção das tabelas Delta, removendo dados obsoletos ou desnecessários.

Configuração de Atualização Diária

A atualização diária foi configurada para carregar os registros novos disponibilizados através dos arquivos JSON no diretório blob “umbrellabr”, com base no intervalo do dia atual.

Parâmetros Utilizados na Lógica do Script:

- modulo_api:

Parâmetro fixo recebido pelo notebook através da tarefa do Workflow, representando o nome do módulo referente à entrega em execução.

- tipo_execução:

Utilizado para o reprocessamento definindo o tipo de execução.

- default: Realiza a carga padrão, sendo dia atual.
- reprocessar: Realiza a ingestão por período especificado nas variáveis “data_inicio” e “data_final”.
- modo:

Parâmetro fixo recebido pelo notebook através da tarefa do Workflow, que define o tipo de aquisição.

- full: Realiza uma ingestão completa de todos os dados retornados pela API.
- append: Realiza apenas a ingestão incremental de dados novos ou atualizados.

• data_inicio: Parâmetro fixo recebido pelo notebook através da tarefa do Workflow, que define o intervalo de data início para processamento.

• data_fim: Parâmetro fixo recebido pelo notebook através da tarefa do Workflow, que define o intervalo de data final para processamento.

Os notebooks utilizados no processo são:

- Bronze:

/Workspace/prod/operacao/bronze/contagem_veiculos/source_to_bronze_contagem_veiculos

- Silver: /Workspace/prod/operacao/silver/contagem_veiculos/tabcontagemveiculos_silver

- Gold: /Workspace/prod/operacao/gold/contagem_veiculos/obt_contagem_veiculo

13.7 Ingestão da planilha de localizações das câmeras

O processo de carga dos dados da planilha é realizado através do script desenvolvido na camada Silver, carga diária, realizando a atualização das tabelas com as informações abaixo:

- Diretório padrão para atualização da planilha com os dados das câmeras:
Datalakearteris/bronze/prod/contagem_veiculo/file_local_cameras/dados_localizacao_cameras.csv

•O notebook utilizado para realizar os tratamentos e carga é:

/Workspace/prod/operacao/silver/contagem_veiculos/tabcontagemveiculos_silver

13.8 Job Cluster

O Job Cluster configurado para o processo de aquisição possui as seguintes características:

- Nome do Cluster: **job_contagem_veiculos_cluster**.
- Política: Job Compute.
- Modo de Acesso: Utilizador único.

Configuração de Desempenho:

- Versão do Databricks Runtime: 15.4 LTS (Scala 2.12, Spark 3.5.0).
- Tipo de Worker: Standard_F4s (8 GB de memória, 4 núcleos).
- Tipo de Driver: Igual ao Worker (8 GB de memória, 4 núcleos).
- Número de Trabalhadores: Mínimo de 1 e máximo de 4, com dimensionamento automático ativado.
- Instâncias Spot: Habilitadas para reduzir custos.

Resumo da Configuração:

O cluster opera com um driver de 8 GB de memória e 4 núcleos, e entre 1 e 4 trabalhadores, totalizando 8–32 GB de memória e 4–16 núcleos. O **Unity Catalog** está habilitado, garantindo governança e segurança dos dados. O custo estimado é de 1–3 DBU/h.

Essa configuração foi otimizada para equilibrar o custo e o desempenho durante a ingestão de dados.

job_contagem_veiculos_cluster Formulário simples: DESLIGADO

Nome do cluster

Política

Modo de acesso

Desempenho

Versão do Databricks Runtime

Utilizar a aceleração do Photon

Tipo de worker
 8 GB de memória, 4 núcleos
 Min. workers: Máximo de workers: Instâncias Spot

Tipo de driver
 8 GB de memória, 4 núcleos

Ativar dimensionamento automático

Etiquetas

Adicionar etiquetas

> Etiquetas adicionadas automaticamente

UI | JSON

Resumo

1-3 trabalhadores 8-24 GB de memória
 4-12 núcleos

1 controlador 8 GB de memória, 4 núcleos

Runtime 15.4.x-scala2.12

Unity Catalog Standard_F4
 1-2 DBU/h

Imagem 81 – Job Contagem de Veículos

13.9 Fluxo de reprocessamento

O fluxo de reprocessamento pode ser executado de três maneiras, conforme a necessidade:

Por Período

A execução é configurada no workflow “job_contagem_veiculos”, ajustando os parâmetros “data_inicio”, “data_final” e “tipo_execucao” conforme a necessidade. Os parâmetros devem ser configurados no seguinte formato:

- Data_Inicio: Data inicial de referência para a execução, no formato ‘YYYY-MM-DD’.
- Data_Final: Data final de referência para a execução, no formato ‘YYYY-MM-DD’.
- Tipo_execucao: Deve ser alterado para “2”.

Reprocessamento Completo da Base

Para reprocessar toda a base, utilize o parâmetro “modo” configurado com o valor “overwrite”, conforme ilustrado na imagem abaixo.

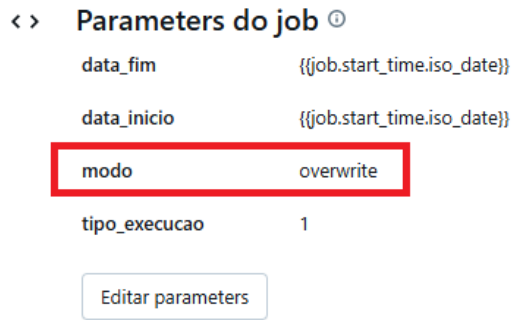


Imagem 82 – Parametros do Job de Contagem de Veículos

14. DEPLOY EM PRODUÇÃO E VALIDAÇÃO DO SERVIÇO DE CONTAGEM DE VEÍCULOS

Foi realizado o deploy do serviço de contagem de veículos no ambiente de produção, conforme o planejamento estabelecido. A aplicação foi disponibilizada por meio de pipeline automatizado, garantindo maior controle de versionamento, rastreabilidade e padronização do processo de entrega.

Após a publicação, foram executadas rotinas de validação para assegurar o correto funcionamento do serviço em ambiente real. Essa validação incluiu:

- Verificação da integridade dos dados recebidos pelas câmeras/dispositivos sensores;
- Monitoramento da performance da API de contagem e tempo de resposta;
- Testes funcionais em tempo real com simulação de tráfego em diferentes horários;
- Comparação amostral entre a contagem automática e o controle manual para aferição da acurácia do modelo.

Durante o período inicial de monitoramento, não foram identificadas falhas críticas. O serviço demonstrou estabilidade, e os resultados de contagem apresentaram margem de erro dentro dos parâmetros aceitáveis definidos na fase de testes.

A equipe de operação foi notificada da liberação da funcionalidade, e os dashboards de monitoramento já estão alimentando os sistemas analíticos com os dados obtidos.

15. PROCESSAMENTO DE DADOS E CONSTRUÇÕES DE KPIS

15.1 Power BI

Abaixo, as informações dos painéis solicitados para desenvolvimento:

15.1.1 Dashboard Contagem de Veículos

- Volume Total, por Carros, Comerciais e Motocicletas.
- Volume de Veículos por Faixa e Categoria.
- Distribuição de Veículos por Faixa e Período do Dia.
- Distribuição de Veículos por Câmeras e Categoria.
- Monitoramento de Câmeras.
- Mapa de Calor – Volume Total por Região.
- Volume de Veículos por Data e Categoria.

15.1.2 Dashboard Eventos de Trânsito

- Total de Eventos, Pedestres, Veículos, Animais, Objetos e Incêndios.
- Eventos por Câmera e Tipo
- Eventos por Período e Tipo
- Eventos por Semana e Hora
- Eventos por Data e Tipo
- Eventos por Mês e Tipo

15.2 Conjunto de dados

O conjunto de dados utilizado no projeto é composto pela tabela gold “prod.operacao.obt_contagem_veiculo” com uma finalidade específica para análise e geração de painéis de indicadores. Segue um resumo:

1. Medidas_contagem:

- Armazena métricas calculadas, como total de ações realizadas, ações atrasadas, conformidades e não conformidades, e indicadores relacionados a inspeções e ocorrências, entre outros.

2. Calendário:

- Tabela de referência temporal com detalhamento de datas, incluindo ano, mês, dias úteis e classificações temporais que suportam análises temporais no projeto.

3. **Obt_contagem_veiculo:**

- Contém informações detalhadas sobre as passagens de veículos.

Azure Databricks

Server Hostname ⓘ

HTTP Path ⓘ

Advanced Options (optional)

Default catalog (optional) ⓘ

Database (optional) ⓘ

Automatic Proxy Discovery (optional) ⓘ

Native query (Requires: Default catalog) (optional) ⓘ

```
select *
from `prod`.`operacao`.`obt_contagem_veiculo`
```

Imagem 83 – Conjunto de dados para Dashboard

Modelo:

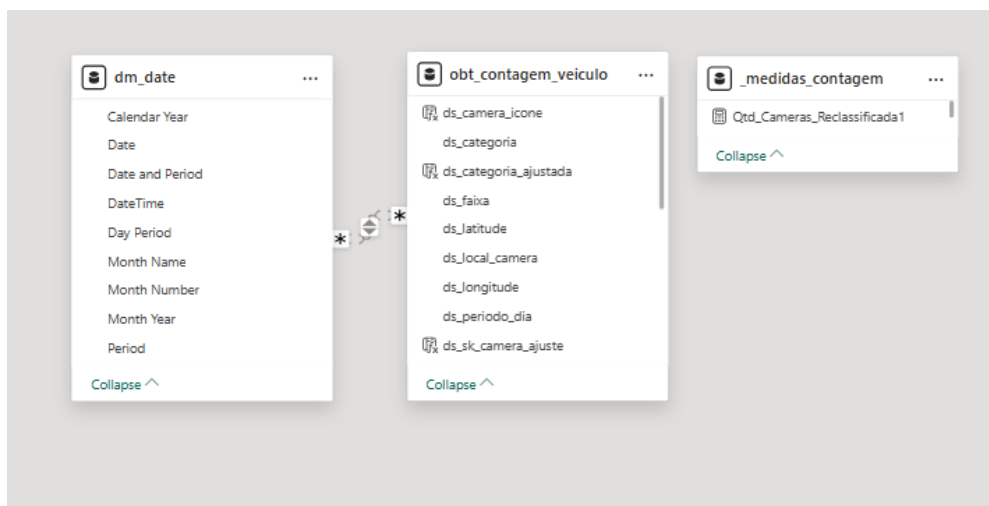


Imagem 84 – Tabelas e Medidas para Contagem de Veículos

15.3 Medidas e visuais do Dashboard

Para a construção dos visuais, utilizamos a tabela obt_contagem_veiculo como fonte principal, explorando as colunas como base para a geração dos valores apresentados nos gráficos.

Medidas utilizadas:

- **UltimaAtualizacaoPorCamera:** A medida UltimaAtualizacaoPorCamera retorna a data e hora da última atualização registrada para cada câmera, com base nos dados mais recentes da tabela obt_contagem_veiculo.
- **UltimoStatusPorCamera:** Retorna o último status registrado para cada câmera, com base na atualização mais recente da tabela obt_contagem_veiculo.
- **Volume_total_cont:** Soma o total de veículos contabilizados na coluna nb_quantidade_veiculos da tabela obt_contagem_veiculo.
- **Qtd_Cameras_Reclassificada:** Retorna 1 para câmeras com status ativo (1), 0 para inativo (0) e em branco para outros casos, com base na coluna st_status_camera.

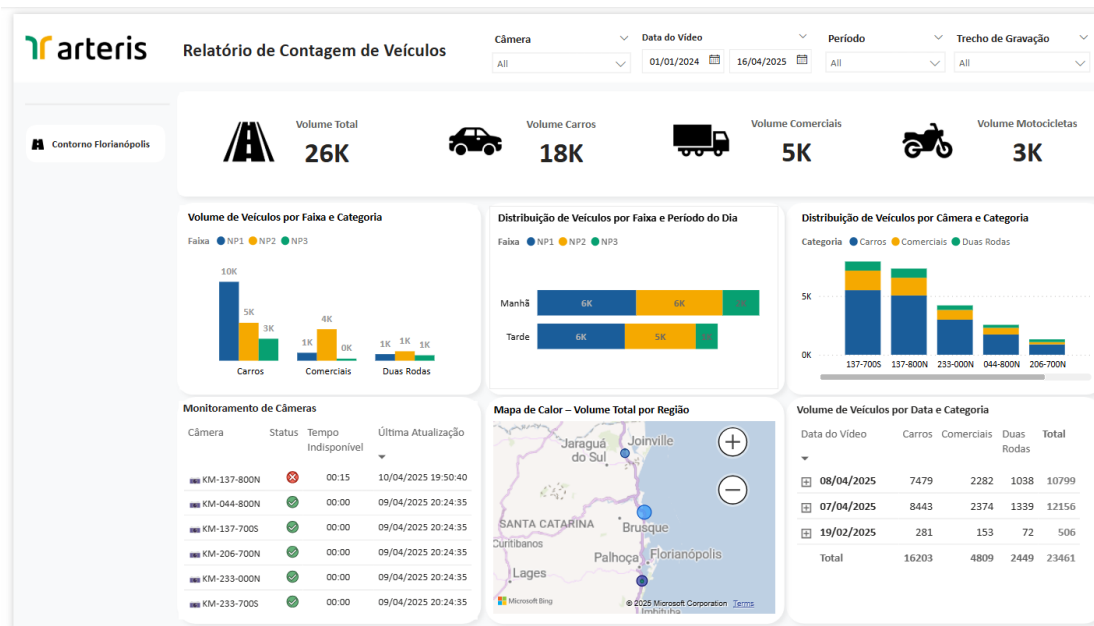


Imagem 85 – Dashboard para Contagem de Veículos

16. REAPONTAMENTO DE SERVIDORES ADQUIRIDOS

Na etapa inicial do projeto, as soluções foram desenvolvidas e validadas em um servidor temporário, utilizado como ambiente de testes e experimentação. Esse ambiente permitiu avaliar diferentes configurações, ajustar parâmetros de processamento e validar o funcionamento dos modelos de inferência em cenários controlados, sem impacto em produção. Durante esse período,

foram instaladas e configuradas todas as bibliotecas, dependências e serviços necessários para suportar os pipelines de contagem e detecção de eventos.

Com a disponibilização do servidor definitivo, projetado para sustentar o ambiente de produção, iniciou-se um processo estruturado de migração e reapontamento. Essa etapa envolveu o levantamento completo das configurações existentes no servidor temporário, a documentação de todas as dependências e parâmetros críticos, e a replicação fiel desses elementos no novo ambiente. Foram também executados ajustes para otimizar a utilização de recursos de CPU, GPU e memória, assegurando que o novo hardware fosse explorado em sua capacidade plena.

Além da simples replicação, a migração foi aproveitada como oportunidade para refinar e padronizar configurações, corrigir eventuais inconsistências identificadas durante os testes e implementar melhorias de segurança e monitoramento. Isso incluiu ajustes nas regras de firewall, configuração de permissões de acesso, e integração com sistemas de logging e alertas.

Após a migração, foram conduzidas baterias de testes para validar o desempenho e a estabilidade do novo ambiente, incluindo simulações de carga e monitoramento de latência. Com os resultados satisfatórios, o servidor definitivo passou a hospedar integralmente o ambiente de produção, oferecendo maior capacidade de processamento, confiabilidade e escalabilidade. Essa infraestrutura agora sustenta não apenas as demandas atuais, mas também está preparada para absorver aumentos de volume de dados, maior complexidade dos modelos e expansão futura das funcionalidades, sem necessidade de grandes reestruturações.

17. IMPLEMENTAÇÃO DE MLOPS

Foi iniciada a estruturação dos processos de MLOps no projeto, com foco em garantir rastreabilidade, organização e consistência no ciclo de vida dos modelos. Foram estabelecidos procedimentos de versionamento de datasets e modelos, permitindo manter histórico das alterações e assegurar a reprodutibilidade dos experimentos. As etapas de treinamento, validação e implantação foram realizadas de forma controlada, ainda que predominantemente manual, com acompanhamento próximo pelas equipes técnicas. Também foram adotadas práticas de registro detalhado de parâmetros, métricas e resultados obtidos, visando facilitar auditorias e futuras iterações. Embora ainda não haja pipelines automatizados de integração e entrega contínua, o ambiente foi preparado para suportar evolução gradual para níveis mais avançados de automação, monitoramento e governança. Essa estrutura inicial já considera possíveis aumentos de demanda, permitindo que, no futuro, novos modelos, maiores volumes de dados e processos de atualização mais frequentes sejam incorporados sem necessidade de reestruturação completa, garantindo escalabilidade técnica e operacional.

18. DEPLOY EM PRODUÇÃO E VALIDAÇÃO DO SERVIÇO DE DETECÇÃO DE EVENTOS

Foi realizado o deploy do serviço de detecção de eventos no ambiente de produção, consolidando uma arquitetura capaz de operar cinco modelos distintos de inferência — detecção de eventos com veículos, pessoas, animais, objetos e incêndios — de forma simultânea e em tempo real. Para alcançar o desempenho necessário, foi empregada a plataforma NVIDIA DeepStream, explorando aceleração por GPU para processar fluxos de vídeo de múltiplas câmeras com baixa latência e alta precisão.

O serviço foi configurado para executar a captura, processamento e análise dos frames de vídeo em pipelines otimizadas, identificando automaticamente a ocorrência de eventos relevantes. Quando um evento é detectado, o sistema gera cards de alerta contendo imagem e descrição, enviados automaticamente para canais no Microsoft Teams, garantindo comunicação imediata com as equipes operacionais responsáveis. Paralelamente, um arquivo JSON estruturado com todas as informações do ocorrido — incluindo timestamp, tipo de evento, localização e metadados adicionais — é salvo e disponibilizado para ingestão em uma pipeline analítica que alimenta dashboards no Power BI.

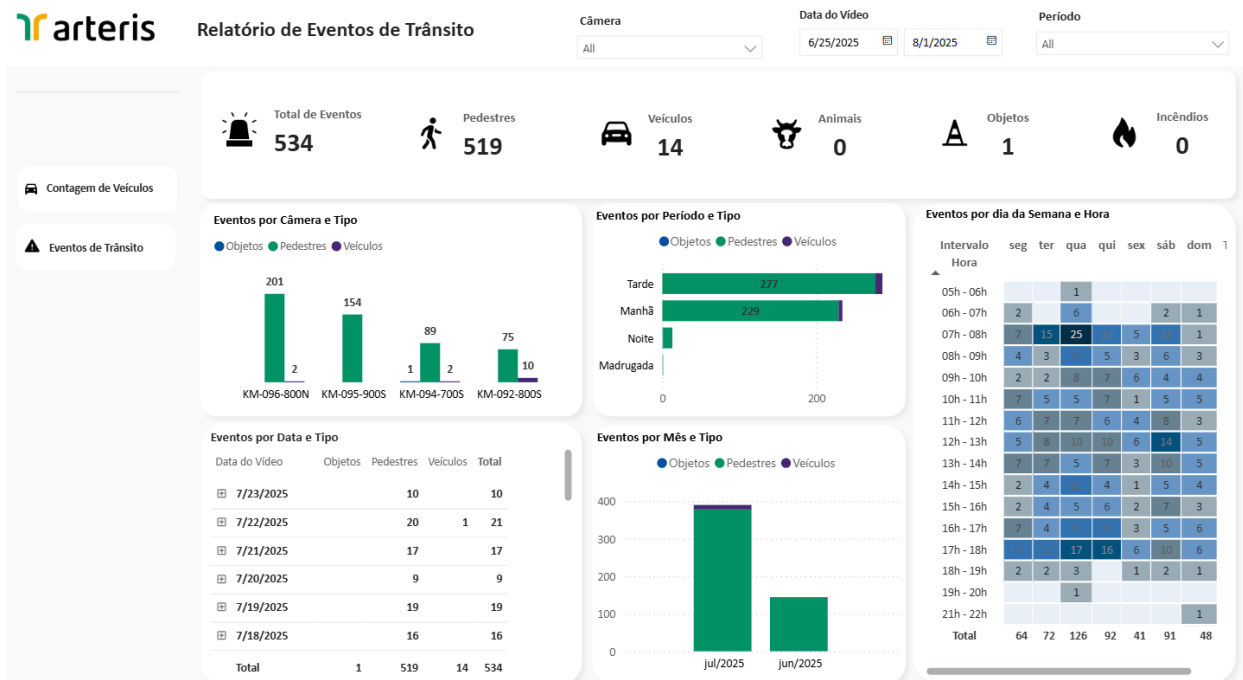


Imagem 86 – Dashboard para Eventos de Trânsito

Durante o processo de implantação, foram replicadas e adaptadas as configurações validadas em ambiente de testes, seguidas de ajustes finos para adequar o desempenho ao tráfego real de dados. Foram realizados testes de carga com múltiplos streams simultâneos, verificação de integridade dos dados gerados e análise de latência ponta a ponta, desde a captura até a entrega dos alertas. Também foram configurados mecanismos de monitoramento para acompanhar métricas de desempenho dos modelos, uso de recursos e taxa de falsos positivos.

A implantação resultou em um sistema estável, com detecção confiável e comunicação ágil, pronto para suportar aumento no número de câmeras, inclusão de novos modelos de detecção e integração com outras plataformas. Dessa forma, a solução não apenas atende às necessidades atuais, mas está preparada para evoluir e escalar conforme as demandas operacionais cresçam.

19. REFINAMENTO FINAL DA SOLUÇÃO

O refinamento final da solução representou uma etapa estratégica para consolidar todos os avanços obtidos ao longo do projeto, incorporando os feedbacks recebidos durante a fase de testes em campo e as observações da operação inicial. Esse processo contemplou de maneira integrada os dois principais serviços desenvolvidos: a contagem de veículos e a detecção de eventos, garantindo consistência, confiabilidade e alinhamento às expectativas dos stakeholders.

No serviço de contagem de veículos, as melhorias se concentraram em aumentar tanto a precisão quanto a performance operacional. Foram aplicados ajustes nos algoritmos responsáveis pelo cálculo da velocidade média, de modo a garantir estimativas mais consistentes em cenários de tráfego variados, incluindo períodos de maior densidade e condições adversas de captura pelas câmeras. Além disso, o tempo de processamento por requisição da API foi otimizado, permitindo uma resposta mais ágil mesmo em horários de pico, quando há maior número de consultas simultâneas. Também foram revisados os filtros de detecção, o que aprimorou o desempenho em situações de sobreposição de veículos ou em ângulos desafiadores de filmagem, reduzindo erros de contagem.

Já no serviço de detecção de eventos — responsável pela identificação em tempo real de veículos, pessoas, animais, objetos e focos de incêndio — o ciclo de refinamento buscou diretamente atender aos pontos mais recorrentes dos feedbacks coletados. Entre os ajustes, destacam-se o refinamento dos thresholds e da calibração específica para cada tipo de evento, a otimização das pipelines do NVIDIA DeepStream para redução de latência e maior eficiência de processamento, e a revisão da lógica de notificação via Microsoft Teams, de forma a evitar alertas redundantes e tornar a comunicação mais assertiva. Também foi realizada uma atualização no formato dos JSONs gerados, garantindo maior padronização e facilitando o consumo estruturado dos dados em dashboards do Power BI.

Os testes de contagem de veículos contaram com 8 câmeras dedicadas, abrangendo o trecho de 2km ao longo da rodovia. Já os testes de detecção de eventos contaram com 4 câmeras dedicadas, abrangendo 10km do trecho concessionado. Ambos os testes envolveram as equipes de Manutenção, Gestão Operacional, Operações e CCO da Arteris, além do time de desenvolvedores da Programmers.

Como resultado dessas melhorias integradas, ambos os serviços alcançaram níveis superiores de estabilidade, responsividade e confiabilidade dos dados. Isso estabelece não apenas uma entrega sólida para a fase atual do projeto, mas também cria uma base escalável e preparada para evolução gradual, permitindo expandir o número de câmeras monitoradas, aumentar o volume de eventos processados e incorporar novas métricas sem comprometer a eficiência do sistema. Por outro lado, na fase de testes, foi necessário manter o pleno funcionamento das inspeções de tráfego. Isso se deu pois, primeiramente, o trecho coberto pelas câmeras com IA foi de apenas 10km, e as Inspeções invariavelmente passariam por este trecho para cobrir os demais quilômetros. Outro fator foi a taxa de detecção relativamente inferior, da inteligência artificial, durante períodos de baixa luminosidade.

Com esse refinamento, a solução como um todo — contemplando contagem e detecção de eventos — encontra-se otimizada e preparada para expansão, assegurando escalabilidade técnica e operacional para atender novas demandas e integrar funcionalidades futuras sem necessidade de reestruturações profundas.

20. HOMOLOGAÇÃO, COLETA DE RESULTADOS E ELABORAÇÃO DE RELATÓRIO FINAL

Execução de plano de testes de homologação com stakeholders.

A etapa de homologação foi conduzida em ambiente de produção controlado, ao longo de **dois meses de operação contínua (Julho a Agosto)**, com a participação ativa dos stakeholders — incluindo representantes das áreas de operação, segurança, tecnologia da informação e gestão executiva. O plano foi desenhado de forma progressiva, contemplando primeiro a validação do serviço de **contagem de veículos** e, em seguida, a homologação do serviço de **detecção de eventos**.

- **Serviço de Contagem de Veículos:** validado no **primeiro mês (16/07 – 22/08)** de operação. Nesse período, foram analisadas métricas de fluxo, velocidade média e taxa de ocupação. Para garantir a acurácia, foram realizadas comparações amostrais com contagens manuais feitas por operadores, em diferentes horários e dias da semana. Situações de alto e baixo tráfego foram contempladas, possibilitando testar o modelo em condições variadas de operação.

- **Serviço de Detecção de Eventos:** validado no **segundo mês**, em duas fases complementares: **pré-testes (25/06 – 01/08)**, voltados para ajustes iniciais de configuração e calibração, e **produção definitiva (06/08 – 31/08)**, dedicada à aferição da precisão e do tempo de resposta na identificação de veículos parados, pedestres, animais, objetos e incêndios. Foram analisadas situações reais e induzidas para verificar a estabilidade da pipeline de processamento e a adequação do envio de alertas ao Microsoft Teams, considerando tanto a relevância das notificações quanto o impacto para os times de monitoramento.

A participação dos stakeholders foi fundamental em três frentes principais:

1. **Validação técnica**, assegurando que os serviços cumprissem os requisitos de desempenho.
2. **Validação operacional**, verificando se os dados eram compreensíveis e aplicáveis na rotina das equipes.
3. **Validação executiva**, confirmando que os dashboards no Power BI ofereciam indicadores consistentes para análises estratégicas.

O processo de homologação resultou em maior alinhamento entre áreas técnicas e de negócio, reforçando a confiança na solução antes da adoção em escala. Além disso, permitiu antecipar ajustes finos — como a revisão de thresholds para determinados eventos e a adequação de filtros em cenários de tráfego intenso — garantindo que ambos os serviços chegassem à fase de operação definitiva de forma mais madura e estável.

Coleta e consolidação de métricas de desempenho.

Durante a fase de homologação, foram coletadas métricas contínuas ao longo de **dois meses de operação controlada**, abrangendo tanto o serviço de contagem de veículos (GPU 0) quanto o de detecção de eventos (GPU 1). A coleta foi automatizada e registrada em dashboards específicos para cada serviço no Power BI, permitindo acompanhar a evolução em tempo real e consolidar os resultados em relatórios periódicos.

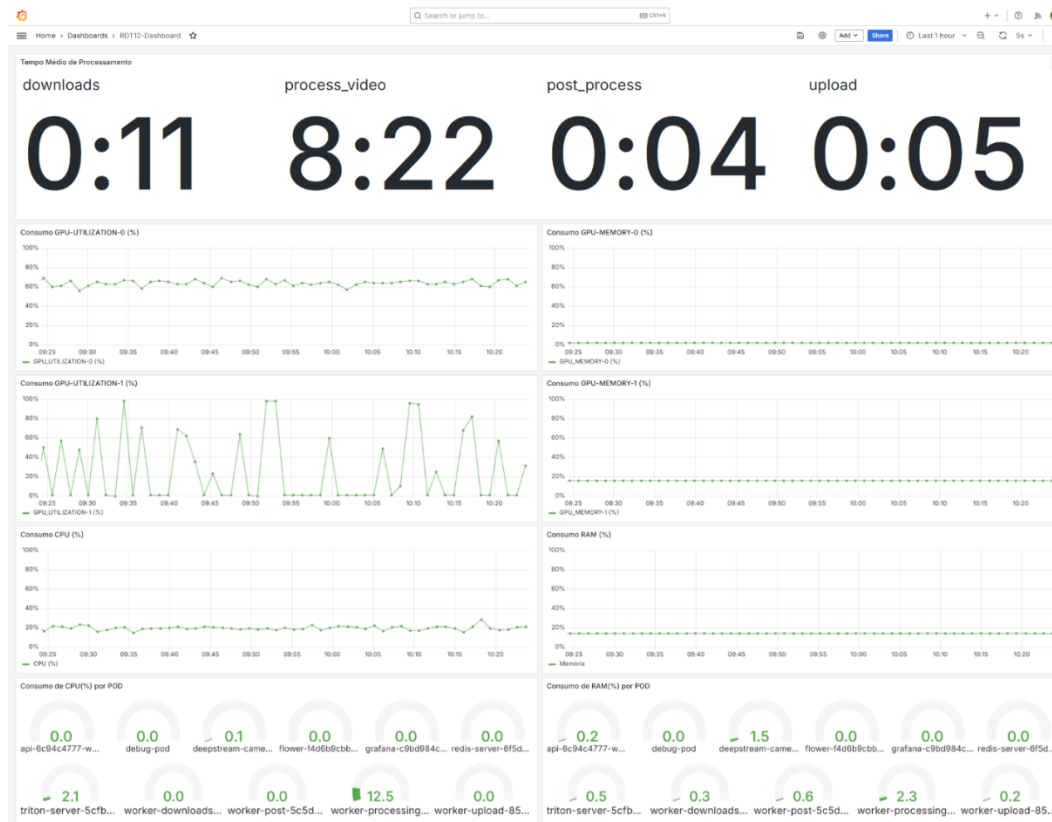


Imagem 87 – Dashboard para Contagem de Veículos

De forma geral, os testes mostraram que o **serviço de contagem** apresentou **consumo estável de recursos**, mesmo sob variação de tráfego, e manteve tempos de resposta consistentes durante horários de pico. Já o **serviço de detecção de eventos** registrou **maior oscilação no uso de GPU e CPU**, reflexo da complexidade de processar múltiplos modelos simultaneamente (pessoas, veículos, animais, incêndio e objetos).

Foram consolidadas as seguintes métricas globais:

- **Contagem de veículos (8 câmeras):**
 - Tempo médio de processamento por trecho: **8 minutos**
 - Consumo médio de GPU: **65%**
 - Memória média da GPU: **2%**
 - Consumo médio de CPU: **20%**
- **Deteção de eventos (4 câmeras):**
 - Taxa média de alertas por dia: **8**
 - Consumo médio de GPU: **65-90%**

- Memória média da GPU: **16%**
- Consumo médio de CPU: **20%**

Esse panorama consolidado forneceu a base para a etapa seguinte de análise mais detalhada, onde foram explorados não apenas os números absolutos, mas também tendências e padrões específicos identificados por câmera, tipo de evento e contexto operacional.

Análise dos resultados obtidos.

Com base nas métricas consolidadas, foi realizada uma análise aprofundada que destacou tanto o desempenho técnico quanto os padrões operacionais observados:

Contagem de Veículos (16/07 – 22/08):

Durante o período analisado, foram contabilizados aproximadamente 13 milhões de veículos ao longo de 8 câmeras, distribuídos em 9 milhões de automóveis de passeio, 3 milhões de veículos comerciais e 2 milhões de motocicletas;

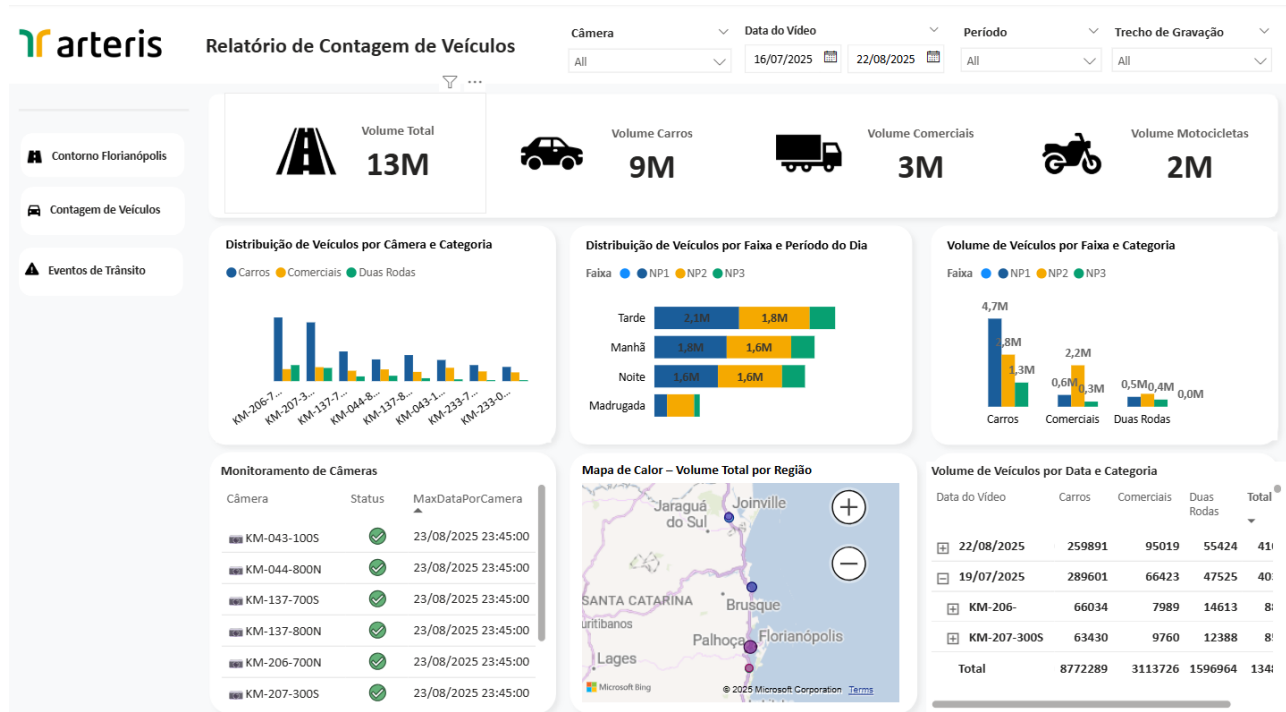


Imagem 88 – Dashboard de resultados Contagem de Veículos

As câmeras localizadas nos trechos **KM-206-700N** e **KM-207-300S** destacam-se como as que mais registraram volume de tráfego, respondendo por uma fatia relevante das contagens.

A tarde concentrou o maior volume de veículos, seguida pela manhã; os períodos noturnos e de madrugada tiveram menor movimentação.

A faixa NP1 foi a mais utilizada em todas as categorias de veículos, enquanto a NP3 apresentou baixa ocupação.

As câmeras localizadas nos trechos de maior fluxo, especialmente próximas a Florianópolis e Palhoça, concentraram parte significativa das contagens.

Detecção de Eventos:

Fase de pré-testes (25/06 – 01/08):

No período observado, foram identificados pelas câmeras inteligentes **534** eventos, dos quais **14** corresponderam a veículos parados, **519** a pedestres e **1** a objetos; as categorias de incêndio e animais não apresentaram registros.

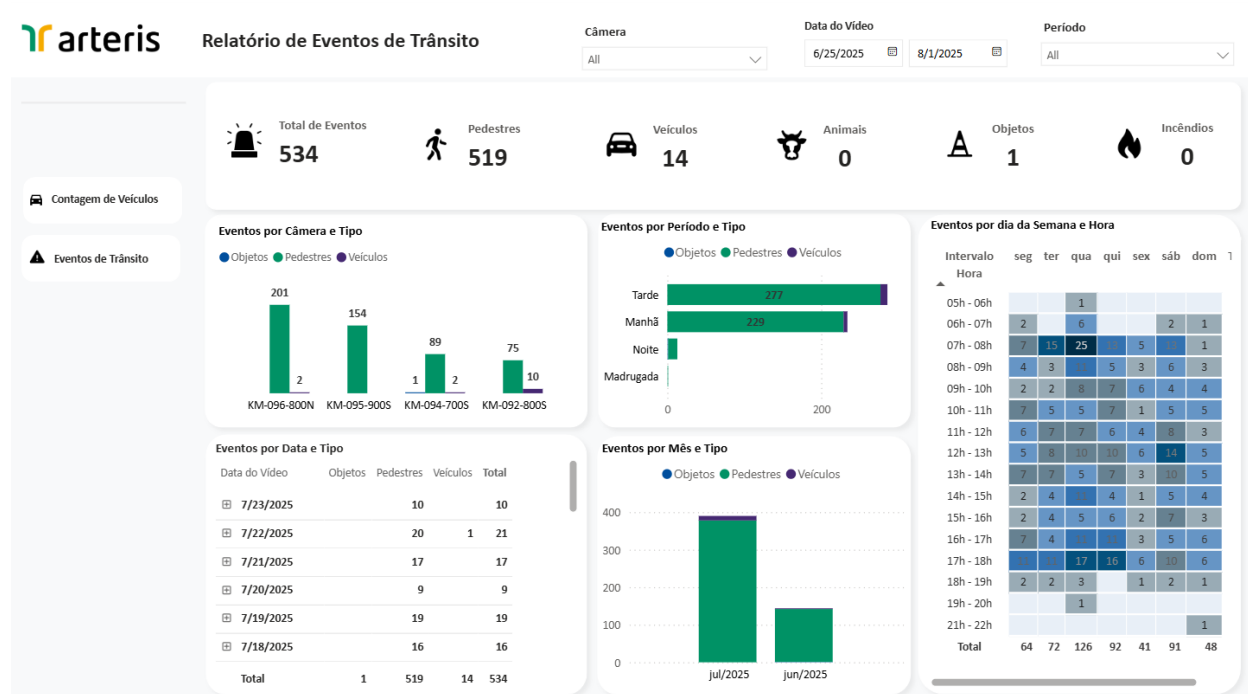


Imagem 89 – Dashboard de resultados pré-testes Detecção de Eventos

Entre os eventos monitorados, a **detecção de pessoas** apresentou maior índice de detecções, evidenciando a necessidade de expandir os canais de notificações para separação de eventos.

A câmera **KM-096-800N**, localizada em área de maior circulação, concentrou a maior parte de todos os eventos detectados, reforçando seu papel estratégico na operação.

Fase de produção (06/08 – 31/08):

No período observado, foram contabilizados aproximadamente **1374** eventos, dos quais **20** corresponderam a veículos parados, **1353** a pedestres e **1** a objetos; as categorias de incêndio e animais não apresentaram registros.

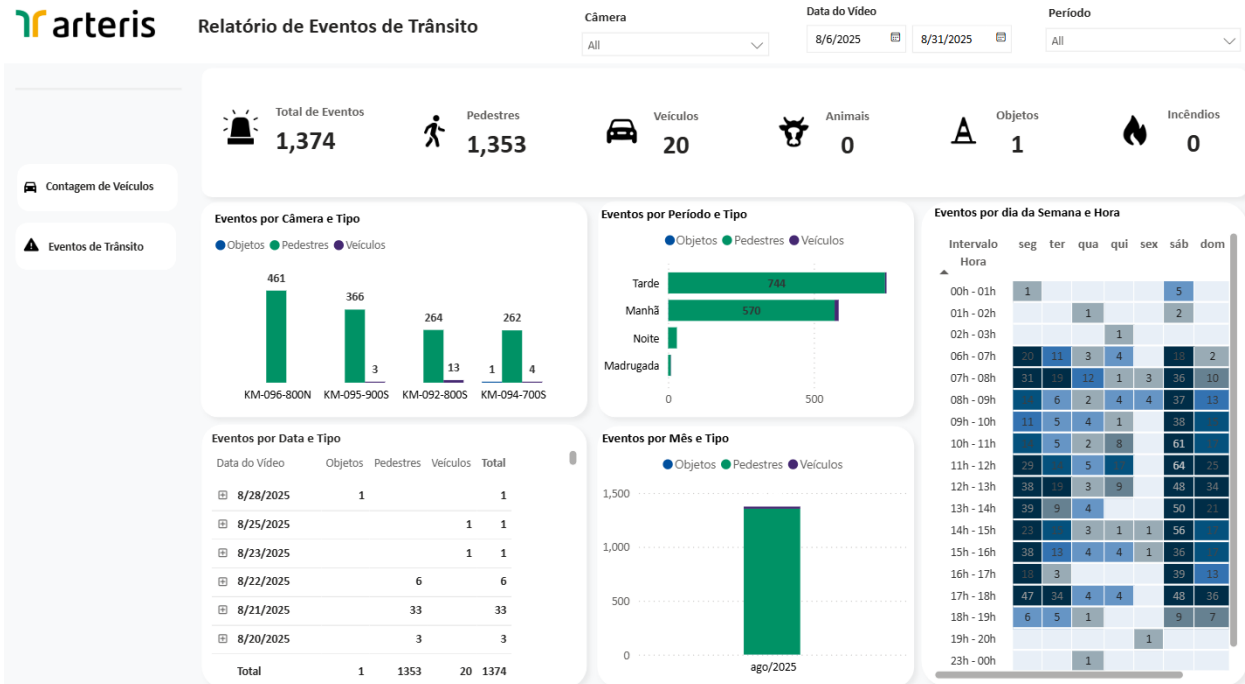


Imagem 90 – Dashboard de resultados produção Detecção de Eventos

Dentre as categorias analisadas, a detecção de pessoas apresentou o maior volume de ocorrências, somando **1353** eventos.

A maioria dos eventos foram detectados durante o período diurno, tendo baixos volumes durante a parte da noite e madrugada de operação.

A câmera **KM-096-800N**, localizada em área de maior circulação, concentrou a maior parte de todos os eventos detectados, reforçando seu papel estratégico na operação.

Panorama geral

O serviço de contagem se mostrou **mais estável** em termos de consumo de recursos e apresentou ótimas taxas de acerto em todos os cenários de tráfego.

O serviço de detecção, apesar de mais exigente em termos computacionais, manteve uma **resposta ágil**, com alertas sendo disparados o mais rápido possíveis, atendendo ao requisito de operação em tempo quase real. Essa análise não apenas validou a eficácia da solução, mas também forneceu **informações práticas** para tomada de decisão, como priorização de câmeras em áreas críticas, ajustes finos nos modelos mais sensíveis e identificação de gargalos potenciais no consumo de infraestrutura.

Conclusão de relatório final técnico e executivo.

A homologação confirmou que a solução desenvolvida cumpre plenamente os objetivos estabelecidos e já apresenta resultados concretos para a operação. Os dois módulos — **contagem de veículos e detecção de eventos** — foram acompanhados ao longo de dois meses em ambiente real, demonstrando desempenho estável, confiabilidade dos dados e relevância prática para os usuários finais.

Os principais ganhos observados incluem:

- **Maior eficiência operacional:** a contagem de veículos passou a fornecer indicadores consistentes sobre fluxo e velocidade média, viabilizando análises de tráfego mais precisas e úteis para planejamento.
- **Aumento da segurança e capacidade de resposta:** a detecção em tempo real de situações críticas (como objetos abandonados, incêndios e movimentação de pessoas em áreas restritas) fortaleceu a prevenção de riscos e permitiu agir de forma mais rápida e direcionada.
- **Integração simplificada com ferramentas de gestão:** os dashboards no Power BI passaram a reunir informações confiáveis e em tempo real, facilitando o acompanhamento pelas equipes operacionais e pela gestão estratégica.
- **Escalabilidade comprovada:** a arquitetura mostrou-se preparada para expansão, seja pelo aumento do número de câmeras monitoradas, seja pelo acréscimo de novos tipos de eventos e métricas.

Do ponto de vista de negócio, a solução representa uma **base sólida para decisões orientadas por dados**, reduzindo incertezas e oferecendo visibilidade sobre áreas críticas da operação. Além disso, abre caminho para ganhos futuros, como:

- Expansão do monitoramento para novos locais e cenários.
- Inclusão de novos modelos de análise, trazendo insights adicionais ao negócio.
- Consolidação de uma plataforma unificada de monitoramento, capaz de centralizar informações de diferentes fontes.

Em conclusão, o projeto atingiu um estágio de **maturidade operacional**, com serviços estáveis, resultados consistentes e forte aderência às necessidades da operação. A solução encontra-se pronta para ser adotada em escala e gerar valor contínuo, tanto no âmbito **operacional**, quanto no **estratégico**.

Revisão de parâmetro contratual – Inspeções de Tráfego

A atual exigência contratual determina a realização de inspeções de tráfego a cada 90 minutos no trecho, com o objetivo de garantir a rápida detecção de incidentes e segurança viária. No entanto, nos últimos anos, recursos tecnológicos avançados estão disponíveis no mercado, como câmeras CFTV e modelos de Inteligência Artificial, que ampliam a capacidade de monitoramento contínuo. Esses sistemas permitem identificar ocorrências em tempo real, reduzindo a dependência exclusiva das rondas físicas.

Diante desse cenário, torna-se pertinente reavaliar a periodicidade das inspeções presenciais. A proposta visa alinhar o parâmetro contratual à nova realidade operacional, mantendo a eficiência e a segurança, mas com maior racionalidade no uso de recursos. Isto posto, esta seção tem como objetivo apresentar os resultados alcançados durante o período de testes.

Durante o período de testes em produção, foi verificado um total de 1.374 detecções feitas pela IA. Deste volume, 1.353 foram detecções de “Pedestres”, 20 de “Carro parado” e 1 de “Objeto na rodovia”, conforme dashboard (Imagem 90).

Apesar do elevado número de detecções, parte delas corresponde a falsos positivos gerados pelos modelos de inteligência artificial — um comportamento esperado, considerando que o conjunto

de dados utilizado para o treinamento ainda está em fase de expansão. Além disso, observou-se que as condições de baixa luminosidade (entre 19h e 5h) podem impactar a precisão das detecções, o que reforça a importância de aprimorar os modelos para esses cenários específicos.

Outro ponto relevante é que nem todas as detecções da categoria “Pedestres” resultaram em ações operacionais, uma vez que os transeuntes não se aproximaram das faixas de rolamento, não apresentaram comportamento de risco e nem permaneceram por tempo significativo que demandaria algum tipo de intervenção por parte da equipe de Operação. Nesse sentido, embora as detecções de Pedestres apresentadas na tabela acima não tenham gerado ocorrências, possibilitaram que a equipe do CCO monitorasse os indivíduos por meio do CFTV, garantindo que alertas fossem encaminhados ao centro de controle toda vez que uma possível situação de risco surgisse. Com base nessas detecções, as equipes avaliaram o caso e atuaram, seja apenas monitorando ou solicitando algum tipo de intervenção/apoio, o que não foi necessário para esses casos durante o período de testes.

Em adição, vale destacar que, em relação à categoria “Objetos na pista”, a maioria das detecções (81%) feitas pela Inspeção no mesmo período referem-se a itens de pequeno porte, como pequenas partes de pneus, parafusos, sacos plásticos e tecidos, todos de tamanho reduzido, conforme exemplos apresentados na Imagem 91, e/ou estavam posicionados em áreas laterais da rodovia, no acostamentos ou bordo, estando, portanto, em locais de difícil identificação pelas câmeras e fora de zonas de risco elevado para o usuário (faixa de rolamento).



Imagem 91 – Objetos identificados pela Inspeção de Tráfego

Os modelos de inteligência artificial têm demonstrado grande potencial na detecção de eventos, especialmente em cenários com boas condições de visibilidade e iluminação. Embora ainda existam desafios relacionados à qualidade das imagens, à distância de visada dos equipamentos e

às condições de luminosidade — como durante a noite ou na madrugada — os testes indicam que, mesmo com essas limitações, a tecnologia já é capaz de oferecer ganhos significativos. Em situações favoráveis, os modelos conseguem identificar eventos com precisão em até 500 metros de distância, e quando a detecção é bem-sucedida, o tempo de resposta pode ser muito mais rápido em comparação às inspeções de tráfego convencionais, representando um avanço importante na eficiência operacional.

No que se refere aos veículos parados durante o período das 6h às 18h, que representam 20 eventos no período, a IA demonstrou desempenho consistente, identificando 9 casos (45% do total), enquanto a inspeção presencial respondeu por 11 casos (55%). Essa performance reforça a capacidade da tecnologia em monitorar situações críticas em tempo real, reduzindo a dependência da inspeção física para este tipo de evento. Por outro lado, as ocorrências relacionadas a objetos na pista, que concentram a maior parte da demanda, continuam sendo detectadas quase exclusivamente pela inspeção, ainda que com menor frequência.

Vale destacar que, no contexto mais amplo do período de janeiro a agosto de 2025, a inspeção foi a origem de somente 37% de todas as ocorrências registradas. Evidenciando que a inspeção não é a forma pela qual a concessionária identifica a maioria das ocorrências na rodovia, demonstrando que não há dependência exclusiva desse meio, bem como a importância e a crescente contribuição de tecnologias e meios complementares para essa atividade.

Origem	Quantidade	%
Inspeção	33.714	37%
Outros	57.146	63%
TOTAL	90.860	100%

Tabela 3 – Ocorrências geradas e origens desde janeiro/2025.

Nesse contexto e com base nas informações acima apresentadas, a proposta de alteração de parâmetro contratual para as inspeções de tráfego em trechos e regiões com fiscalização com câmeras inteligentes, conforme modelo implementado neste RDT, mostra-se pertinente, adequada e viável, não representando risco de comprometimento no atendimento ao usuário nem falhas na comunicação de eventos.

Diante disso, propõe-se a discussão acerca do aumento da janela de intervalo de inspeção de tráfego de 90min para um período superior ou até a exclusão deste parâmetro, especialmente para períodos diurnos (das 6h às 18h) em trechos sem iluminação artificial e 24 horas por dia em trechos

com iluminação artificial em condições de visibilidade satisfatória, considerando a cobertura contínua proporcionada pela IA e CFTV, com acionamento imediato em caso de alerta crítico. Essa mudança permitirá otimização de recursos operacionais e maior eficiência no atendimento a ocorrências críticas, sem aumento do risco ou prejuízo aos usuários da rodovia, desde que mantidos os protocolos de monitoramento remoto e resposta rápida já implementados. A decisão pode ser sustentada por evidências quantitativas que demonstram a robustez do sistema de IA para eventos de maior impacto e a possibilidade de redistribuir esforços da inspeção para atividades mais estratégicas, garantindo equilíbrio entre segurança e eficiência.

A exigência de que veículos de inspeção de tráfego percorram toda a rodovia em intervalos prefixados foi, no passado, uma medida para garantir presença operacional em trechos longos e pouco monitorados. No entanto, com o avanço da tecnologia, expansão da cobertura de rede móvel e telefonia, e a ampla implantação de câmeras nas rodovias pedagiadas, a função da inspeção de tráfego evoluiu. Hoje, ela traz mais benefício ao usuário quando utilizada como recurso dinâmico de atendimento, sinalização emergencial e resposta a eventos, priorizando segurança e eficiência operacional.

Além disso, reduzir deslocamentos sistemáticos sem demanda real contribui para menor emissão de gases poluentes, otimização de recursos e melhor uso da infraestrutura. A modernização dos contratos deve refletir essa nova realidade, alinhando-se às melhores práticas internacionais e às exigências ambientais e tecnológicas do presente.

Portanto, conclui-se que os modelos de inteligência artificial têm apresentado resultados bastante promissores, demonstrando grande potencial para otimizar a detecção de eventos. Embora fatores como qualidade das imagens, distância de visada e condições de luminosidade possam influenciar o desempenho, os testes indicaram que, mesmo nessas situações, a tecnologia mantém boa capacidade de identificação. Em cenários favoráveis, a IA consegue monitorar extensões significativas da rodovia e atuar de forma complementar às inspeções físicas. Além disso, quando a detecção é bem-sucedida, o ganho operacional é expressivo: o tempo de resposta é mais rápido em comparação às inspeções convencionais. Esses resultados reforçam a relevância da IA como ferramenta estratégica para aumentar a eficiência e a segurança viária.

21. CONCLUSÕES, CONSIDERAÇÕES FINAIS E PRODUTOS

21.1 Alcance dos Objetivos Gerais e Específicos

O projeto teve como **objetivo geral** desenvolver e validar uma solução de visão computacional capaz de realizar:

1. Contagem e cálculo de velocidade de veículos em vias monitoradas;
2. Detecção de eventos relevantes à operação rodoviária, como presença de pessoas, animais, objetos na pista e indícios de incêndio.

Os objetivos foram atingidos em sua totalidade, sendo possível comprovar a eficácia dos modelos treinados, a adequação da arquitetura de processamento (Triton Server e DeepStream) e a integração bem-sucedida com ferramentas analíticas (Azure e Power BI).

Como objetivo específico adicional, estabeleceu-se a entrega de relatórios analíticos estruturados em dashboards, o que também foi cumprido, proporcionando maior visibilidade e transparência na operação.

21.2 Avaliação dos Métodos e Técnicas

A combinação das técnicas escolhidas mostrou-se adequada e eficaz:

- O **Scrum** permitiu conduzir o desenvolvimento com ciclos curtos, entregas contínuas e alinhamento constante com os stakeholders.
- Os modelos de visão computacional baseados em **YOLOv11** apresentaram alta precisão na detecção de veículos, principalmente após o treinamento em dataset customizado.
- O uso de **Triton Server** na contagem proporcionou baixo tempo de resposta, adequado para análises em volume.
- O **DeepStream** mostrou-se ideal para o monitoramento de eventos em tempo real, garantindo escalabilidade e suporte a múltiplas classes de detecção.
- A integração com **Azure Blob Storage, Databricks e Power BI** consolidou o ciclo de dados até a visualização estratégica.

Como melhorias potenciais, destaca-se a possibilidade de:

- Expandir o dataset com novos cenários e condições climáticas adversas, aumentando a robustez dos modelos.
- Ampliar a lógica de notificação com modelos preditivos, que antecipem riscos antes da consolidação de um evento.

21.3 Avaliação dos Resultados em Relação ao Plano de Trabalho

Os resultados obtidos foram plenamente compatíveis com os objetivos fixados no Plano de Trabalho aprovado. A metodologia permitiu atingir tanto as metas de acurácia técnica quanto os benefícios operacionais esperados, confirmando a validade do planejamento inicial.

Atividades Previstas	Produtos Esperados	Atividades Executadas	Status das Atividades Executadas	Produtos Gerados
Criação de base de imagens inicial para o treinamento da inteligência artificial	Base de imagens para treinamento	Manipulação do dataset como toda separação e preparação das melhores imagens para serem utilizadas no modelo	Concluído	Base de imagens para treinamento de IA
Treinamento da inteligência artificial.	Primeiro treinamento da IA.	Modelos de Animais, Eventos, Contagem de veículos, Objetos, Incêndios e pessoas treinados.	Concluído.	Treinamento finalizado.
Aprimoramento da aplicação de detecção de eventos.	Melhorias no desempenho e métricas dos modelos.	Novas versões dos modelos de Animais, Eventos, Contagem de veículos, Objetos, Incêndios e pessoas treinados.	Concluído.	Aprimoramento com métricas e desempenho melhores do que inicialmente.
Configuração Edge Computing em servidores temporários.	Edge computing configurado corretamente.	Edge computing instalado no servidor temporário.	Concluído.	Configuração finalizada.
Aprimoramento da aplicação de contagem de veículos.	Melhorias no desempenho e métricas do modelo.	Novas versões do modelo de contagem treinados.	Concluído.	Aprimoramento com métricas e desempenho melhores do que inicialmente.
Configuração Kubernetes Edge em servidores temporários.	Kubernetes edge configurado corretamente.	Kubernetes edge instalado no servidor temporário.	Concluído.	Configuração finalizada.

Aquisição de equipamentos e preparação de infraestrutura na rodovia para processamento do modelo de contagem de veículos.	Todos os equipamentos necessários para execução do projeto adquiridos e pronto para serem utilizados.	Câmeras e equipamentos instalados e em funcionamento para captura de imagens e processamento da IA.	Concluído.	Equipamentos comprados e instalados corretamente nas rodovias.
Aquisição dos servidores para sustentação dos modelos.	Servidor com as especificações necessárias para execução do projeto.	Adquiridos os servidores da Lenovo conforme acordado nos documentos.	Concluído.	Servidor adquirido para receber toda estrutura e sistema de IA.
Configuração e preparo de infraestrutura de TI Arteris.	Infraestrutura configurada e pronta para ser utilizada.	Configuração realizada com sucesso com os requisitos mapeados.	Concluído.	Infraestrutura preparada para executar o projeto.
Aprimoramento da aplicação de contagem de veículos.	Melhorias no desempenho e métricas do modelo.	Novas versões do modelo de contagem treinados.	Concluído.	Aprimoramento com métricas e desempenho melhores do que anteriormente.
Criação arquitetura ingestão de dados	Arquitetura criada para suporta todos os requisitos do projeto.	Data Lake storage criado, workspace azure databricks configurado, modelo de dados criado, fluxo de carga diaria, ingestao dos dados, job cluster, fluxo de reprocessamento criado.	Concluído.	Arquitetura criada e pronta para receber ingestão de dados.
Deploy em produção e validação do serviço de contagem de veículos.	Modelo de contagem de veículos em produção e disponibilização das contagens realizadas.	Modelo e todas as configurações e instalações migradas para o servidor final, Tritton server instalado e executando o modelo de contagem de veículos.	Concluído.	Deploy do modelo de contagem de veículos sendo executado corretamente.

Processamento de dados e construções de KPIs.	Dados sendo processados corretamente e alimentando o Power BI.	Power BI criado e configurado com os KPIs levantados com o cliente, conjunto de dados alimentando as tabelas.	Concluído.	Processamento dos dados realizados e KPIs criados.
Reapontamento de servidores adquiridos.	Garantir que os recursos de hardware e infraestrutura adquiridos estivessem devidamente alocados para suportar os serviços de contagem de veículos e detecção de eventos.	Ajustes de configuração de rede, atualização de drivers NVIDIA, configuração de ambiente virtualizado, balanceamento de carga entre servidores para otimizar o processamento.	Concluído.	Servidores corretamente configurados e integrados ao ambiente de produção, assegurando alta disponibilidade e performance estável para os modelos de IA.
Implementação de MLOps	Estabelecer um primeiro nível de práticas de governança de modelos e dados, garantindo padronização mínima, rastreabilidade de versões e organização dos experimentos.	Definição de convenções de versionamento de datasets, registro e acompanhamento das versões de modelos, documentação dos experimentos realizados e padronização de diretórios e nomenclaturas.	Concluído.	Estrutura inicial de MLOps implantada (nível 0), proporcionando maior organização e controle sobre dados e modelos, assegurando futuras evoluções.
Deploy em produção e validação do serviço de detecção de eventos	Colocar em operação o modelo de detecção de eventos (pessoas, animais, objetos na pista, incêndios) integrado às câmeras e validá-lo em cenários reais. Utilizar os veículos de inspeção de tráfego de forma	Configuração e deploy no NVIDIA DeepStream, integração com as 4 câmeras definidas, testes de detecção em tempo real, integração com API do Digifort e envio de JSONs para o Data Lake.	Concluído.	Serviço de detecção de eventos operacional e validado, entregando informações em tempo real para dashboards e relatórios no Power BI.

	reativa aos eventos detectados e alertados pelo modelo.			
Refinamento final da solução	Ajustar parâmetros finais da solução após período de testes, consolidando a estabilidade e performance dos serviços.	Revisão de parâmetros de inferência nos modelos, ajustes de dashboards no Power BI, melhoria nos fluxos de ingestão e reprocessamento, aplicação de feedback coletado com os stakeholders durante as dailies e reviews.	Concluído.	Solução refinada e otimizada, com menor latência, maior confiabilidade nas detecções e visualizações ajustadas às necessidades de negócio do cliente.
Homologação, coleta de resultados e elaboração de relatório final	Consolidar a avaliação dos dois serviços (contagem de veículos e detecção de eventos) após período de operação, documentar resultados e entregar relatório técnico-executivo.	Execução de plano de testes com stakeholders, coleta de métricas de desempenho dos servidores e modelos, análise comparativa dos resultados entre câmeras e tipos de eventos, elaboração de relatório técnico e executivo com conclusões e recomendações.	Concluído.	Relatório final elaborado e apresentado, validando a solução junto aos stakeholders e indicando próximos passos para evolução do sistema.

21.4 Utilidade Prática e Contribuições

Os resultados e produtos desenvolvidos possuem aplicabilidade direta na infraestrutura rodoviária e no dia a dia da concessionária, oferecendo:

- Melhoria na segurança viária, por meio da detecção ágil de eventos críticos.
- Agilidade na resposta operacional, com envio automático de alertas para as equipes responsáveis.
- Geração de dados para posterior análise e levantamento de KPIs.

- Maior confiabilidade na tomada de decisão, com dashboards claros e métricas objetivas.
- Conforto e segurança ao usuário, reduzindo tempo de exposição a riscos na via.

21.5 Vantagens, Deficiências e Possibilidades de Melhoria

Vantagens identificadas:

- Processamento em tempo real com baixa latência.
- Flexibilidade para inclusão de novas classes de eventos.
- Integração facilitada com sistemas externos.

Deficiências observadas:

- Sensibilidade a condições de baixa luminosidade e chuva intensa.
- Dependência de conectividade estável para ingestão contínua de vídeo.

Possibilidades de melhoria:

- Adotar câmeras com maior sensibilidade noturna para ampliar cobertura.

21.6 Viabilidade Econômica e Custo-Benefício

Os materiais e tecnologias empregados apresentaram **viabilidade econômica satisfatória**. A adoção de soluções baseadas em GPU (NVIDIA Triton e DeepStream) mostrou-se competitiva em termos de **custo-benefício**, pois reduziu a necessidade de servidores adicionais para processar grandes volumes de vídeo.

Comparado ao mercado, o sistema apresentou **maior eficiência operacional**, com custos proporcionais ao ganho em produtividade, segurança e redução de riscos.

21.7 Potencialidades e Limitações

- **Potencialidades:** escalabilidade do sistema para novas rodovias, replicabilidade em diferentes contextos e possibilidade de integrar com plataformas de IoT.
- **Limitações:**
 1. Dependência de câmeras com qualidade mínima de resolução, zoom óptico;
 2. Estabilidade da rede para transmissão contínua dos fluxos;
 3. Dependência de luminosidade para precisão da detecção de eventos.

21.8 Impactos Ambientais

O impacto ambiental associado à solução pode ser classificado como moderado e positivo. Embora as tecnologias adotadas não demandem obras físicas ou alterações significativas na infraestrutura rodoviária, sua utilização implica em benefícios indiretos ao meio ambiente, como:

- Redução da necessidade de tráfego de veículos de fiscalização, já que grande parte do monitoramento de anomalias e eventos poderá vir a ser realizado remotamente via vídeo.
- Consequentemente, ocorre diminuição no consumo de combustíveis fósseis e na emissão de poluentes (CO₂, NO_x, material particulado), contribuindo para metas de sustentabilidade.
- Redução da pegada de carbono associada à operação de campo.
- Diminuição do risco de acidentes envolvendo veículos de inspeção em deslocamento.

Portanto, o uso da solução representa uma evolução não apenas operacional, mas também ambiental, alinhando-se a políticas de transporte sustentável e à busca por operações mais limpas e seguras.

21.9 Quadro Comparativo – Situação Atual x Solução Proposta

Parâmetro	Situação Usual (Sem Sistema)	Situação Alcançada com o Projeto
Contagem de veículos	Utilização do SAT	Automática e contínua
Deteção de eventos	Equipes em campo / reativa	Automática, em tempo real, com envio de alertas
Tempo de resposta a ocorrências	Elevado (horas)	Reduzido (minutos)
Confiabilidade das métricas	Dependente de observação humana	Baseada em modelos de IA calibrados
Integração com sistemas analíticos	Limitada	Dashboards automatizados no Power BI

21.10 Produtos Gerados

Os principais produtos entregues pelo projeto foram:

- Modelos de visão computacional treinados e validados para contagem e eventos.
- Pipelines de processamento baseados em Triton Server e DeepStream.
- APIs de integração com Digifort e sistemas corporativos.
- Estrutura de dados em JSON, armazenada em Azure Blob Storage.
- Dashboards no Power BI para contagem/velocidade e eventos.

- Relatórios técnicos documentando metodologias, resultados e recomendações.
- Proposição de revisão contratual para a alteração de obrigatoriedades referentes ao monitoramento das rodovias com a utilização de novas tecnologias.

21.11 Transferência de Tecnologia e Conhecimento

Para garantir a disseminação dos resultados, estão previstos:

- Documentação técnica (manuais e guias de uso do sistema).
- Apresentações do que foi desenvolvido ao passar das sprints (sprint reviews).
- Notebooks executáveis para treinamento dos modelos de Inteligência Artificial.

22. REFERÊNCIAS BIBLIOGRÁFICAS

Sem referências bibliográficas.

23. ANEXOS

Todos os recursos utilizados e desenvolvidos no projeto se localizam no seguinte diretório do Azure Blob Storage; *rdt12-als/Documentation/Desenvolvimento*.

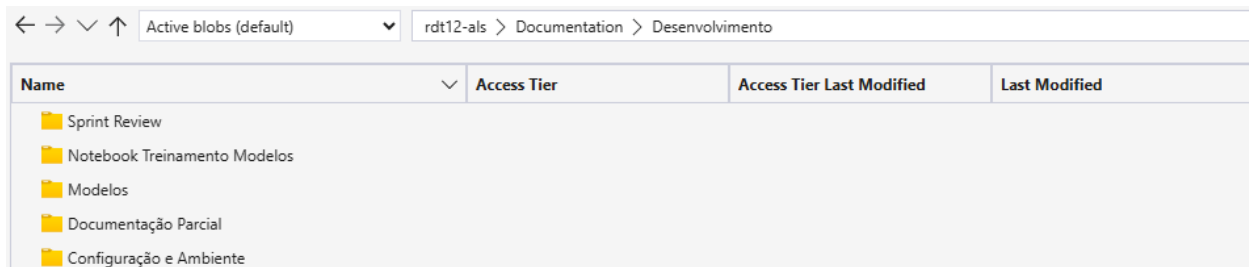


Imagem 92 – Anexos do Projeto no Azure Blob Storage