UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ACHILLES MACARINI NETO
EDUARDO CESAR PATROCINIO ANDREOLI

DETECÇÃO DE ANOMALIAS EM VIAS URBANAS: UMA ABORDAGEM BASEADA EM REDES ADVERSÁRIAS GENERATIVAS WASSERSTEIN E VISÃO COMPUTACIONAL

CURITIBA
2025

ACHILLES MACARINI NETO
EDUARDO CESAR PATROCINIO ANDREOLI

DETECÇÃO DE ANOMALIAS EM VIAS URBANAS: UMA ABORDAGEM BASEADA EM REDES ADVERSÁRIAS GENERATIVAS WASSERSTEIN E VISÃO COMPUTACIONAL

Anomaly Detection in Urban Roads: A Wasserstein Generative Adversarial Network and Computer Vision–Based Approach

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).
Orientador: Roberto Zanetti Freire.
Coorientador: Narco Afonso Ravazzoli Maciejewski.

CURITIBA
2025

ACHILLES MACARINI NETO
EDUARDO CESAR PATROCINIO ANDREOLI

DETECÇÃO DE ANOMALIAS EM VIAS URBANAS: UMA ABORDAGEM BASEADA EM REDES ADVERSÁRIAS GENERATIVAS WASSERSTEIN E VISÃO COMPUTACIONAL

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica do curso de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 25/novembro/2025

Roberto Zanetti Freire
Doutorado
Universidade Tecnológica Federal do Paraná

Elder Oroski
Doutorado
Universidade Tecnológica Federal do Paraná

Lucas Pioli Rehbein Kurten
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA
2025

RESUMO

O aumento da quantidade de veículos e a intensificação do tráfego urbano elevaram a ocorrência de acidentes em vias públicas, tornando essencial o uso de tecnologias para monitoramento eficiente do trânsito. Este trabalho propõe um sistema inteligente baseado em redes adversárias generativas Wasserstein, aprendizado de máquina e visão computacional para o registro e avaliação de anomalias no trânsito. O sistema emprega técnicas de processamento de imagens e redes neurais artificiais para identificar eventos anômalos. A metodologia utiliza a base de dados da *track* 4 do *AI City Challenge* de 2021, aplicando o YOLOv8n para detecção de veículos e padronizando trajetórias em sequências de 20 pontos, das quais são extraídas cinco características cinemáticas: posição horizontal, posição vertical, velocidade, aceleração e direção. Três modelos de detecção foram desenvolvidos com a arquitetura Wasserstein *Generative Adversarial Network with Gradient Penalty* (WGAN-GP): uma implementação com camadas lineares e duas com camadas *Long Short-Term Memory* (LSTM) para representação temporal, empregando diferentes bibliotecas Python. A avaliação quantitativa demonstrou que a implementação LSTM em PyTorch apresentou os melhores resultados, com *F1-Score* de 0,2897 e AUC-ROC de 0,6329. Na comparação com os resultados do desafio, o modelo alcançou pontuação S4 de 0,1995, correspondendo à oitava posição na classificação oficial da track 4. O desempenho mostrou-se consistente em cenários com trajetórias estáveis, evidenciando limitações em vídeos com falhas de detecção ou rastreamento. O estudo discute essas restrições e sugere direções para pesquisas futuras.

Palavras-chave: visão computacional; anomalias no trânsito; redes adversárias generativas; rastreamento de veículos; detecção de anomalias.

ABSTRACT

The increase in the number of vehicles and the intensification of urban traffic have raised the occurrence of accidents on public roads, making the use of technologies for efficient traffic monitoring essential. This work proposes an intelligent system based on Wasserstein generative adversarial networks, machine learning, and computer vision for the recording and evaluation of traffic anomalies. The system employs image processing techniques and artificial neural networks to identify anomalous events. The methodology uses the dataset from track 4 of the 2021 AI City Challenge, applying YOLOv8n for vehicle detection and standardizing trajectories into sequences of 20 points, from which five kinematic features are extracted: horizontal position, vertical position, speed, acceleration, and direction. Three detection models were developed using the Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) architecture: one implementation with linear layers and two with Long Short-Term Memory (LSTM) layers for temporal representation, employing different frameworks. Quantitative evaluation showed that the LSTM implementation in PyTorch achieved the best results, with an F1-Score of 0.2897 and AUC-ROC of 0.6329. Compared to the challenge results, the model achieved an S4 score of 0.1995, corresponding to the eighth position in the official ranking of track 4. The performance was consistent in scenarios with stable trajectories, highlighting limitations in videos with detection or tracking failures. The study discusses these constraints and suggests directions for future research.

Keywords: computer vision; traffic anomalies; generative adversarial networks; vehicle tracking; anomaly detection.

LISTA DE TABELAS

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

Segundo Silva (2013), no século XX, o automóvel era visto como um símbolo de progresso e liberdade, em especial para a Europa do pós-guerra, tendo iniciado uma grande definição de políticas públicas e estratégias de acessibilidade para manter o domínio do automóvel e da mobilidade urbana. A ideologia do progresso devido ao trânsito continuou crescendo e em ascensão, e atualmente com uma nova leva de países emergentes em forte crescimento econômico, pode-se notar o aumento da demanda por veículos particulares.

Ainda, para o Jornalista, a mobilidade urbana no século XXI se torna um direito social, um desafio ambiental e ainda como um símbolo de progresso, mas mantendo o desafio de garantir mobilidade segura e acessível para todos. Para Aguinaldo Ribeiro (Ministério do Desenvolvimento Regional, 2014), ex ministro das cidades, o deslocamento é considerado atualmente como um grande investimento público, tanto para o bem estar da população como o desenvolvimento econômico da região, podendo-se afirmar que:

> “Quando se trata de mobilidade, considera-se a qualidade de vida das pessoas.” (Adaptado de: Ministério do Desenvolvimento Regional, 2014).

Tendo conhecimento da importância do bem-estar no trânsito para a sociedade, identifica-se um problema de grande relevância social. Segundo dados da Organização Mundial da Saúde (OMS, 2022), 1,19 milhões de pessoas perdem suas vidas devido a acidentes em estradas, enquanto 20 a 50 milhões sofrem acidentes não fatais, onde muitos resultam em deficiências. As ocorrências em vias urbanas são a principal causa de morte de crianças e jovens entre 5 e 29 anos.

Devido à grande importância do trânsito e sua escala de ocorrências, a OMS, segundo a Assessoria Especial de Comunicação do Ministério da Infraestrutura do Brasil (2021), idealizou o Plano Global para a Década de Ação para Segurança Viária, entre 2021-2030, com o propósito de reduzir em 50% o total de mortes no trânsito no mundo nos próximos 10 anos. O Brasil teria criado em 2018 e publicado na Resolução do Conselho Nacional de Trânsito (Contran), de 13 de setembro de 2021, o Plano

Nacional de Redução de Mortes e Lesões no Trânsito, com a intenção de disseminar por todo o território nacional ações para permitir alcançar as metas estabelecidas de redução de mortes e feridos, podendo salvar cerca de 86 mil pessoas nas vias urbanas até 2030.

No entanto, segundo estudo do Instituto de Pesquisa Econômica Aplicada (2023) as mortes no trânsito tiveram um aumento de 13,5%. De acordo com os pesquisadores do Ipea, os acidentes com motocicletas foram os principais responsáveis pelo aumento das mortes, que dobraram entre 2010 e 2019. Eles também destacam que, com o reaquecimento da economia, as taxas de mortalidade tendem a crescer significativamente no curto prazo, tornando essencial a implementação de mudanças voltadas à educação e à infraestrutura.

Com o objetivo de mitigar esse problema e promover a segurança viária, além de redirecionar recursos econômicos para áreas mais produtivas, foi idealizado o desenvolvimento de uma tecnologia capaz de detectar anomalias no tráfego urbano, por meio de técnicas de visão computacional e aprendizado de máquina. Ao detectar esses eventos, é possível encontrar padrões que podem ser utilizados dentro da segurança dos usuários das vias e, possivelmente, reduzindo o risco de fatalidades.

1.1 Problema

O significativo aumento no número de acidentes de trânsito, com consequências graves para a saúde pública, causado pela crescente demanda por veículos particulares e pela situação econômica das nações, tornou-se um desafio global. Segundo dados da Organização Mundial da Saúde (World Health Organization, 2023), acidentes de trânsito resultam na morte de 1,19 milhões de pessoas anualmente, além de provocar lesões em 20 a 50 milhões de indivíduos, podendo ocasionar deficiências permanentes.

Esse cenário destaca a necessidade de promover a mobilidade urbana segura e acessível. Apesar de iniciativas dos ministérios brasileiros, como o Plano Global para a Década de Ação para Segurança Viária e o Plano Nacional de Redução de Mortes e Lesões no Trânsito no Brasil, demonstrarem esforços para mitigar o problema (Ministério dos Transportes do Brasil, 2021), a realidade mostra que as mortes no trânsito aumentaram na última década (Instituto de Pesquisa Econômica Aplicada, 2023). Este aumento evidencia a urgência de implementar tecnologias,

como aquelas baseadas em visão computacional e aprendizado de máquina, para detectar anomalias no tráfego, contribuindo assim para a segurança viária e a qualidade de vida das pessoas.

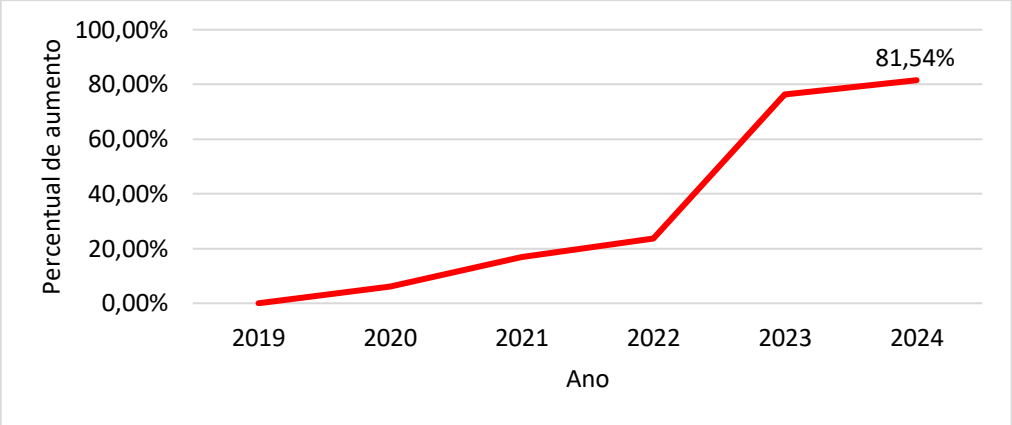
O número de pesquisas na área de monitoramento do tráfego cresce continuamente, esse crescimento é causado pela complexidade dos problemas tratados e seus desafios. Isso demonstra que o desenvolvimento de soluções cada vez mais robustas é importante para os cenários urbanos. (Azfar et al., 2024)

1.2 Justificativa e motivação

Considerando os riscos significativos relacionados aos acidentes de trânsito, conforme a Organização Mundial da Saúde (World Health Organization, 2023), a monitorização do tráfego se destaca como uma das maiores prioridades dos países na preservação e proteção da vida humana. No Brasil, em 2023, foi destinado um montante de R$23 bilhões para investimentos na manutenção e conservação de estradas e ferrovias. Segundo o atual secretário nacional de trânsito, Adrualdo Catão, tais investimentos são cruciais não apenas para salvar vidas, mas também para impulsionar o país de volta ao caminho do crescimento (Ministério dos Transportes do Brasil, 2023).

Segundo uma análise do número de artigos no Google Scholar, com a pesquisa realizada utilizando a consulta "*Traffic Management Systems*" e verificando a quantidade de resultados no Google Scholar em cada ano, de 2019 a 2024, pode-se notar um aumento de aproximadamente 82% em relação ao primeiro ano da pesquisa, como demonstrado na Figura 1.

Figura 1 - Pesquisas relacionadas a sistemas de gerenciamento de trânsito

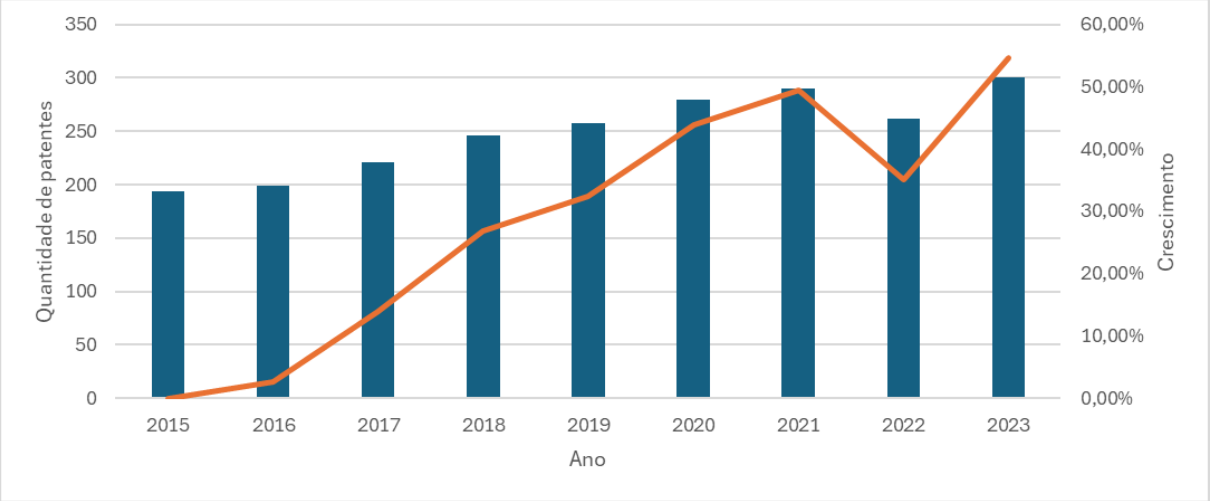


Fonte: Os autores (2025).

O aumento significativo em pesquisas relacionadas a sistemas de gerenciamento de trânsito pode ser considerado um indicador de crescente interesse acadêmico e financeiro no setor, bem como que projetos relacionados a supervisão de vias urbanas são vistos como importantes e eficientes.

Além disso, conforme ilustra o gráfico da Figura 2, a partir do número de patentes registradas de sistemas parecidos com o proposto na Organização Mundial de Propriedade Intelectual (*World Intellectual Property Organization,*) encontrados utilizando a busca "*Traffic Management Systems*", nota-se um crescimento significativo de registros de sistemas de gerenciamento de trânsito no período de 2015 a 2023 com um aumento de aproximadamente 54,64% nos registros.

Figura 2 - Pesquisas relacionadas a patentes de sistemas de gerenciamento de trânsito



Fonte: Os autores (2025).

O interesse econômico, a crescente produção acadêmica aliado a criação de patentes relacionadas a gerenciamento de trânsito, indicam uma necessidade de soluções inovadoras de sistemas de gerenciamento de trânsito com o intuito de reduzir acidentes e prevenir mortes prematuras.

A proposta de uma possível solução para o problema das mortes e acidentes em vias urbanas a partir de modelos computacionais de monitoramento de trânsito se tornou um estudo recorrente e com investimentos notáveis em todo o mundo.

A presente pesquisa tem como objetivo a identificação e detecção de anomalias de tráfego. Segundo Yadav, et al. (2018), anomalias são eventos de curta duração que fogem dos padrões esperados. Para sua detecção, é necessário o monitoramento do tráfego, permitindo classificá-las, a fim de aplicação de uma política específica para determinada situação. Em seu trabalho, Zhankaziev, et al. (2022), discorre sobre o uso de sistemas de monitoramento e gerenciamento de tráfego urbano, podendo ser usado para uma avaliação abrangente da segurança viária, possibilitando o cálculo do risco social de uma via pública, podendo obter estatísticas de acidentes de trânsito de maneira preditiva, a partir da análise de imagens e dados recebidos de detectores de transporte em tempo real.

Dessa forma, o presente estudo propõe uma solução de forma explicativa e quantitativa, com a aplicação de tecnologia de aprendizado de máquina. Acredita-se que, ao desenvolver um modelo de detecção de anomalias no trânsito em um sistema de monitoramento viário, abrem-se possibilidades de economizar recursos de grandes

nações, aprimorar tecnologias já existentes e se tornar um produto útil para assegurar a qualidade de vida no trânsito.

1.3 Objetivos

1.3.1 Objetivo geral

Desenvolver um sistema de detecção de anomalias em vias urbanas e avaliar o seu desempenho em uma base de dados de teste.

1.3.2 Objetivos específicos

Para o desenvolvimento e avaliação do sistema de detecção de anomalias, os seguintes objetivos específicos foram estabelecidos:

- Análise da teoria relacionada a modelos de aprendizado de máquina e métricas de avaliação de desempenho aplicadas a problemas de visão computacional;
- Análise da bibliografia relacionada a *track 4* do *AI City Challenge* de 2021;
- Realizar a extração de trajetórias utilizando Ultralytics YOLO;
- Implementação de sistemas de detecção de anomalias utilizando WGAN-GP;
- Testes dos sistemas utilizando a base de dados da *track 4* do *AI City Challenge* de 2021;
- Avaliação dos *frameworks* de detecção de anomalias em relação aos principais colocados da *track* 4 do *AI City Challenge* de 2021.

1.4 Metodologia da pesquisa

1.4.1 Enquadramento da pesquisa

A pesquisa explicativa conduzida neste trabalho, conforme descrito por Azevedo e Ensslin (2020), tem como propósito analisar os fatores ou anomalias relacionadas a acidentes de trânsito. Quanto à sua natureza, classifica-se como aplicada, pois busca propor uma solução prática para o problema do aumento de acidentes de trânsito.

Foi utilizada a abordagem quantitativa, focada no desenvolvimento e avaliação do sistema de detecção de anomalias aplicado ao trânsito urbano. Inicialmente, o problema foi definido e, havendo uma revisão bibliográfica para identificar os modelos de aprendizado de máquina mais adequados. Foram utilizadas a base de dado do *AI City Challenge* de 2021, disponível na literatura. Posteriormente, o modelo apropriado de aprendizado de máquina foi desenvolvido, e realizada uma pesquisa experimental, treinando o modelo com os dados e testando-o para avaliação de sua precisão e eficiência.

1.4.2 Desenvolvimento e análise do modelo de detecção de anomalias

O modelo de detecção de anomalias foi implementado e avaliado em condições variadas de trânsito, utilizando a base de dados da *Track* 4 do *Ai City Challenge* de 2021. Os resultados foram analisados quantitativamente em relação ao dos melhores colocados, a fim de verificar a capacidade do modelo em detectar anomalias em vias urbanas, visando sua possível aplicação na otimização do trânsito e rotas. As métricas de desempenho incluem precisão, velocidade e sensibilidade.

1.5 Contribuição do trabalho

A presente pesquisa busca contribuir para a área de segurança viária por meio do desenvolvimento de um sistema de detecção de anomalias no trânsito utilizando técnicas de aprendizado de máquina e visão computacional. A principal inovação está na integração das abordagens mais bem-sucedidas do *AI City Challenge* de 2021, mais especificamente da *track 4*, detecção de anomalias no trânsito, uma competição de relevância acadêmica e internacional, com participação de equipes de companhias com alto investimento em inteligência artificial.

A partir dessa base, propõe-se o desenvolvimento de um *framework* próprio, inspirado nas soluções de destaque do desafio, mas com inovações no processo de geração de dados sintéticos. Será investigado o uso de redes adversárias generativas Wasserstein com penalidade de gradiente (WGAN-GP) como estratégia para ampliar a diversidade e a robustez dos dados de treinamento, buscando avaliar sua eficácia em comparação com outras abordagens de aprendizado profundo aplicadas à detecção de anomalias.

Espera-se que o *framework* desenvolvido seja capaz de identificar eventos anômalos de forma automatizada e eficiente, contribuindo para que gestores públicos e órgãos de trânsito possam tomar decisões mais embasadas, mitigar situações de risco e otimizar recursos voltados à segurança viária.

A contribuição deste sistema se alinharia aos esforços globais e nacionais para a redução de acidentes de trânsito, sendo uma solução tecnológica que pode auxiliar na identificação de padrões críticos de risco, por conta disto, acredita-se que o aumento de investimentos na área seria um demonstrativo do potencial econômico do projeto. O *framework* proposto seria capaz de detectar eventos anômalos de forma automatizada e eficiente, desta forma, gestores públicos e órgãos de trânsito poderão tomar decisões mais embasadas para mitigar situações de perigo, otimizando o uso de recursos e promovendo maior segurança nas vias.

Além do impacto na saúde pública e econômico, este estudo visa agregar valor à literatura acadêmica ao explorar a viabilidade de arquiteturas avançadas de aprendizado profundo na detecção de anomalias de tráfego. A crescente relevância dos sistemas inteligentes de monitoramento viário, evidenciada pelo aumento na publicação de pesquisas e registros de patentes na área, reforça a importância de soluções inovadoras que possam ser implementadas em larga escala.

Dessa forma, esta pesquisa não está apenas propondo um modelo para a supervisão de trânsito, mas também um estudo capaz de abrir possibilidade para futuras investigações em cima do mesmo tema e com aplicações similares sobre a inteligência artificial na segurança viária.

1.6 Estrutura do trabalho

Este trabalho, foi dividido em cinco capítulos. Ao final, a pesquisa apresenta a conclusão do estudo realizado e de seus resultados.

A seguir, é apresentada a descrição dos capítulos.

O capítulo 1, a introdução, apresenta o contexto da pesquisa, destacando a relevância da segurança viária e os desafios enfrentados no controle e monitoramento do trânsito, como os altos índices de acidentes de trânsito. É realizada também a análise das iniciativas existentes para mitigação desses riscos e a justificativa para o desenvolvimento de um sistema de detecção de anomalias. É neste capítulo que são expostas as contribuições esperadas deste estudo para a área.

A fim de garantir a melhor compreensão dos temas tratados no trabalho, o segundo capítulo aborda o embasamento teórico, realizando um estudo dos conceitos fundamentais desta pesquisa. São discutidos aspectos relacionados à mobilidade urbana, segurança viária e o papel da tecnologia no monitoramento de tráfego, além de uma introdução às principais técnicas de aprendizado de máquina e visão computacional aplicadas à detecção de anomalias.

O Capítulo 3 apresenta a revisão da literatura do tema, analisando estudos e soluções recentes voltados à detecção de anomalias no trânsito. São exploradas pesquisas acadêmicas, métodos empregados em competições como o AI City Challenge e o uso de redes generativas para ampliação de bases de dados, assim como a discussão de seus resultados obtidos.

A descrição do sistema de detecções de anomalias proposto, está presente no capítulo 4, explicitando a concepção do modelo de detecção de anomalias e os fundamentos por trás das técnicas utilizadas. São descritas as estratégias adotadas para a construção do sistema, seus métodos e ferramentas, bem como os desafios e diferenciais da abordagem escolhida.

O capítulo 5 apresenta os trabalhos futuros, discutindo a aplicação prática da revisão bibliográfica feita, o modelo que foi implementado e os resultados esperados.

2 EMBASAMENTO TEÓRICO

O objetivo deste capítulo é explicar os princípios teóricos e os componentes fundamentais que embasam a análise realizada neste estudo. São apresentados os conceitos, metodologias e referências que fundamentam o aprofundamento do tema, oferecendo o suporte necessário para entender os resultados e as opções técnicas escolhidas. Nesta seção, procura-se definir os fundamentos teóricos que serão empregados nas etapas seguintes.

2.1 Visão computacional

A visão computacional é uma subárea da inteligência artificial que através de diversos modelos matemáticos busca capacitar sistemas digitais a interpretar e compreender o mundo visual a partir de imagens estáticas ou dinâmicas (vídeos).

Nesse contexto Szeliski (2021), explica que a visão computacional tem como objetivo descrever o mundo a partir de imagens e reconstruir sua forma, iluminação e distribuição de cores. Além disso, embora pareça simples para seres sencientes, é uma tarefa extremamente desafiadora para algoritmos computacionais. O autor ressalta que na visão computacional, tenta-se fazer o inverso isso é, descrever o mundo em uma ou mais imagens e reconstruir suas propriedades e que enquanto os humanos e animais são capazes de fazer isso sem esforço, algoritmos de visão computacional estão sujeitos a erros.

Esses modelos são baseados em técnicas avançadas de processamento de imagens, aprendizado de máquina e algoritmos capazes de extrair informações relevantes de dados visuais. Essa tecnologia é extremamente complexa e está em constante evolução. Além disso, Prince (2012) diz que além de ser uma tarefa difícil a visão computacional nos últimos 40 anos está em constante pesquisa e desenvolvimento e está longe de se construir um modelo de propósito geral. destaca que, apesar dos avanços das últimas décadas, a visão computacional ainda está longe de alcançar um modelo de propósito geral, permanecendo como uma área em contínuo desenvolvimento.

Dessa forma, a visão computacional segue como uma área desafiadora, capaz de transformar a interação entre o mundo digital e físico.

2.2 Processamento de imagens

O processamento de imagens é uma área dedicada à aplicação de métodos e algoritmos para melhorar e modificar imagens digitais. De acordo com Gonzalez E Woods (2008), uma imagem digital é descrita como uma função bidimensional f(x,y), onde x e y são coordenadas espaciais, e a amplitude de f em um determinado ponto especifico representa a intensidade ou nível de cinza correspondente. O resultado é uma imagem digital, quando essas variáveis são representadas por valores finitos e discretos. É dito que cada imagem digital é composta por um número finito de elementos, os quais tem um local e valor particular. Esses elementos são comumente chamados de *Pixels*

Além disso, Gonzalez e Woods (2008) ressaltam que não há uma fronteira bem definida entre visão computacional e processamento de imagem. Porém, é interessante considerar 3 tipos de processos: Baixo, médio e alto nível.

O processo de baixo nível tem como entrada e saída uma imagem como por exemplo a aplicação de realce de contraste e detecção de bordas. Já um processo de nível médio apresenta o mesmo tipo de entrada, porém sua saída são atributos, como a resolução da imagem. Já o processo de alto nível envolve realizar funções cognitivas, normalmente associadas com a visão humana, como a classificação de objetos.

Entretanto, conforme destacado por Gonzalez e Woods (2008), as técnicas tradicionais de processamento digital de imagens apresentam limitações quando aplicadas a cenários complexos ou dinâmicos, como variações de iluminação, ruído e oclusões.

Essa limitação motivou o surgimento de abordagens baseadas no aprendizado de máquinas e redes neurais, capazes de aprender padrões diretamente dos dados. Essa evolução marca a transição entre o processamento de imagem e o aprendizado de máquina, tema abordado na seção 2.3.

2.3 Aprendizado de máquina

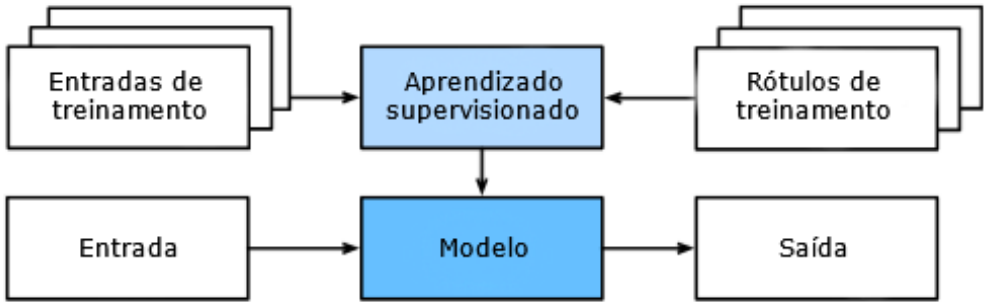
Essa ciência, também conhecida como *machine learning*, é um campo multidisciplinar artificial focada na construção de algoritmos capazes de identificar padrões em dados e prever ou decidir com base neles. Mitchell (1997) define

aprendizado de forma ampla incluindo qualquer programa no qual seu desempenho em uma tarefa melhore através da experiência.

Dessa forma, Szelisk (2021) afirma que “Técnicas de aprendizado de máquina sempre desempenharam um papel importante e muitas vezes central no desenvolvimento de algoritmos de visão computacional”. Assim é possível concluir a importância desse campo em trabalhos de visão computacional.

Os algoritmos de *machine learning* podem ser categorizados como supervisionados (*supervised),* no qual os dados de entrada com saídas rotuladas são fornecidos ao algoritmo de aprendizado como visto na Figura 3, ou não supervisionados (*unsupervised*), onde amostras estatísticas são fornecidas sem quaisquer saídas rotuladas. De acordo com Szeliski (2021).

Figura 3 - Aprendizado supervisionado



Fonte: Adaptado de Szeliski, 2021.

2.3.1 Aprendizado supervisionado

Em termos práticos, o aprendizado supervisionado envolve alimentar o algoritmo de aprendizado com pares de entradas e os valores correspondentes de sua saída. Esse algoritmo deve ser capaz de ajustar os parâmetros do modelo para maximizar a compatibilidade entre a saída prevista pelo modelo e a esperada (Szeliski, 2021).

O autor ainda afirma que é possível definir a tarefa como **classificação** quando sua saída é um conjunto discreto de rótulos ou **regressão** se a saída for um conjunto contínuo.

De forma parecida, Russell et al. (2010) descreve esse tipo de aprendizado como uma busca de hipóteses através do espaço, sendo necessário fornecer um

conjunto de testes de exemplos do conjunto de treinamento para medir a precisão de uma hipótese.

2.3.2 Aprendizado não supervisionado

Como citado anteriormente, esse tipo de aprendizado é utilizado em algoritmos onde não há rótulos definidos em pares de dados de entradas e saídas. Szeliski (2021) reforça que o aprendizado não supervisionado é utilizado em aplicações nas quais se deseja caracterizar um conjunto de dados.

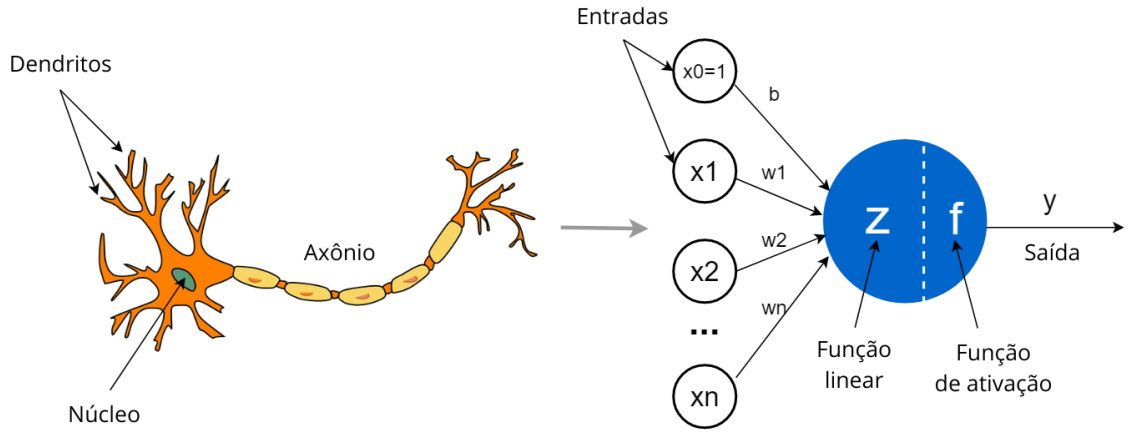
2.4 Redes neurais

As redes neurais são modelos computacionais inspirados em como o cérebro humano funciona, elas são compostas por unidades chamadas neurônios organizados em camadas (*layers*).

> “O trabalho com redes neurais artificiais, comumente chamadas de redes neurais. Foi motivado desde o começo pelo reconhecimento da capacidade do cérebro humano de computar de uma forma diferença de um computador convencional”. (Haykin, 2009, p. 1, tradução dos autores)

Além disso, Haykin (2009) discute a complexibilidade cerebral humana e como ele é não-linear e processa informações de forma de paralela com sua estrutura organizada em neurônios capazes de realizar reconhecimento de padrões, percepção e controle motor muitas vezes mais rápido que o computador mais veloz, atualmente. Ele também diz que o cérebro dos seres vivos, através da experiencia, desde o nascimento já tem uma estrutura considerável e a habilidade de construir suas próprias regras de comportamento e como esse órgão se desenvolve durante a vida para se adaptar ao ambiente em sua volta. Ele usa isso como uma analogia ao funcionamento das redes neurais artificiais. Essa comparação entre neurônio real e artificial, é ilustrada na Figura 4

Figura 4 - Neurônio real x neurônio artificial



Fonte: Adaptado de Pramoditha, (2021).

No contexto de inteligência artificial, Szeliski (2021) defende que diferentemente de outros métodos de aprendizado de máquina, os quais necessitam de diversos estágios de pré-processamento. As redes neurais são processos que vão de *pixels "crus"* para a saída de dados desejada. Além disso, essas redes são grafos *feedforward* (Uma possível tradução seria a ideia de que a sinal de saída de um neurônio pode ser o de entrada de outro) compostos por milhares de unidades chamadas de "neurônios", como citado acima, interconectados.

2.4.1 Pesos em redes neurais

Segundo Haykin (2009), os neurônios são responsáveis por efetuar somas ponderadas de suas entradas. Dessa forma, o peso utilizado na soma é definido como uma característica da conexão, ou sinapse (analogia ao neurônio humano). Ele diz que para um número *j* de entradas conectadas a um neurônio *k* haverá um peso que representa a "força da conexão" de cada entrada ao neurônio. Ele pode ser chamado de peso sináptico w_{kj}, onde o subscrito *k* refere-se ao neurônio em questão e o subscrito *j* à entrada cuja sinapse esse peso se refere. É também ressaltado que, o peso sináptico de um neurônio artificial pode incluir tanto valores positivos quanto valores negativos. Assim, apresenta a seguinte equação para o cálculo dessa soma ponderada.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \qquad (2.4.1)$$

Na equação (2.4.1) x_j são os sinais de entrada, w_{kj} os pesos sinápticos de um neurônio *k* e u_k o resultado da soma ponderada.

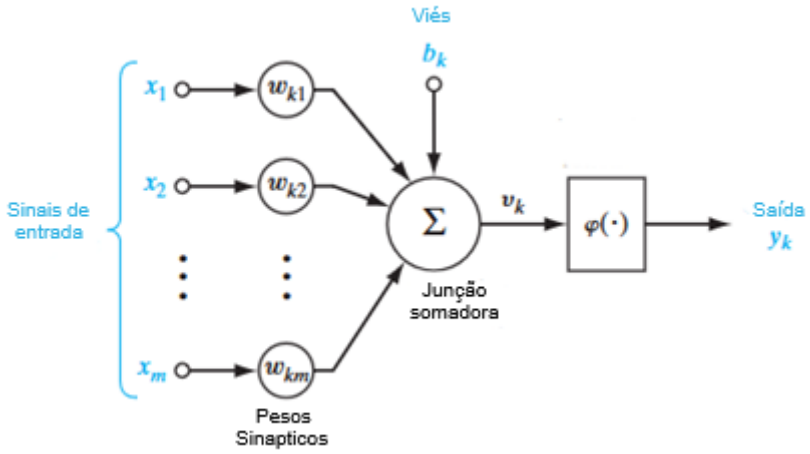
2.4.2 Função de ativação

Para Haykin (2009) essa função é definida como “A função de ativação pode ser também referida como uma função de compressão, isso é, ela comprime a amplitude permissível na faixa de saída do sinal para valores finitos”. Além disso, (Haykin, 2009) mostra que essa função trabalha em conjunto com a soma ponderada das entradas em um neurônio *k*, agora com um viés b_k considerado. Dessa forma, a saída y_k do neurônio *k* pode ser calculada da seguinte maneira:

$$y_k = \varphi(u_k + b_k) \tag{2.4.2}$$

Na equação (2.4.2), u_k é a soma ponderada calculada pela equação (2.4.1), b_k o viés no neurônio k, $\varphi(\cdot)$ a função de ativação, $\cdot$ o argumento da função de ativação e y_k o sinal de saída do neurônio *k.* A Figura 5 apresenta o modelo de um neurônio artificial, baseado nas equações (2.4.1) e (2.4.2).

Figura 5 - Modelo de um neurônio não linear 2



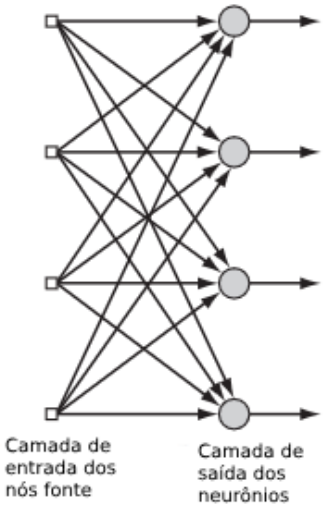
Fonte: Adaptado de Haykin (2009).

2.4.3 Arquitetura de uma rede neural

Para Haykin (2009), uma rede neural, os neurônios são organizados na forma de camadas. Ele diz que essa rede pode apresentar apenas camada (sua forma mais simples) ou multicamadas. Essas camadas são chamadas de *layers*. Uma rede

de apenas uma camada, como a representada na Figura 6, é chamada de *single-layer network* na literatura estrangeira. Abaixo um exemplo de rede com camada única.

Figura 6 - Exemplo de rede neural *feedforward* com camada única



Fonte: Adaptado de Haykin (2009).

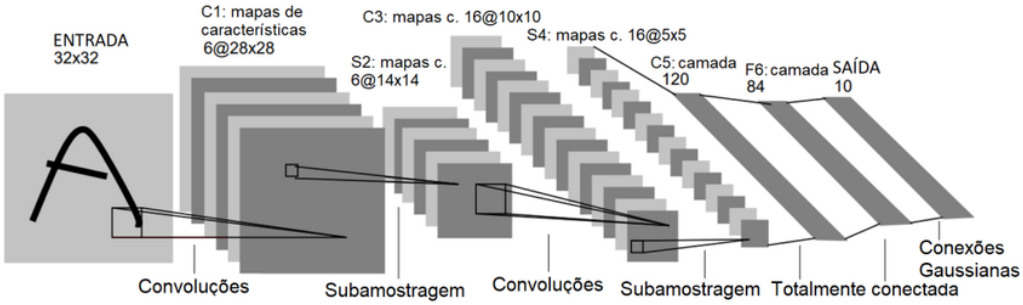
De maneira análoga, Stanford (2024) classifica a arquitetura como uma coleção de neurônios presentes em um grafo acíclico e que esses neurônios são, regularmente, organizados em camadas distintas e normalmente representados em camadas totalmente conectadas (*fully-connected layer*), isso é cada neurônio em uma camada entre dois outros *layers* é conectado a todos os neurônios dessas camadas adjacentes e neurônios na mesma camada nunca se conectam.

2.5 Redes neurais convolucionais

Em seu livro, Goodfellow et al. (2016) define as redes neurais convolucionais, (*convolutional neural networks)*, ou CNN, como um tipo de rede neural especializado para processar dados em uma topologia parecida com malhas. Além disso, explicam que é uma técnica que obtém ótimos resultados práticos e funciona bem com aprendizado que envolve imagens, pois os *pixels* dessa imagem estão organizados em uma malha bidimensional. O nome “convolucional”, como descrito por Goodfellow et al. (2016), é uma operação linear especializada, e as redes convolucionais são *neural networks* que utilizam convolução no lugar da multiplicação de matrizes em pelo menos uma de suas camadas.

Szeliski (2021) defende que o uso de redes convolucionais com multicamadas é o componente mais importante para o processamento de imagens e visão computacional. Ele explica, que diferente da arquitetura de uma rede neural simples, ao invés dos neurônios serem conectados a todos os neurônios em camadas adjacentes, a CNN, com a arquitetura representada na Figura 7, organiza cada *layer* em um mapa de atributos, é utilizada a analogia de planos paralelos ou canais.

Figura 7 - Arquitetura de uma CNN



Fonte: Adaptado de Szeliski (2021).

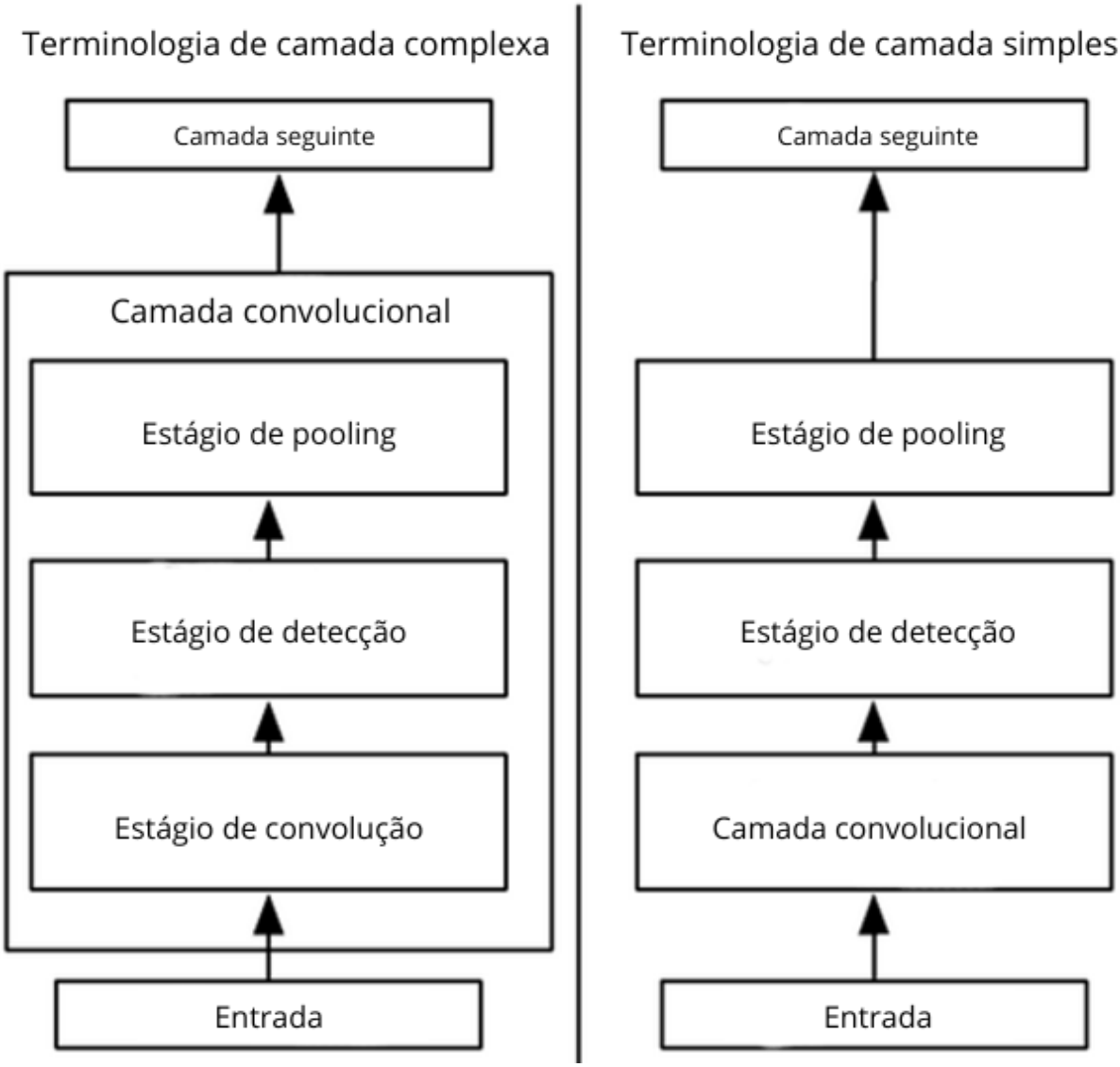
Já a soma ponderada (item 2.4.1) é efetuada apenas de forma local e os pesos são idênticos para todos os *pixels.* Redes neurais convolucionais combinam de forma linear as ativações (item 2.4.2) para cada canal de sinal de entrada em um *layer* anterior e utilizam diferentes matrizes de convolução para cada canal de saída, explica Szeliski (2021) que apresenta a seguinte equação para cálculo da soma ponderada.

$$s(i,j,c_2) = \sum_{c_1 \epsilon \{C_1\}} \sum_{(k,l) \in N} w(k,l,c_1,c_2) a(i+k, j+l, c_1) + b(c_2) \qquad (2.5)$$

Na equação (2.5) a representa as ativações da camada anterior, C_1 são os canais de entrada, c_1 cada item presente no conjunto C_1, C_2 são os canais de saída, c_1 cada item presente no conjunto C_2.

Goodfellow et al. (2016) define que um *layer* tipico de uma CNN consiste em 3 estágios, representados na Figura 8. Primeiro, a camada realiza diversas convoluções em paralelo para produzir um conjunto de ativações lineares. Depois, cada ativação linear passa por uma função de ativação não linear e por fim, é utilizada uma função de *pooling* para modificar a saída da camada adiante.

Figura 8 – Terminologias de camadas simples e complexa de CNN

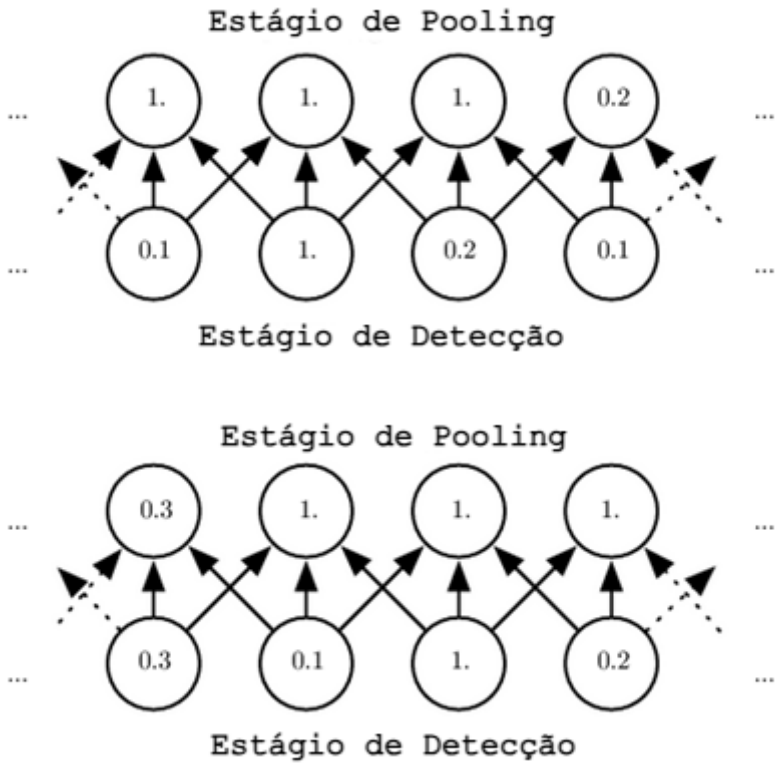


Fonte: Adaptado de Goodfellow et al. (2016).

2.5.1 *Pooling*

Para Goodfellow et al. (2016), uma função *pooling* realiza a mudança da saída de uma rede em determinado local com uma base estatística das saídas vizinhas. De acordo com eles, é uma técnica necessária para manter uma representação invariante a pequenas translações da entrada. Isso é, ao translacionar a entrada, os valores da maioria das saídas do *pooling* irão se manter. A Figura 9 representa um exemplo dessa técnica:

Figura 9 - Exemplo de pooling



Fonte: Adaptado de Goodfellow et al. (2016).

2.6 Redes adversárias generativas

Creswell et al. (2018) define que “Modelos generativos aprendem a capturar a distribuição estatística dos dados de treinamento, nos permitindo sintetizar amostras da distribuição aprendida”. Além disso, Creswell et al., (2018) diz que o objetivo principal dessas redes é a estimação de densidade, isso é, conseguir representar uma distribuição de probabilidade dos dados observados no mundo real.

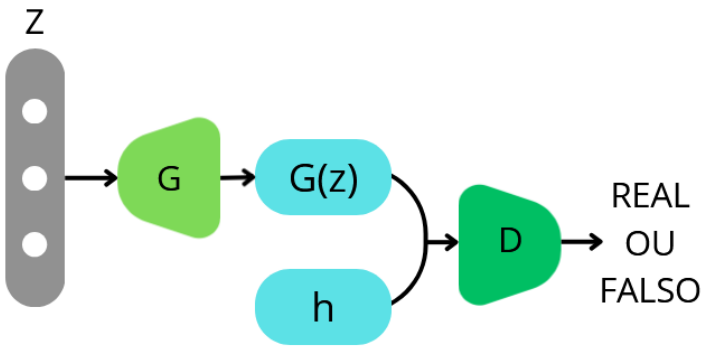
Nessa linha, Goodfellow et al. (2014) complementa que a abordagem de aprendizado profundo nesse contexto é encontrar modelos capazes de representar essas distribuições sobre diversos tipos de dados (Desde imagens até formatos de onda de áudio) de forma eficaz.

Goodfellow, et al. (2014) e Mirza, et al. (2014) introduzem as Redes Adversárias Generativas (GANs) como uma estrutura para treinamento de modelos generativos. Ela é baseada em um processo adversário, no qual dois modelos são treinados de maneira simultânea em competição. Assim como na estrutura da Figura 10, um modelo generativo (G), cuja função é capturar a distribuição de dados, e um modelo

discriminativo (D), o qual estima a probabilidade de uma amostra ser proveniente dos dados de treinamento ao invés de G.

Goodfellow et al. (2016) complementa que “Redes adversárias generativas são baseadas em um cenário de um jogo teórico no qual a rede geradora deve competir contra um adversário, o Discriminador”

Figura 10 – Exemplo de estrutura GAN



Fonte: Elaborado pelos autores com base em Goodfellow et al. (2016)

Segundo o trabalho de Goodfellow et al. (2016), explica-se que a rede geradora produz as amostras $h = g(z; \theta^{(g)})$, onde, h é a amostra gerada, z uma amostra selecionada aleatoriamente, $g(z)$ a função geradora e $\theta^{(g)}$ os parâmetros de $g(z)$.

Em Goodfellow et al. (2014) é explicado que a função $g(z)$ é uma rede neural multicamadas, e que de forma parecida o Discriminador, uma rede neural multicamadas, $d(h; \theta^{(d)})$ representa a probabilidade de x pertencer aos dados do Gerador.

2.6.1 Treinamento das redes adversárias generativas

De acordo com Goodfellow et al. (2014) o Discriminador é treinado para maximizar essa probabilidade de assimilar os rótulos de ambas as amostras de treino e geradas e de forma simultânea o Gerador é treinado para minimizar $log(1 - d(g(z))$. Dessa forma pode ser definida a função descrita na equação 2.6.1:

$$\min_{G} \max_{D} V(d, g) = E_{x \sim P_{\text{data}}} [\log d(x)] + E_{z \sim p_{modelo}} [\log(1 - d(h))] \quad (2.6.1)$$

O primeiro termo representa o valor esperado de $log(d(x))$ de uma amostra x retirada da distribuição de dados reais. Já o segundo termo o valor esperado de $\log(1 - d(h))$ para uma amostra $h = g(z; \theta^{(g)})$.

Creswell et al. (2018) explica que a otimização desse treinamento é realizada alternando a otimização dos parâmetros do Discriminador e Gerador, no intuito de alcançar um ponto de equilíbrio (Equilíbrio de Nash).

Nessa linha, Goodfellow et al. (2014) explica que o treinamento GAN exige que se encontre um ponto de sela que representa um equilíbrio ideal, no qual as amostras do Gerador são iguais as amostras reais e o Discriminador atinge sua máxima confusão, prevendo 50% para todas entradas.

2.6.2 Wasserstein GAN

Wasserstein GAN (WGAN) é um algoritmo alternativo ao treinamento GAN tradicional. Esse algoritmo foi introduzido por Arjovsky et al. (2017). Nele é apresentado o conceito de distância de Wasserstein, que mede quão longe as distribuições real e geradas estão. A WGAN modifica a GAN com o intuito de minimizar tal distância.

Em *Optimal Transport,* Villani (2009) descreve a distância de Wasserstein como uma forma de reduzir o grau de dificuldade de transporte entre pontos, como um custo de transporte ótimo entre duas medidas. Essa distância é definida como,

$$W_p(\mu, \nu) = \left[\inf_{\pi \in \prod(u,v)} \int_X d(x, y)^p d\pi(x, y) \right]^{1/p} \quad (2.6.2)$$

onde $p \in [1, \infty[$ pertencente ao espaço (X, d). p representa a ordem da distância entre μ e ν.

Arjovsky et al. (2017) explica que o WGAN se destaca, em comparação com o GAN padrão, ao utilizar a distância de Wasserstein como métrica central. Essa distância, é continua e diferencial em quase todos os pontos o que permite um treinamento com menor propensão a erros.

2.6.3 Wasserstein GAN com gradiente de penalidade

O modelo WGAN-GP é uma variação da arquitetura *Generative Adversarial Network* (GAN) fundamentada na métrica de Wasserstein, introduzida para resolver as limitações do WGAN original quanto à estabilidade do treinamento. Conforme Guo, Liu e Yang (2023), esta abordagem substitui o recorte de pesos utilizado no modelo tradicional pela penalização do gradiente. Essa modificação torna o processo de otimização mais controlável e reduz a instabilidade decorrente de gradientes nulos. O termo de penalidade introduzido atua diretamente na norma do gradiente do discriminador, assegurando que a diferença entre amostras reais e geradas siga o comportamento esperado pela métrica de Wasserstein.

Segundo Liu, Si e Wang (2025), a inclusão do termo de penalidade de gradiente permite que o WGAN-GP evite o colapso de modos e o problema de não convergência presentes em redes adversariais tradicionais. Essa característica possibilita a geração de dados sintéticos de alta fidelidade, inclusive sob condições de conjuntos amostrais limitados, como observado em aplicações industriais. A formulação proposta por Liu, Si e Wang (2025) inclui ainda mecanismos complementares, como campos aleatórios condicionais e módulos de atenção, para aperfeiçoar a aprendizagem espacial e temporal das amostras geradas, evidenciando a flexibilidade estrutural do WGAN-GP para diferentes domínios.

Zhang et al. (2022) integram o WGAN-GP a uma rede *Long Short-Term Memory* (LSTM) na modelagem de trajetórias aeronáuticas em quatro dimensões (DGPNM). Nesse trabalho, o WGAN-GP é empregado como módulo gerador de dados, responsável por ampliar o conjunto de treinamento com amostras artificiais, enquanto o LSTM é utilizado como módulo preditor. O uso do WGAN-GP nessa arquitetura permite a produção de dados sintéticos temporalmente consistentes e próximos da distribuição real das trajetórias, reduzindo o sobreajuste e aprimorando o desempenho da previsão. O modelo resultante é capaz de capturar as dependências dinâmicas e temporais dos dados de voo, contribuindo para maior precisão na estimativa das trajetórias.

2.6.3.1 *Long Short-Term Memory* (LSTM) associado a WGAN-GP

O modelo *Long Short-Term Memory* (LSTM) é uma estrutura de rede neural recorrente projetada para lidar com dependências de longo prazo em sequências temporais. Zhang et al. (2022) descrevem o LSTM como composto por três portas principais (1) de entrada, controlando a incorporação de novas informações; (2) de esquecimento, regulando o descarte de estados anteriores; e (3) de saída, definindo o conteúdo a ser transmitido para a próxima etapa temporal.

Essa estrutura possibilita que o modelo retenha informações relevantes por períodos prolongados, evitando problemas de desaparecimento ou explosão de gradientes. No contexto da predição de trajetórias aéreas, o LSTM é aplicado para capturar a relação entre variáveis temporais sucessivas, garantindo a continuidade e coerência espacial das previsões quando associado ao módulo de geração de dados baseado em WGAN-GP.

2.7 Detecção de objetos

Segundo Szeliski (2021) tarefas de análise de imagem são mais efetivas quando utilizadas um detector, isso é, um algoritmo capaz de rapidamente encontrar padrões nessas imagens. E que para algoritmos de detecção de objetos de classes diferentes atualmente é utilizado a tecnologia de redes neurais. A Figura 11 mostra exemplos de detecções de veículos.

Figura 11 - Exemplo de detecção de objetos

Fonte: He et al. (2019)

2.7.1 Extração de máscara

Szeliski (2021) define as máscaras como um rotular cada pixel por classe, isso é, ser capaz de isolar regiões de interesse para realizar análises posteriores. Um exemplo de uso dessa técnica é a capacidade de remoção do plano de fundo de uma imagem. Zivkovic (2004) propõe um modelo adaptativo capaz de efetuar essa remoção.

O uso desse modelo se define um período de tempo T e para o tempo t se tem $X_T = \{x^{(t)}, \dots, x(t-T)\}$, onde para cada nova amostra é calculada a probabilidade de ela pertencer ao plano de fundo (BG) $\hat{p}(\vec{x}|X_t, BG)$. Zivkovic (2004) diz que é necessário estimar como $\hat{p}(\vec{x}|X_t, BG + FG)$ devido a chance de alguns valores pertencerem ao plano de frente (FG). Dessa forma, como mostrado na equação 2.7.1, é utilizado o GMM com M componentes:

$$\hat{p}(\vec{x}|X_t, BG + FG) = \sum_{m=1}^{M} \hat{\pi}_m \, N(\vec{x}; \hat{\mu}_m, \hat{\sigma}_m I) \tag{2.7.1}$$

Na equação (2.7.1), $\hat{\mu}_1, \dots, \hat{\mu}_M$ representa a estimativa das médias e $\hat{\sigma}_1, \dots, \hat{\sigma}_M$ as estimativas que descrevem os componentes Gaussianos. Assume-se que a matriz de covariância é diagonal e a matriz de Identidade I tem as corretas proporções. Os pesos $\hat{\pi}_m$ são valores positivos e $\sum_{m=1}^{M} \hat{\pi}_m = 1$.

2.7.2 Detecção de vários Objetos

Também conhecida como *Multiple Object Tracking* (MOT), de acordo com Luo et al. (2021) esse algoritmo consiste em uma estimativa de um problema com várias variáveis. Isso é, é definido um objeto s_i^n que define o estado do n-ésimo objeto no i-

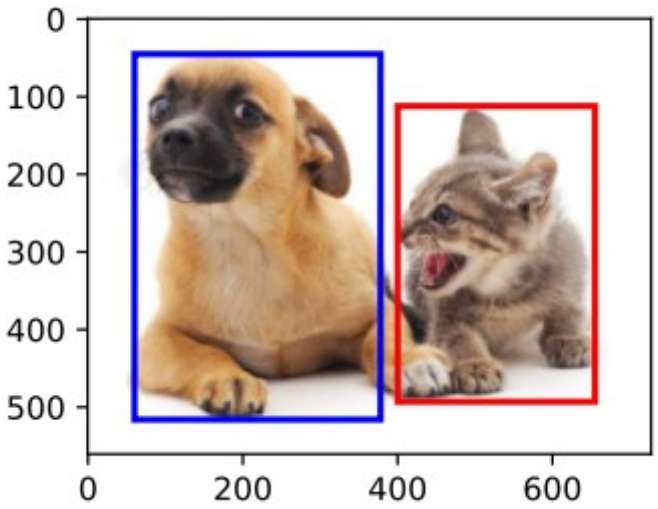
ésimo frame, $S_i = (s_i^1, s_i^2, s_i^3, \ldots, s_i^{Mt})$ determina o estado de todos os objetos *Mt* no frame i.

2.7.3 Caixa delimitadora

No contexto de detecção de objetos, uma caixa delimitadora ou *bounding box* é utilizada para descrever um objeto. Zhang et al. (2021) explica que essas caixas são formadas por um retângulo bidimensional com coordenadas x e y. Também podem ser representadas por um coordenadas (x, y) que representam o centro da caixa delimitadora com w sua largura e h sua altura.

A Figura 12 representa um exemplo do funcionamento de caixas delimitadoras, detectando animais diferentes ao usar azul para o cachorro e vermelho para o gato.

Figura 12 - Exemplo de caixas delimitadoras detectando animais

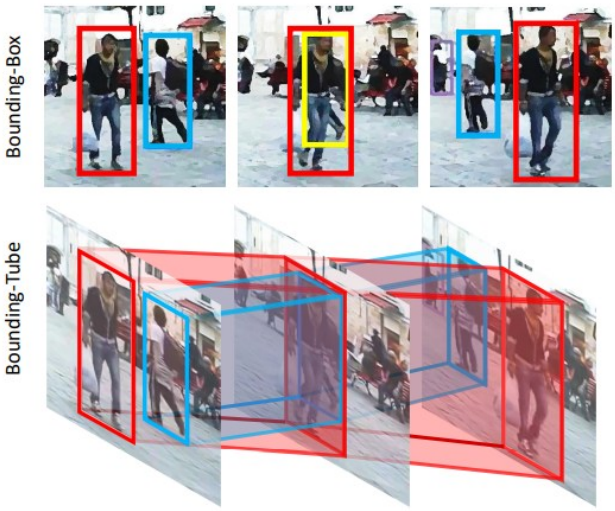


Fonte: Zhang et al. (2021).

2.7.4 Tubos delimitadores

Pang et al. (2020) propõe um modelo no qual é criado um tubo delimitador, ou *bounding tube*. Diferente das caixas delimitadoras que são bidimensionais esses tubos são tridimensionais, são adicionadas informações temporais que melhoram a localização espacial do objeto. Com essas informações, o modelo é capaz de prever a próxima posição do objeto e com isso evitar a não detecção devido a oclusões.

Figura 13 - Comparação de caixa delimitadoras com tubo delimitador



Fonte: Pang et al. (2020).

2.8 Avaliação de desempenho

Neste capítulo serão apresentadas algumas métricas de avaliação de desempenho que serão utilizadas no contexto deste trabalho.

2.8.1 Interseção sobre união

Essa métrica, no inglês *Intersection over Union (IoU)*, é utilizada na visão computacional com o intuito de avaliar a precisão da detecção de objetos. Ela foi popularizada em desafios como o PASCAL VOC Everingham et al. (2010) onde serviu como critério para calcular a precisão média das detecções.

É calculada a sobreposição entre o predito pelo modelo e o real da seguinte forma:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{2.8.1}$$

Na equação (2.8.1), $B_p \cap B_{gt}$ representa a interseção da *bounding box* predita e a região verdadeira da *bounding box* e $B_p \cup B_{gt}$ a sua união.

Valores de IoU próximos a 1 indicam uma maior sobreposição, isso é, uma predição mais precisa. De forma oposta, valores próximos a 0 indicam uma predição ruim.

2.8.2 *F1-Score*

Segundo Sitarz (2022), essa é uma métrica utilizada para classificação de classificadores binários. É calculada através da média harmônica da precisão e da revocação. É calculada através da equação (2.8.2).

$$F_1 = \frac{TP}{TP+0.5(FP+FN)} \tag{2.8.2}$$

Na equação (2.8.2), TP são os verdadeiros positivos, FP os falsos positivos e FN os falsos negativos.

2.8.3 Área sob a curva ROC

De acordo com o trabalho de Fawcett (2006), *Receiver Operating Characteristic* (ROC) é um gráfico no qual está representada a taxa de verdadeiros positivos em função da taxa de falsos positivos. A área sob essa curva, denominada de AUROC, define a capacidade do modelo de distinguir as classes. Essa métrica vária de 0 a 1, valores próximos a 1 indicam um modelo altamente discriminativo

2.8.4 RMSE

A *Root Mean Squared Error* (RMSE), é a raiz quadrada da função Erro médio quadrático (MSE). Dawani (2020), explica que são as funções que quantificam o erro entre o previsto pelo modelo e a realidade, eles avaliam quão mal um modelo está performando. A MSE é calculada conforme a equação 2.8.3.

$$MSE = \frac{1}{N}\sum_i |\hat{y}_i - y_i|^2 \tag{2.8.3}$$

Na equação 2.8.3, é calculada a média da diferença absoluta de N amostras reais e n valores previstos. O conjunto $y(yi, i = 1,2,\dots,N)$ representa os dados reais e o conjunto $\hat{y}(yi, i = 1,2,\dots,N)$ as previsões do modelo. Assim é possível calcular a RMSE, como visto na equação 2.8.4

$$RMSE = \sqrt{\frac{1}{N}\sum_i |\hat{y}_i - y_i|^2} \tag{2.8.4}$$

2.9 Algoritmos de otimização combinatória

A otimização combinatória foca em encontrar a solução ótima dentro de um conjunto finito de possibilidades, sendo o "problema de atribuição" um dos seus exemplos clássicos (Kuhn, 1955). Este problema é formalmente descrito como a busca pela melhor atribuição de n pessoas a n trabalhos, de modo que a soma das pontuações obtidas nessa atribuição seja a maior possível (Kuhn, 1955). Para solucionar essa questão, foi desenvolvido o Método Húngaro, um algoritmo de otimização que utiliza ideias latentes de matemáticos húngaros (Kuhn, 1955).

O Algoritmo Húngaro é especificamente um método de otimização usado para resolver problemas de atribuição, geralmente buscando minimizar o custo total em uma matriz de custo n×n (Hu et al., 2024). O método aborda o problema de atribuição geral, que utiliza matrizes de classificação (Kuhn, 1955), e o soluciona tratando-o como um problema de correspondência ideal em um gráfico bipartido (Hu et al., 2024).

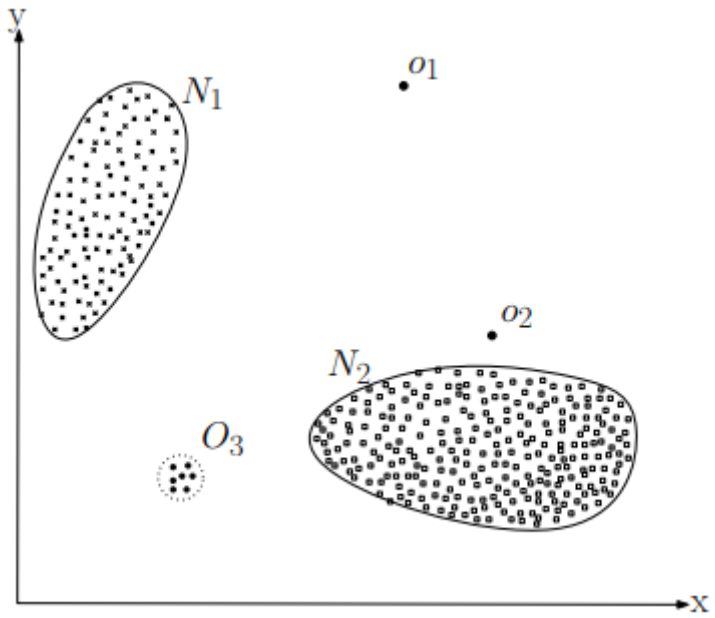
2.10 Detecção de anomalias

A detecção de anomalias é uma área de estudo que visa identificar padrões que tem um comportamento significativamente diferente do esperado em um conjunto de dados. Tais desvios são chamados de anomalias.

> “Detecção de anomalias se refere ao problema de encontrar padrões que não seguem um comportamento esperado em um conjunto de dados. Esses padrões não conformes são chamados de anomalias, *outliers* […] A importância da detecção de anomalias se dá pelo fato que anomalias em dados se traduzem como alarmes significantes em uma grande variedade de aplicações...” (Chandola et al., 2009, p. 1,Tradução dos autores).

Chandola et al. (2009) define essas anomalias como padrões que fogem da normalidade de um conjunto e que diferentes fatores podem tornar essa detecção desafiadora. Um desses fatores é a disponibilidade de dados rotulados para treino/validação dos modelos utilizados por técnicas de detecção de anomalias.

Figura 14 - Imagem que mostra pontos O (anomalias) fora das regiões N (Normalidade)



Fonte: Chandola et al. (2009).

O uso de *autoencoders* (Um tipo de rede neural artificial) e GANs se mostraram bastante promissores para realizar a detecção de anomalias. Como visto em Carrara et al. (2020).

Com base nos conceitos teóricos apresentados nesse capítulo, uma revisão da literatura será realizada com o intuito de identificar como esses métodos têm sido aplicados em estudos anteriores e suas

3 REVISÃO DE LITERATURA

Esse capítulo apresenta analisando estudos e soluções voltados à detecção de anomalias no trânsito.

A seção 3.1 realiza a investigação dos métodos computacionais utilizados e a sua eficácia na tarefa de detecção de objetos, buscando identificar a aplicação desses métodos em trabalhos. Examina o uso de redes generativas para a ampliação de bases de dados, além da discussão dos resultados obtidos. A seção 3.2 concentra a análise em WGANs aplicadas em sistemas de detecção de anomalias. Por fim, a seção 3.3 apresenta as principais contribuições da análise bibliográfica.

3.1 Técnicas computacionais relevantes na detecção de anomalias

Para Yadav, et al. (2018), um dos mais importantes objetivos da vigilância do trânsito é a detecção de anomalias. Muitos dos métodos aplicados em trânsito se baseiam em aprendizado de máquina supervisionado, no entanto, os métodos supervisionados necessitam de um massivo número de dados com anormalidades para garantir um resultado com boa performance em termos da precisão dos resultados. Nesse contexto, o artigo sugere que anomalias de trânsito precisam ser detectadas com modelos de aprendizado baseados em aprendizado de máquina não supervisionados, como a visão computacional.

Segundo Szeliski (2021), desde os anos 1970, a visão computacional tem sido uma área de pesquisa desafiadora, inicialmente vista como parte da inteligência artificial para imitar a percepção visual humana. A visão computacional tem aplicações amplas, como fotografia digital, efeitos visuais, imagem médica, busca de imagens e o reconhecimento de objetos. A evolução em hardwares e algoritmos de aprendizado de máquina, como as redes neurais convolucionais, teria sido uma grande evolução dentro do campo de *computer vision*.

Características importantes das CNNs incluem compartilhamento de pesos em kernels, uso de *padding*, *stride* e convoluções dilatadas (ou "*atrous*") para captar padrões em escalas variadas. Outras inovações incluem convoluções parciais e "*gated convolutions*", que ajustam pesos dinamicamente para melhorar a confiança em cada pixel.

No entanto, nas décadas mais recentes, modelos baseados em características, ou as também chamadas features, têm se destacado na detecção de objetos. Mori et al. (2004) reitera que essa abordagem engloba técnicas que utilizam pontos de interesse e patches para reconhecimento de objetos, cenas, panoramas e locais, além de explorar estratégias como reconhecimento baseado em contornos e segmentação de regiões. Diversos trabalhos demonstram a eficácia dessas técnicas, que combinam aprendizado de máquina com extração de características locais e estruturais, consolidando-se como uma tendência predominante na pesquisa de visão computacional atual.

Tendo em vista o expressivo avanço proporcionado pela evolução e pela aplicação de técnicas de aprendizado de máquina no campo da visão computacional, destaca-se a necessidade de investigar os métodos computacionais mais amplamente utilizados e eficazes na tarefa de detecção de objetos. Essa análise se torna particularmente relevante diante da crescente demanda por soluções inovadoras que alavanquem a precisão e a eficiência em aplicações práticas, incluindo a análise de imagens em tempo real, a automação de processos industriais e o monitoramento de sistemas de trânsito. Nesse contexto, compreender as abordagens predominantes e seus impactos configura-se como um passo essencial para a consolidação do estado da arte nessa área de pesquisa.

Yadav, et al. (2018) abordam em seu trabalho, Detecção de Anomalias no cenário de vigilância de trânsito (*Detection of Anomalies in Traffic Scene Surveillance*), a classificação e aplicação de diversos métodos computacionais voltados para a detecção de anomalias em cenas de tráfego, apresentando uma taxonomia abrangente dessas técnicas. Entre os métodos analisados, destacam-se abordagens de visão computacional baseadas em fluxo óptico, modelos probabilísticos, sistemas neuro-fuzzy e análise de trajetórias, cada uma com suas características, desafios e aplicações específicas.

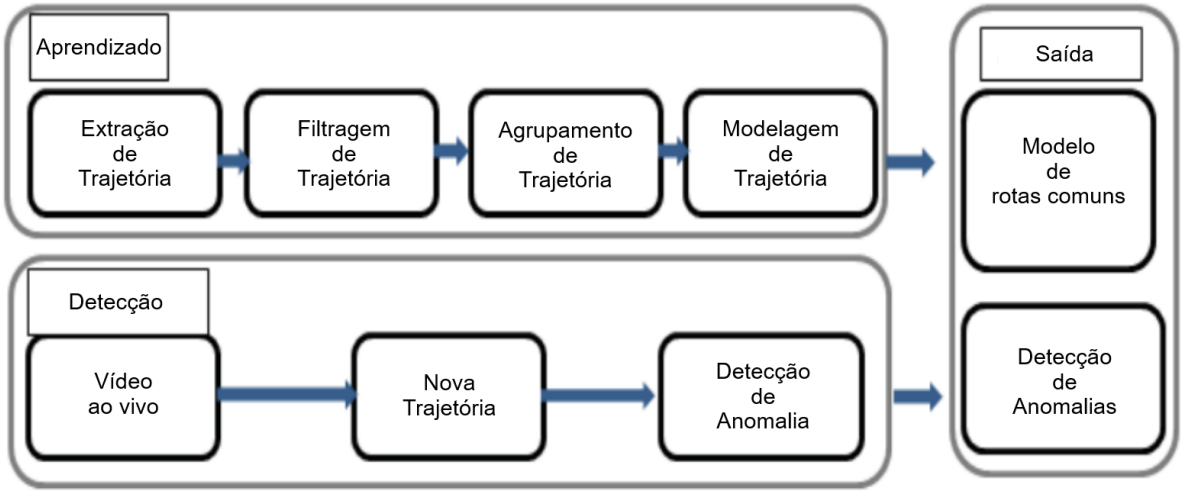
Os autores propõem um modelo baseado na análise de trajetórias para detectar anomalias em vídeos de tráfego. O processo inicia com a captura de vídeos que são convertidos em quadros estáticos. A partir de cada quadro, características dos objetos em movimento são extraídas, como posição, velocidade e direção. Em seguida, um ponto centróide é definido para cada objeto, permitindo analisar a trajetória de cada elemento em relação a outros pontos agrupados. Alterações

significativas na posição do centróide em relação a outros objetos podem ser classificadas como anomalias, representando possíveis violações de tráfego.

A análise de trajetória do projeto de Yadav, et al. (2018), representada na Figura 15 abaixo, foi dividida em três partes principais:

- **Fase de Aprendizado:** Nesta etapa, as trajetórias extraídas são filtradas e agrupadas com base em suas características utilizando o algoritmo K-Means, formando clusters que representam padrões normais de tráfego.
- **Fase de Detecção:** Novos vídeos são processados em tempo real, e as trajetórias dos objetos detectados são comparadas com os padrões aprendidos. Alterações significativas são identificadas como anomalias.
- **Fase de Saída:** Os resultados da análise são exibidos, destacando as anomalias detectadas e categorizando-as com base nos padrões identificados.

Figura 15 - Sistema de análise de trajetória



Fonte: Adaptado de Yadav et al. (2018).

Para finalizar, os autores empregaram também a *Hierarchical Temporal Memory* (HTM), uma abordagem inspirada no funcionamento do neocórtex humano. O HTM é dividido em subcomponentes de previsão, classificação e detecção de anomalias. Ele calcula a pontuação de anomalia com base em dados históricos, utilizando escores de anomalias brutas e sua probabilidade. A direção do veículo é destacada como uma característica determinante, formando clusters que facilitam a identificação de outliers em dados de tráfego.

O estudo de Yadav et al (2018) enfatiza que, embora existam diversas técnicas para detecção de anomalias no tráfego, nenhuma delas é completamente

eficiente ou livre de falhas, especialmente em cenários com grande variabilidade e complexidade. Os resultados mostraram que o método de *clustering*, ou agrupamento, detecta anomalias com maior precisão. Embora técnicas como regressão linear e método Z-score sejam boas para análise inicial de dados, elas não apresentam impacto significativo em dados de alta dimensionalidade. Assim, o algoritmo K-Means de *clustering* proposto é considerado o mais eficiente em comparação com outras técnicas de detecção de anomalias.

Para uma análise mais aprofundada das técnicas computacionais relevantes na detecção de objetos, foi realizada uma pesquisa entre os competidores da competição AI CITY 5ª edição de 2021. Esta competição, voltada para a inovação no campo de inteligência artificial aplicada à vigilância de tráfego, reuniu diversas abordagens e soluções tecnológicas, que podem fornecer insights valiosos para aprimorar as metodologias de detecção de anomalias no trânsito. A seguir, será detalhada a competição, destacando como foi realizada e as principais contribuições dos participantes para o avanço dessas técnicas.

O artigo de NAPHADE et al., (2021) aborda a detecção de anomalias no tráfego em vídeos capturados em interseções e rodovias em Iowa, EUA. As equipes participantes da competição tinham como objetivo identificar anomalias como acidentes, veículos parados e outras irregularidades no tráfego. A competição foi estruturada em duas fases principais: treinamento e teste, onde o conjunto de treinamento consistia em 100 vídeos, incluindo 18 vídeos com anomalias, enquanto o conjunto de teste consistia em 150 vídeos. Cada vídeo tinha uma resolução de 800×410 pixels e uma duração aproximada de 15 minutos.

A principal tarefa dos participantes do desafio 4 (*Track 4*) foi identificar anomalias de tráfego, como acidentes de veículos individuais e múltiplos, e veículos parados. O tráfego congestionado não era considerado uma anomalia. O vencedor da competição seria determinado pela maior precisão média e pela precisão do tempo de início da anomalia na previsão dos eventos detectados. O desempenho das equipes foi avaliado com base em dois critérios principais: *F1-score*, que mede a precisão geral da detecção, e erro de tempo de detecção, que é calculado através do erro quadrático médio normalizado das previsões de tempo de acidente.

O artigo detalha as metodologias das equipes que se destacaram na competição, principalmente a Baidu-SIAT, que obteve o melhor desempenho com um score de 0,9355. A equipe vencedora usou uma abordagem baseada em modelagem

de fundo, detecção de veículos, construção de máscara de estrada para remover veículos estacionados e rastreamento de veículos anômalos. A principal inovação foi a utilização de padrões espaço temporais e padrões de movimento para determinar com precisão o tempo de início das anomalias. Além disso, um módulo de pós-processamento foi aplicado para refinar ainda mais o tempo de início da anomalia no tráfego, garantindo maior precisão na detecção.

Segundo NAPHADE et al., (2021), a equipe ByteDance, que ficou em segundo lugar, utilizou um método de rastreamento em nível de caixa para identificar tubos espaço temporais anômalos, uma técnica que também foi usada para prever com precisão os períodos de anomalias. A WHU, equipe em terceiro lugar, combinou rastreamento em nível de caixa com rastreamento em nível de pixel para identificar anomalias, além de usar um módulo bidirecional de rastreamento de dupla modalidade, que também contribuiu para o refinamento da detecção dos períodos anômalos.

Essas metodologias destacam o uso crescente de técnicas de rastreamento dinâmico e análise espaço-temporal para detecção de anomalias no tráfego, mostrando como as abordagens atuais de aprendizado de máquina e análise de vídeo têm se tornado eficazes para resolver problemas complexos de detecção. A competição demonstrou que, embora as tecnologias atuais já sejam eficazes, ainda há espaço para aprimoramentos no refinamento do tempo de detecção e na precisão da identificação de anomalias.

A competição mostrou que a detecção de anomalias no tráfego é um problema complexo que pode ser abordado com técnicas avançadas de modelagem de fundo e rastreamento de veículos (Naphade et al., 2021). A metodologia baseada em padrões espaço temporais e o uso de rastreamento em níveis de caixa e pixel têm se mostrado eficazes para identificar com precisão as anomalias, como acidentes e veículos parados. O artigo defende que a disputa foi uma demonstração do potencial das tecnologias atuais para resolver problemas de detecção de anomalias no tráfego, mas também revelou que existem desafios a serem superados, especialmente na precisão do tempo de início das anomalias.

A equipe Baidu-SIAT, vencedora do 5º AI City Challenge na track 4 de 2021, apresentou o artigo Good Practices and A Strong Baseline for Traffic Anomaly Detection (Zhao et al., 2021). Os autores determinam que o objetivo de um sistema prático de detecção de anomalias é sinalizar em tempo hábil uma atividade que se

desvia dos padrões normais e identificar a janela de tempo em que a anomalia ocorre. Dessa forma, a detecção de anomalias seria considerada como uma compreensão de um vídeo de forma computacional em baixo nível, filtrando anomalias dos padrões normais.

As redes neurais convolucionais, segundo ZHAO et al., (2021), possuem ótima capacidade de modelagem e seriam capazes de aprender representações discriminativas a partir de dados visuais em grandes conjuntos de dados supervisionados. Para alcançar este marco, o treinamento destes modelos de aprendizado demanda um grande número de dados, tendo em vista que anomalias raramente ocorrem quando comparadas com as atividades normais do sistema analisado, contudo o conjunto para treinamento da *track* 4 era composto por apenas 100 vídeos.

Para mitigar essa limitação, a equipe investigou uma séria de boas práticas para localização temporal e detecção de anomalias, como estabilização de vídeo, detecção de colisões veiculares, refinamento de bordas temporais e etc.

Ao investigar metodologias para detecção de anomalias, a equipe as dividiu em duas abordagens: a metodologia tradicional e a baseada em aprendizado profundo (*deep learning*). Com o grande desenvolvimento da visão computacional alavancada pelo *deep learning*, redes baseadas em *autoencoders* e funções de perda bem projetadas se tornam pontos chave na tarefa de previsão de anomalias.

A detecção de anomalias no trânsito, trabalhada pela equipe Baidu-SIAT, foi determinada como uma forma mais refinada da detecção de anomalias comum, a qual incluiria diversos tipos de violações de regulamento de trânsito, como estacionamento ilegal, direção perigosa e excesso de velocidade, por exemplo. O grupo, ao estudar projetos passados do NVIDIA AI CITY Challenges, notou como métodos de aprendizado de máquina não supervisionados eram comuns dentro da detecção de anomalias, no passado foram utilizados métodos baseados em:

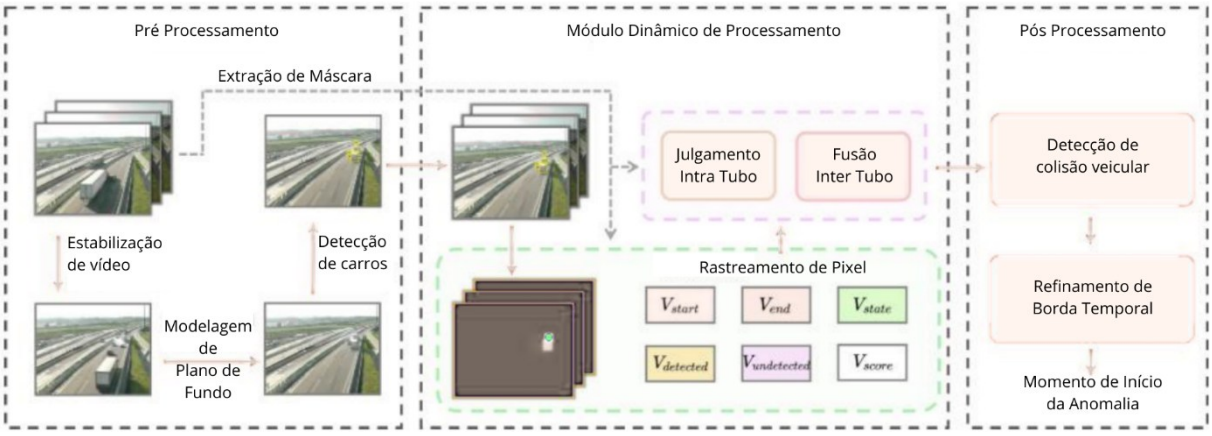
1. Método baseado na remoção de veículos em movimento do plano de fundo utilizando MOG2, pra performar uma detecção e classificação multi-escala de anomalias;
2. Método de aprendizado não supervisionado incluindo 3 fases: uma fase de extração de plano de fundo, uma de detecção de anomalias, e uma de confirmação das anomalias;

3. Método baseado em matriz de informações espaço-temporais, transformando a análise de trajetória em faixa em análise de posição espaço-temporal;
4. Método baseado em subtração de plano de fundo usando GMM e o detector YOLO para selecionar candidatos a anomalia, para detectar anomalias por aprendizado de transferência sem usar dados de treinamento.
5. Método baseado em rastreamento multi-granular, combinando um ramo de nível de caixa e um ramo de nível de pixel para analisar os veículos candidatos anômalos em diferentes níveis de granularidade.

Sendo que, o método número 5 teria ficado em primeiro lugar no AI CITY Challenge de 2020.

Após os estudos feitos, a equipe Baidu-SIAT desenvolveu seu primeiro sistema baseado nos melhores resultados encontrados, demonstrado na Figura 16, é baseado em 3 fases: O pré-processamento, com o objetivo de encontrar os candidatos a anomalias com estabilização de vídeo, modelagem de plano de fundo e detecção veicular; O módulo de rastreamento dinâmico, utilizando padrões de movimento e o status espaço-temporal dos veículos, bem como o momento de início das anomalias; E o pós processamento, refinando as bordas temporais das anomalias.

Figura 16 - *Framework* da equipe Baidu-SIAT



Fonte: Adaptado de Zhao et al., (2021).

O pré-processamento foi feito utilizando estabilização de vídeo, a partir de técnicas de estabilização digital de vídeo (DVS), corrigindo oscilações em câmera. Em seguida, a modelagem de plano de fundo, tenta distinguir o primeiro plano do plano de fundo, modelando dinamicamente os *backgrounds*, baseada em mistura de gaussianas (MOG) e análise de ablação, para assim a modelagem de plano de fundo

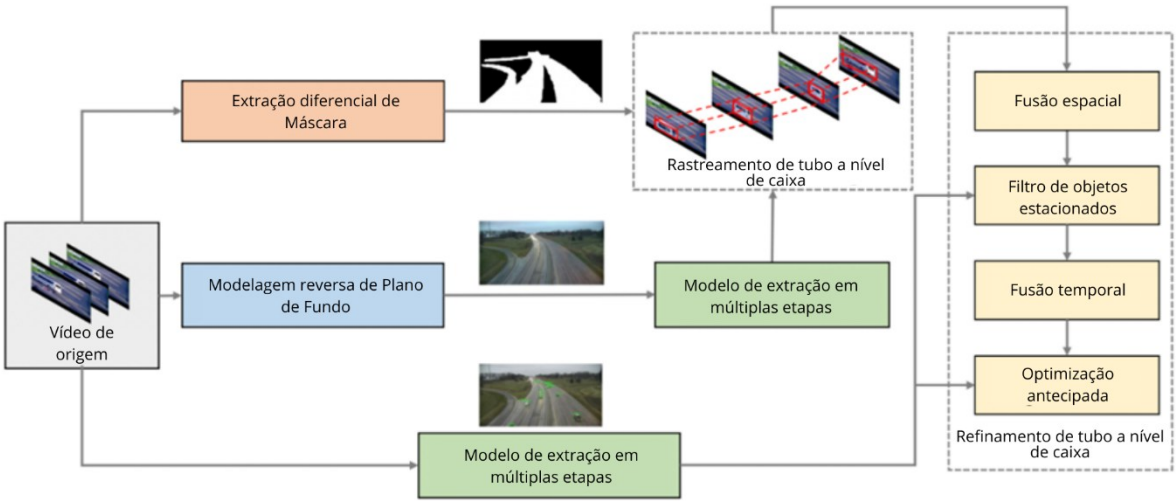
inversa (*backward*) destaque veículos parados enquanto a direta (*forward*) auxilia a determinar com mais precisão o tempo de início de uma anomalia. A detecção de veículos foi baseada em dois métodos de detecção de dois estágios, o *Faster R-CNN* com SENet-152 e o *Cascade R-CNN* com CBResnet-200, implementando ao final uma *Feature Pyramid Network* (Rede de características piramidal) para a construção em alto nível de um mapa de características. Por fim, a geração de máscaras é baseada na identificação de anomalias de tráfego, que geralmente ocorrem em veículos que transitam na via principal, sendo necessário filtrar veículos estáticos em vias secundárias e estacionamentos, utilizando o algoritmo de rastreamento multi-objeto DeepSORT para obter as trajetórias dos veículos das caixas delimitadoras (*bounding box*) dos veículos, que são então agrupadas em partes primárias e secundárias com base no ângulo da direção de movimento.

O módulo dinâmico de rastreamento, inicia-se com o rastreio de pixels (*Pixel Tracking*), utilizando o algoritmo IoU, comparando o objeto atual com o do próximo quadro de vídeo, fazendo uma comparação do resultado do IoU com um valor de limite determinado, para desenvolver uma em correlação espaço-temporal. Os candidatos da fase de rastreio de pixels, são então utilizados na etapa de *intra tube judgement* (julgamento intra-tubo), removendo as partes que não pertencem ao atual veículo anômalo, finalizando com apenas os tubos jugados como verdadeiramente anômalos. Por fim, na etapa de *inter tube fusion* (fusão inter-tubo), para evitar dados duplicados, como por exemplos o mesmo veículo em 2 tubos diferentes, é realizado o passo de fusão de tubos, com base na similaridade.

A fase de pós processamento, inicia-se com a detecção de colisão veicular, sendo batidas a anomalia mais comum na base de treinamento, sendo necessário estimar o tempo em que o veículo chega a uma parada completa e então retroceder no eixo temporal para analisar os eventos anteriores, observando também mudanças no primeiro plano e diferenças em imagem no plano de fundo, foi possível determinar se foi uma colisão e também, com precisão, o momento da colisão. Por fim, o refinamento de borda temporal é baseado na modelagem de plano de fundo direta (*forward*), podendo causar atrasos na previsão temporal devido à exibição de anomalias como pós-imagens, e melhorando a precisão na localização temporal, utilizando a modelagem de plano de fundo inversa (*reverse*) combinada com a similaridade de aparência, refinando assim os tempos de início e término da anomalia.

A equipe ranqueada em segundo lugar, ByteDance, propôs, em seu trabalho desenvolvido por Wu et al. (2021), um *framework* de rastreamento e refinamento baseado em caixas, que utilizou máscaras diferenciais, Modelos de Mistura Gaussiana para eliminar distúrbios dinâmicos e detecção em múltiplas etapas para identificar veículos anômalos, como aqueles parados por períodos excessivos. O método vinculou resultados de detecção para construir trajetórias de anomalias e aplicou refinamentos espaciais e temporais para melhorar a precisão das previsões. A Figura 17 abaixo, demonstra o *framework* proposto pela equipe, iniciando com a modelagem de plano de fundo inversa e a extração das máscaras diferenciais de anomalias, prosseguindo para detecção em múltiplas etapas, mas a etapa de rastreio de tubo a nível de caixa foi seu diferencial. Com isso, o time alcançou um *F1-score* de 0,9318 e ficou em segundo lugar na competição.

Figura 17 - *Framework* da equipe ByteDance



Fonte: Adaptado de Wu et al. (2021).

A partir do vídeo original, os autores iniciam a extração da máscara diferencial, utilizando diferenças notáveis entre dois quadros subsequentes, para assim evitar a interferência de objetos estáticos, como veículos estacionados, nos objetos em movimento significativos, considerados anômalos. A modelagem de plano de fundo, realizada no sentido inverso para minimizar erro temporal, similar a utilizada pela equipe Baidu-SIAT (2021), foi o modelo MOG2, pela capacidade de continuidade do fundo e pelo tempo de processamento. O time optou por uma detecção de veículos realizada com o uso de Cascade R-CNN, uma abordagem de detecção em múltiplos

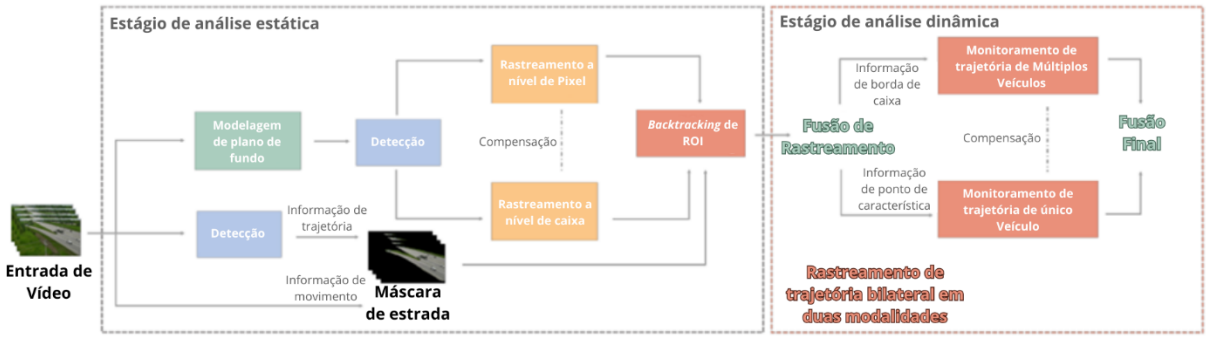
estágios, que emprega a arquitetura ResNeXt com FPN (*Feature Pyramid Networks*) para melhorar a performance na detecção de objetos pequenos.

Foi também idealizado pelos autores, um algoritmo de refinamento baseado em rastreio de tubos a nível de caixa (*box*) para analisar os candidatos a veículos anômalos. Inicialmente, as caixas de detecção são geradas através do método de detecção de múltiplos estágios e são filtrados com base na interseção entre as caixas e na confiança da detecção. Quando o valor calculado para a intersecção de duas caixas consecutivas, utilizando um algoritmo de IoU, essas duas caixas são conectadas para formar um tubo, representando assim a trajetória do veículo ao longo do tempo. A busca por conexões entre as caixas é realizada estendendo o rastreamento para frente e para trás, até que não haja mais interseções possíveis. Esse processo continua até todos as caixas detectadas sejam agrupados em tubos específicos.

Por fim, o refinamento dos dados, proposto por WU et al. (2021), é feito também a nível de caixa. Envolvendo quatro mecanismos principais para melhorar a detecção e reduzir falsos positivos, a fusão espacial combina tubos relacionados ao mesmo veículo, enquanto o filtro de objetos estáticos elimina falsos positivos, como placas de sinalização ou carros estacionados, utilizando detecções dos quadros originais. Esta fusão temporal combina os tubos que pertencem a um mesmo evento anômalo contínuo detectado, e a otimização retroalimenta o algoritmo com informações, ajustando os tubos de acordo com as detecções nos quadros originais, atualizando o tempo de início do evento anômalo.

Diferentemente das equipes Baidu-SIAT, que obteve o primeiro lugar, e ByteDance, que obteve o segundo lugar, a equipe WHU, em seu artigo por Chen et al. (2021), apresentou um sistema com uma diferença principal no método de detecção de objetos e pela ausência de rastreio de tubos. O *framework* da equipe era baseado em duas etapas, como ilustrado na Figura 18 abaixo. A primeira etapa, a análise estática, fornece o tempo e a localização em que um veículo parou completamente, enquanto a segunda etapa, análise dinâmica, introduz o módulo bilateral de dupla modalidade para recuperar o instante de colisão para anomalias em movimento.

Figura 18 - *Framework* da equipe WHU



Fonte: Adaptado de Chen et al., (2021).

A fase de análise estática inicia-se logo após a inserção de dados (vídeos), a modelagem de fundo, utilizando o método MOG2 (Gaussian Mixture Model - GMM), é eficaz na detecção de mudanças de cena e oscilações da câmera. Além disso, técnicas de rastreamento de tempo de início das anomalias são introduzidas para lidar com esse atraso. Para a detecção veicular, foi implementado o YOLOv5, escolhido devido ao seu equilíbrio entre custo, eficiência e precisão. Ele implementa o *módulo Cross-Stage-Partial-Connections* (CSP), que facilita a detecção de objetos pequenos, algo relevante para o conjunto de dados utilizado. O treinamento do modelo envolveu a rotulagem manual de uma fração dos quadros para a classe de veículos, a utilização de ancoragem adaptativa de caixas, com base na distribuição estatística das caixas delimitadoras utilizando o algoritmo de agrupamento K-Means, superando outros modelos como o *Faster R-CNN*.

Assim como as duas equipes anteriores, foi construída uma máscara de estrada baseada em movimento e em trajetória para evitar a interferência de objetos estáticos, como veículos estacionados. Esta máscara foi refinada utilizando o DeepSORT, um algoritmo de rastreamento de múltiplos objetos, para distinguir áreas com ou sem resultados de rastreamento, identificando assim as regiões da estrada que não são anômalas. Por fim, o rastreamento de veículos foi dividido em pixel e caixa. O primeiro analisou a dinâmica de veículos parados, usando matrizes espaço-temporais para identificar regiões suspeitas. O segundo, no entanto, utilizando o algoritmo DeepSORT acompanhou objetos detectados, realizando o refinamento dos candidatos a veículos anômalos. No final, o retrocedimento de ROI (Região de Interesse), aperfeiçoou a detecção do tempo de início das anomalias, compensando o atraso na detecção de veículos estáticos causado pela modelagem de fundo.

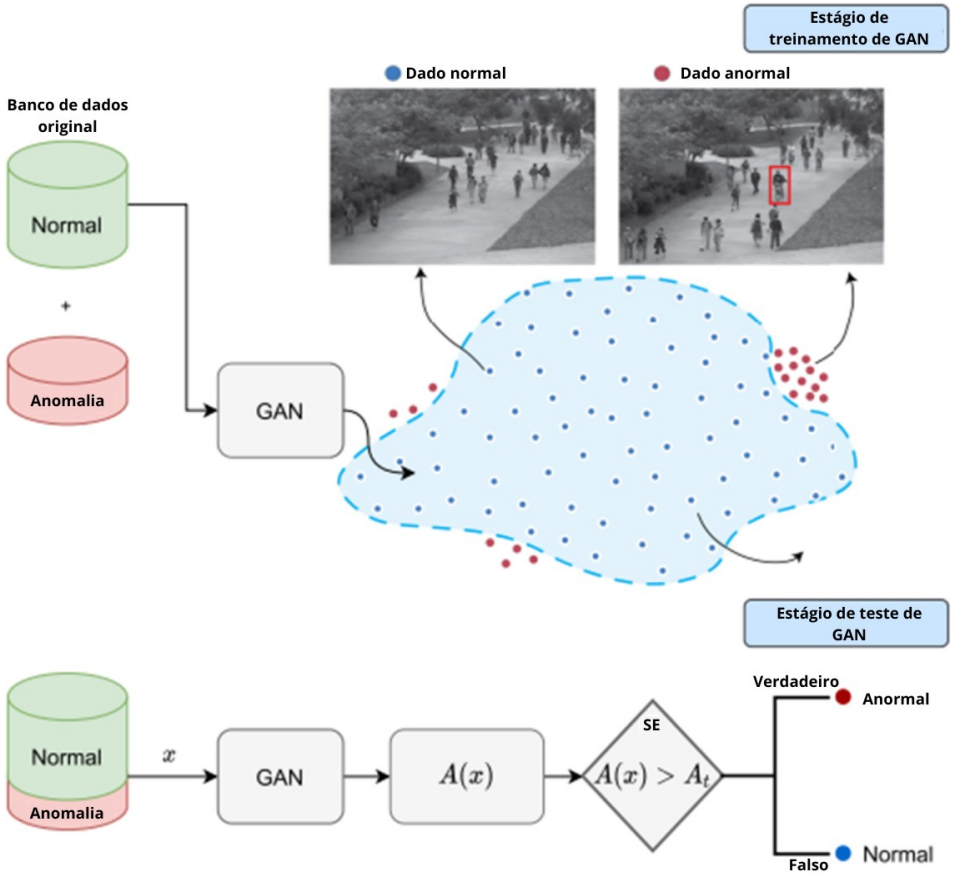
Apesar dos resultados obtidos pelos grupos anteriores, um problema em comum era notável em todas as abordagens, a dependência de grandes volumes de dados anotados e a dificuldade em generalizar para cenários complexos. Dessa forma, uma metodologia vantajosa para a detecção de anomalias em trânsito, seria uma metodologia capaz de trabalhar com a geração de dados sintéticos para o aumento de bases de treinamento, simulação de cenários de anomalias mais raras e rastreamento de condições adversas. Dentro deste cenário, pode se reconhecer a vantagem do uso de redes adversárias generativas, ou GAN (*Generative Adversarial Network).*

3.2 WGANs em sistemas de detecção de anomalias

As GANs, como modelo generativo vem sendo implementadas em pesquisas de detecção de anomalias, como por exemplo a de Sabuhi et al. (2021). Este trabalho apresenta uma revisão sistemática de 128 estudos, para encontrar as possíveis aplicações de redes generativas adversárias em detecção de anomalias.

Recuperando acerca das redes adversárias generativas (GAN), segundo Goodfellow et al. (2016), é uma metodologia em modelagem generativa, onde um uma rede generativa compete contra uma rede discriminativa. A rede generativa produz exemplos de dados, enquanto o discriminador avalia estas mesmas informações geradas, com uma saída da probabilidade da entrada ser um dado real ou artificial. O processo de aprendizado, representado na Figura 19, começa com o discriminador, que busca maximizar a sua saída de dados artificiais, enquanto o gerador busca minimizá-la. A convergência acontece quando os dados gerados e os da base de dados são indistinguíveis para o discriminador. Esta convergência é conhecida como equilíbrio de Nash (*Nash Equilibrium*).

Figura 19 - Estágios de treinamento e testes de GAN



Fonte: Adaptado de Sabuhi et al. (2021).

SABUHI et al. (2021) apresentam dois principais papeis das GANs na detecção e anomalias, sendo eles:

1. Incremento de dados (GAN-Assisted): As redes generativas adversárias geram dados sintéticos para aumentar o conjunto de dados, tendo se demonstrado útil em casos de escassez de exemplos de anomalias, equilibrando conjuntos de dados onde a classe de anomalias é minoritária.
2. Aprendizado de representação (GAN-Based): Os modelos de redes generativas adversárias exercitam a diferenciação de distribuição de dados normais e, durante a execução de testes, identificam as anomalias como desvios da distribuição, sem a necessidade de dados rotulados de anomalias. Esse papel é particularmente vantajoso em cenários onde os dados são difíceis de se obter.

Para os autores, as principais vantagens da utilização das GANs para detecção de anomalias, se encontra principalmente no incremento de dados. Ter uma metodologia útil com poucos exemplos de anomalias, aliado com o aprendizado semi-

supervisionado, onde não é necessária a rotulação de anomalias, faz das redes um método poderoso. Os estudos iniciais do trabalho apresentaram que o uso de GAN para o aumento de dados, reportou um aumento significativo na precisão em comparação com as técnicas mais tradicionais ou sem o aumento de dados. Em um teste de precisão para detecção de anomalias cardiovasculares, o uso de GAN representou um aumento de precisão de 81,93% para 84,19%. Enquanto que, o uso de técnicas tradicionais, conquistou uma precisão inferior, de 83,12%.

O problema mais crucial identificado pelos autores na utilização de GANs é o *mode collapse* (colapso de modelo). Esses colapsos são quando o gerador identifica uma forma de minimizar a sápida de dados, produzindo uma variedade limitada de dados, resultando em falha na diversidade de dados gerados pelo sistema. São um obstáculo para a detecção de anomalias, sendo que instabilidades no treinamento e abaixa variedade de dados são possíveis impeditivos para alcançar o *Nash Equilibrium*.

A vigilância de detecção de anomalias em vídeos é o segundo campo com maior domínio e aplicação desta técnica, segundo os autores, ficando atrás de medicina.

Foram identificados 21 diferentes tipos de GANs para a detecção de anomalias e identificada sua aplicação em dominância. A Tabela 1 abaixo, demonstra as mais utilizadas para o ramo de reconhecimento de imagens (*Image Recognition* - IR), supervisão de vídeos (*Surveillance* - SU) e detecção de trajetórias (*Trajectory detection* - TD).

Tabela 1 - Tipos de GANs mais utilizadas e os ramos em que são mais utilizadas

Tipo de GAN	IR	SU	TD
Deep Convolutional GANs	✓	✓	
Standard GANs		✓	✓
Conditional GANs	✓	✓	✓
Bi-Directional GANs		✓	
Wasserstein GANs	✓		
Wasserstein GANs with Gradient Penalty	✓		
Variational AutoEncoder GANs		✓	
Patch GANs		✓	

Fonte: Os autores (2025).

Os autores prosseguem com a explicação de que, em uma tentativa de minimizar o *mode colapse* e os desafios para a Standard GAN alcançar o *Nash equilibrium*, foi idealizado usar a *Earth Move Distance* (EMD), também conhecida como *Wasserstein-1 distance*. Diferentemente de outros modelos de GANs, que tentariam estabilizar modificando seu próprio custo de função, as WGANs atualizariam o gerador a partir da distância, a partir de gradientes mais úteis.

E, mesmo que as WGANs tenham apresentado uma melhora em lidar com os colapsos, apresentou dificuldades para convergir. Foi então proposto uma melhoria, a introdução do gradiente de penalidade (*Gradient penalty*) para o discriminador, resultando em melhores em convergências, velocidade de treinamento e qualidade de exemplos.

As GANs, conforme o estudo dos autores, necessita utilizar métodos de detecção de anomalias úteis para cada caso para apresentar os melhores resultados.

As técnicas supervisionadas de detecção de anomalias, ou seja, os modelos treinados com dados rotulados de comportamentos normais e anômalos, as GANs são utilizadas principalmente para o incremento de dados, equilibrando os conjuntos e em conjunto com modelos de classificação de dados. É o comum o uso das *Support Vector Machines* (SVMs), classificando dados como anômalos ou normais após o

aumento de dados, assim como técnicas baseadas em redes neurais em cenários mais complexos como imagens médicas.

As técnicas semi-supervisionadas, os modelos são treinados principalmente a partir de dados normais, e as anomalias são identificadas como desvios da distribuição aprendida, utilizando as GANs principalmente para realizar o cálculo de pontuação de anomalia (*anomaly score*). As pontuações que exercem um limite pré definido, são consideradas anômalas. A AnoGAN é a ténica baseada em GAN mais utilizada como uma base de comparação para novos métodos, em termos de aprendizado semi-supervisionado. Os estudos demonstram que muitos trabalhos utilizaram Mixture o Dynamic Texture como técnica de detecção de anomalias em cenas com multidões, no entanto, alguns estudos propuseram o uso de técnicas de *Long Short Tem Memory* (memórias de curto longo prazo).

Nas técnicas não supervisionadas, os modelos não utilizam dados rotulados, nem mesmo para a classe normal. As WGANs foram frequentemente utilizadas pelos modelos de detecção de anomalias em casos não supervisionados, segundo o estudo de Sabuhi et al. (2021), no entanto, baseia-se em assumir que os dados normais são muito mais frequentes que os anômalos. Caso esta suposição seja falsa, a detecção de anomalias apresentará diversos casos inválidos. Portanto, para casos supervisionados, pode-se adaptar modelos de detecção de anomalias em conjunto com os não supervisionados, treinando parte do banco de dados não rotulado.

Para as técnicas não supervisionadas, os WGAN e WGAN-GP se demonstraram em destaque entre as referências buscadas pelos autores.

Ainda segundo Sabuhi et al. (2021), apenas 27 dos 128 estudos utilizados avaliaram a qualidade dos dados gerados pelas implementações, utilizando 9 diferentes métricas de medição de performance. A maior parte dos estudos, analisaram os dados quantitativamente e qualitativamente, enquanto 9 deles realizaram uma inspeção visual para avaliar a qualidade das saídas. A métrica mais comum para avaliar a qualidade dos dados gerados foi SSIM (*Structural Similarity Index*). Enquanto que, para avaliar o desempenho na detecção de anomalias, a métrica mais utilizada é a AUROC (Area Under the Receiver Operating Characteristic Curve), presente em 53% dos estudos.

3.3 Principais contribuições da análise bibliográfica

A detecção de anomalias é uma área crítica em diversas aplicações, onde a identificação de dados anômalos pode prevenir acidentes e melhorar a segurança das vias urbanas. Após os estudos realizados, esta seção representa as principais contribuições adquiridas após a revisão de literatura.

Os principais métodos de detecção de anomalias encontrados nos trabalhos estudados, se baseavam em aprendizado supervisionado e não supervisionado. O primeiro, necessita de dados rotulados, incluindo anomalias, para treinar seus modelos. O segundo, no entanto, identifica padrões em cenários não rotulados.

Dentro dos aprendizados não supervisionados, foram destacadas as equipes como Baidu-SIAT e ByteDance, do AI City Challenge de 2021, que alcançara, o primeiro e segundo lugar, respectivamente. Ambas utilizaram técnicas de modelagem de fundo (como MOG2) e rastreamento dinâmico para identificar veículos anômalos. A combinação de rastreamento em nível de caixa e pixel permitiu a detecção precisa de anomalias, como acidentes e veículos parados.

Porém, ambas as equipes notaram dificuldades por conta do baixo volume de dados disponibilizada pela produção do desafio em 2021. A fim de mitigar o problema da falta de dados em detecção de anomalias, as GANs foram analisadas. As redes adversárias generativas são utilizadas tanto para incremento de dados (gerando dados sintéticos para treinamento) quanto para aprendizado de representação (identificando anomalias como desvios da distribuição normal). Variantes como Wasserstein GANs (WGANs) e WGANs com Penalidade de Gradiente (WGAN-GP) são destacadas por sua capacidade de lidar com o *mode collapse* e melhorar a convergência do modelo.

Por fim, as principais métricas de avaliação de desempenho encontradas foram:

- *F1-Score*: Uma das métricas mais utilizadas para avaliar a precisão geral da detecção de anomalias. Equilibra a precisão e o recall, sendo especialmente útil em cenários onde há um desbalanceamento entre classes normais e anômalas.
- Erro de Tempo de Detecção: Medido pelo erro quadrático médio normalizado, essa métrica avalia a precisão do tempo de início das anomalias detectadas.

- AUROC (*Area Under the Receiver Operating Characteristic Curve*): Utilizada para avaliar a capacidade do modelo em distinguir entre classes normais e anômalas. Uma AUROC alta indica um bom desempenho na detecção de anomalias.
- SSIM (*Structural Similarity Index*) e PSNR (*Peak Signal-to-Noise Ratio*): Métricas comuns para avaliar a qualidade dos dados gerados por modelos generativos, como GANs. O SSIM mede a similaridade estrutural entre imagens, enquanto o PSNR avalia a qualidade da reconstrução de imagens.

A detecção de anomalias no tráfego se mostrou um desafio complexo, com altos investimentos, mas que requer abordagens avançadas, como a combinação de técnicas de aprendizado de máquina não supervisionado, altos volumes de dados, análise de trajetórias e a modelagem de fundo. As metodologias baseadas em redes neurais convolucionais e GANs se mostraram eficazes, especialmente quando combinados com técnicas de rastreamento dinâmico e refinamento temporal. Para avaliar o desempenho desses métodos, métricas como *F1-Score*, AUROC e erro de tempo de detecção foram utilizadas por pesquisadores e pelas equipes participantes do *AI City Challenge*, avaliando a precisão e confiabilidade dos modelos implementados.

Diante dos *frameworks* adotadas pelas principais equipes do *AI City Challenge* e do potencial da WGAN-GP na geração de dados sintéticos para aprimorar a detecção de anomalias, percebe-se uma oportunidade promissora para o desenvolvimento de um sistema híbrido que una essas abordagens. A combinação dessas técnicas não apenas aproveita os pontos fortes dos métodos mais bem-sucedidos da competição, como também busca superar limitações por meio da ampliação e diversificação dos dados de treinamento. Com base nesse embasamento teórico, as próximas seções apresentarão a proposta deste trabalho, detalhando a concepção, implementação e validação desse sistema no mesmo cenário utilizado pelas equipes mais bem colocadas do desafio, a fim de avaliar sua eficácia e viabilidade prática.

4 SISTEMA DE DETECÇÃO DE ANOMALIAS

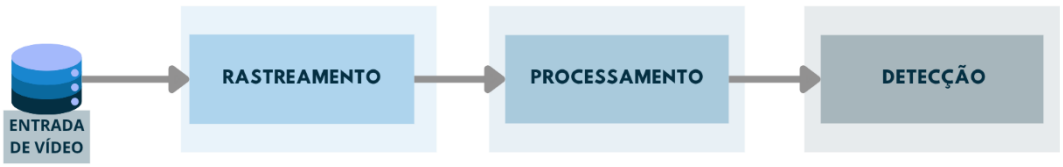
O objetivo deste capítulo é descrever a arquitetura do sistema de detecção de anomalias desenvolvido nesta pesquisa. Inicia explicitando a concepção do modelo de detecção de anomalias e os fundamentos por trás das técnicas utilizadas. A arquitetura geral é composta por quatro módulos. O sistema detalha o rastreamento de veículos, o processamento de trajetórias, a detecção de anomalias baseada em WGAN-GP e o pós-processamento e avaliação de resultados. O sistema foi projetado com base nos dados da *track* 4 do *AI City Challenge* 2021.

4.1 Estrutura geral do sistema

Esta seção descreve a arquitetura do sistema de detecção de anomalias no trânsito urbano desenvolvido nesta pesquisa, representado na Figura 20 abaixo, detalhando seus componentes, fluxos de dados e módulos funcionais, e suas diferentes adaptações. O sistema é estruturado de forma modular e sequencial, visando garantir a consistência do processamento e a escalabilidade do modelo de aprendizado.

O sistema foi projetado com base na base de dados da *track* 4 - *Traffic Anomaly Detection* do *AI City Challenge* 2021, com o objetivo de permitir a comparação direta de desempenho entre o *framework* proposto e os modelos das equipes participantes. A arquitetura geral é composta por quatro módulos principais: (1) rastreamento de veículos, (2) processamento de trajetórias, (3) detecção de anomalias baseada em WGAN-GP e (4) pós-processamento e avaliação de resultados.

Figura 20 - *Framework* de detecção de anomalias desenvolvido neste projeto



Fonte: Os autores (2025).

Todo o código desenvolvido durante este trabalho, está disponível no GitHub, pelo seguinte link: https://github.com/AchillesMacarini/utfpr-traffic-anomaly-detection.

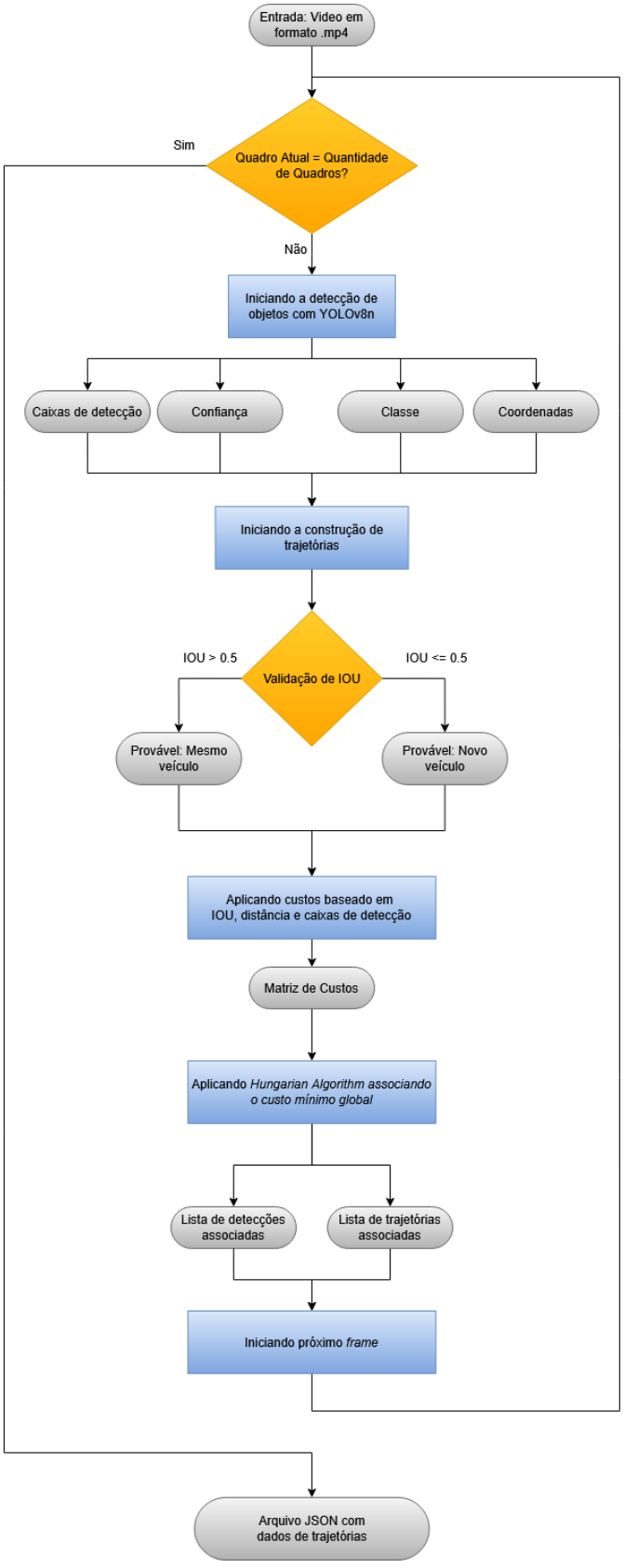
4.2 Rastreamento de veículos

O primeiro módulo é responsável pela detecção e rastreamento de múltiplos veículos em vídeos de tráfego urbano, representado pelo diagrama na Figura 21, utilizando a biblioteca PyTorch e o modelo YOLOv8n para detecção de objetos. A biblioteca OpenCV é fundamental para este processo, controlando a abertura, a leitura frame a frame, e a liberação dos recursos de vídeo.

A detecção de veículos é realizada sobre as classes *car* (carro), *motorcycle* (motocicleta), *bus* (ônibus) e *truck* (caminhão). Impõe-se um limite de confiança mínima de 0.3 e um *threshold* de Interseção sobre União (IoU) de 0.7 para a supressão não-máxima. A saída bruta, um objeto de resultados do Ultralytics (YOLO), contém todas as detecções, das quais são extraídas as caixas delimitadoras, os valores de confiança, as classes e coordenadas. A lista de detecções é submetida a uma filtragem que descarta detecções com confiança inferior a 0.4 ou classes que não representam veículos.

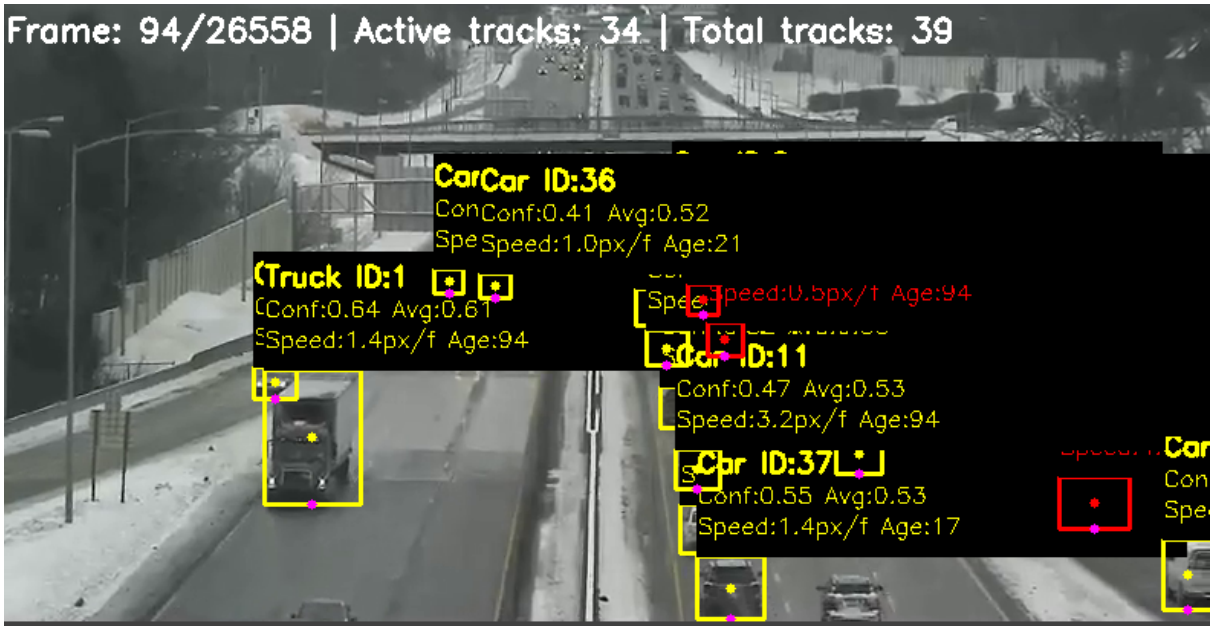
Para manter a identidade do veículo, o IoU é utilizado como métrica de sobreposição. Ao receber duas caixas delimitadoras, é calculada as coordenadas da interseção e verificada a sobreposição, retornando um valor entre 0 e 1 para a continuidade do objeto. Um IoU superior a 0.5 sugere a probabilidade de ser o mesmo veículo, enquanto um valor inferior ou igual a 0.5 sugere a probabilidade de ser um veículo novo. O processo de detecção e identificação de veículos, pode ser visto na Figura 22, aplicando o rastreamento de veículos ao vídeo 29.

Figura 21 - Diagrama do algoritmo de detecção de veículos



Fonte: Os autores (2025).

Figura 22 - Demonstração de vídeo 29 detectando veículos



Fonte: Os autores (2025).

O processo de associação utiliza uma matriz de custos associados, que representa a probabilidade de uma detecção atual pertencer a uma trajetória já existente. O custo total é uma combinação ponderada de três componentes, conforme a equação 4.2.1.

$$Custo = (Distância\ Euclidiana\ \times 0{,}5) + \big((1 - IoU) \times 100\big) + ((1 - Razão\ de\ Tamanho) \times 50) \tag{4.2.1}$$

O componente de falta de sobreposição possui o peso 100, conferindo-lhe influência na decisão de associação, pois a sobreposição é o indicador de continuidade. O componente de razão de tamanho possui o peso 50, penalizando detecções com tamanhos divergentes. O componente de distância euclidiana, medindo a distância em pixels entre os centros, possui o peso 0,5, conferindo influência reduzida para não dominar a decisão em cenários de movimento rápido.

Com a matriz de custos definida, o sistema aplica o algoritmo húngaro (*Hungarian Algorithm*), associando o custo externo global e encontrando a correspondência de menor custo total entre as detecções e os tracks ativos.

Então, ao iniciar a fase de extração de trajetórias, o algoritmo realiza a filtragem de tracks com comprimento inferior a 5 frames para validação da trajetória.

Ao final do processamento, os dados de trajetória são salvos em um arquivo JSON, completando a extração para o módulo subsequente.

4.3 Processamento de trajetórias

O segundo módulo é responsável pela transformação das trajetórias brutas em representações numéricas padronizadas adequadas para entrada na rede adversarial. O processamento é implementado utilizando as bibliotecas NumPy, SciPy e Scikit-learn, transformando dados das extrações de trajetórias, realizadas durante o rastreamento de veículos, em arquivos NumPy com dimensão fixa e gráficos de amostra para visualização.

A primeira etapa é a extração de trajetórias onde as informações de quadro, coordenadas e confiança são armazenadas e adicionadas à lista de trajetórias, indexada pelo do veículo. Após a extração, cada trajetória é ordenada sequencialmente pelo número do quadro.

A segunda etapa é a filtragem, aplicando um critério de remoção para trajetórias com comprimento inferior a cinco pontos. O processo de filtragem remove a trajetória se o número de pontos for menor que o comprimento mínimo.

A terceira etapa é o cálculo de *features*. As características são:

- Posição normalizada dos veículos: Cada coordenada ‘x’ é dividida pela largura do quadro, e a coordenada ’y’ é dividida pela altura do quadro.
- Velocidade: Para cada veículo em um instante, a velocidade é dada a partir de um intervalo temporal Δ pela diferença entre o quadro atual e o quadro anterior.
- Aceleração: Calculada pela diferença entre a velocidade v atual e a anterior, dividida por Delta t.
- Direção: Utiliza a função arco tangente de dois argumentos $arctan\,2(v_x, v_y)$.

A quarta etapa é a normalização global, começando pela coleta de estatísticas, percorre todas as trajetórias para extrair e adicionar às listas globais as velocidades e acelerações. Desta coleta, são calculados o valor máximo de velocidade e o valor máximo de aceleração. Utilizando esses valores para dimensionar as features, a velocidade normalizada é obtida pela divisão da velocidade pela velocidade máxima, e a aceleração normalizada é obtida pela divisão da aceleração pela aceleração máxima.

A quinta etapa é a interpolação para tamanho fixo, para padronizar todas as trajetórias para o comprimento fixo de 20 pontos. Para tal, são consideradas três decisões: (1) se o comprimento atual for igual ao comprimento alvo, a trajetória retorna sem alteração; (2) se o comprimento atual for inferior a dois, um arranjo de zeros com a dimensão alvo é criado, os pontos existentes são copiados, e a lista é retornado preenchido com zeros. (3) No caso geral, quando o comprimento atual é maior que dois, são criados índices originais e índices alvo (vinte pontos espaçados uniformemente).

A sexta e última etapa do módulo é a geração de arquivos, na qual a lista de trajetórias padronizadas são convertidas para um arquivo NumPy, com a forma (número de trajetórias, 20, 5), onde o Eixo 0 representa cada trajetória individual, o Eixo 1 representa os 20 pontos temporais, e o Eixo 2 contém as 5 features.

4.4 Detecção de anomalias com WGAN-GP

O terceiro módulo é responsável pela modelagem e identificação de padrões anômalos em trajetórias de veículos. A arquitetura segue uma evolução progressiva dos modelos de *Wasserstein Generative Adversarial Network with Gradient Penalty* (WGAN-GP). O processo de avaliação define o *threshold* de anomalia a partir de um percentil. Quando os rótulos de anomalia estão disponíveis, o sistema avalia o desempenho utilizando as métricas de precisão, *recall*, *F1-Score*, *accuracy*, AUC-ROC e AUC-PR.

Para este desenvolvimento, foram implementadas 3 diferentes abordagens, com variações em complexidade e estabilidade: (1) camadas lineares, (2) camadas LSTM, e (3) utilizando também camadas LSTM, mas com a aplicação de bibliotecas específicas para os cálculos.

4.4.1 Camadas Lineares com Pytorch

Esta primeira abordagem baseia-se na utilização de camadas lineares na rede geradora e uma arquitetura convolução unidimensional (Conv1d) na rede discriminadora, como demonstrado na Figura 23. O desenvolvimento descreve a estrutura dos componentes, os parâmetros de treinamento e o fluxo de detecção de anomalias, fornecendo subsídios para a reprodutibilidade do modelo. O pacote utilizado para o desenvolvimento é o PyTorch.

O processamento dos dados inicia com a gestão dos rótulos de um arquivo CSV que contém a identificação do vídeo, o tempo de início e o tempo de fim de cada anomalia. O sistema organiza estas informações em um dicionário, no qual cada identificador de vídeo está associado a uma lista de tuplas representando os intervalos anômalos.

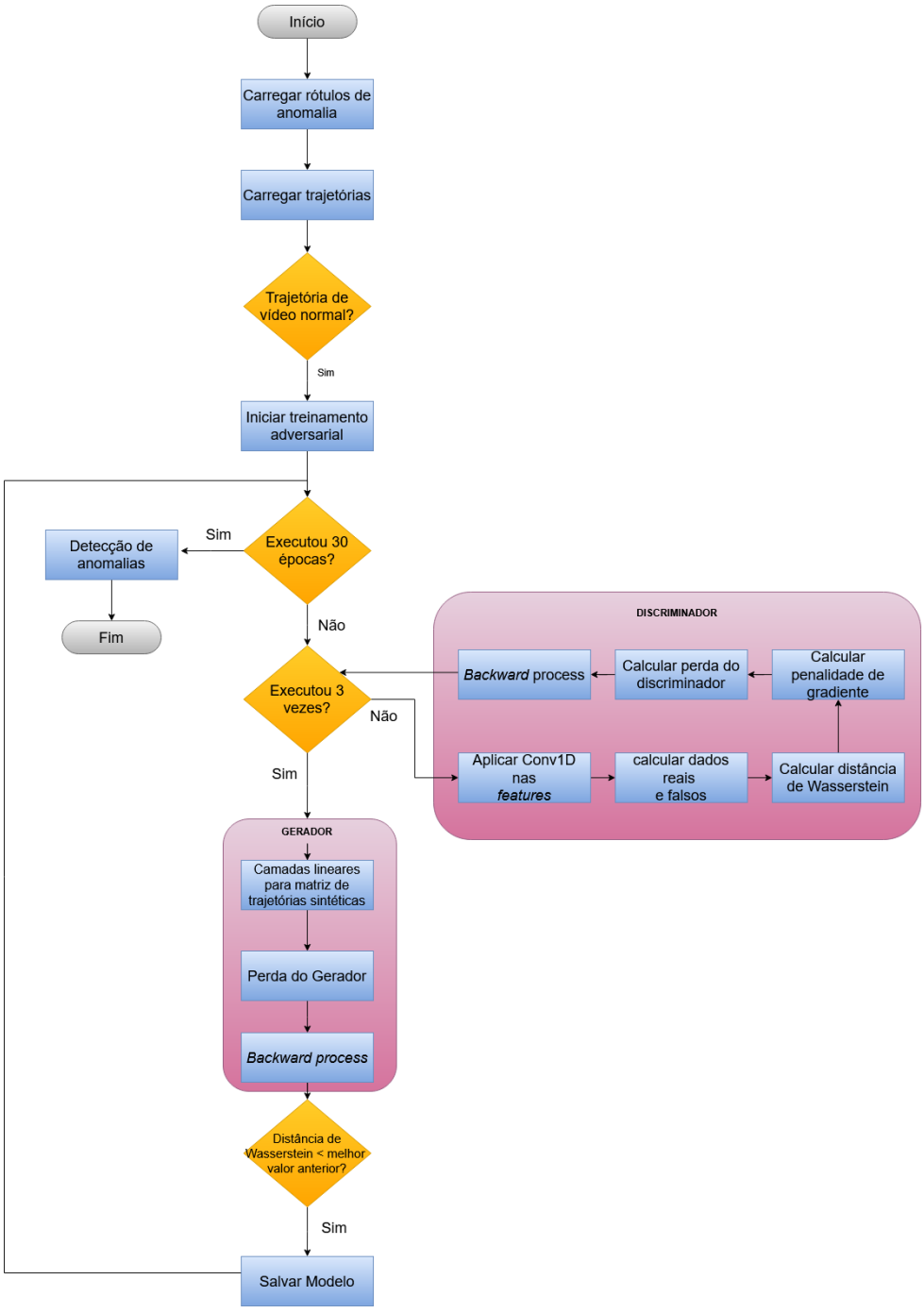
O sistema localiza os arquivos .npy contendo as trajetórias processadas. O carregamento impõe um limite de 500 trajetórias por vídeo para otimizar a memória. Para o treinamento do modelo, o conjunto de dados é configurado para utilizar apenas trajetórias oriundas de vídeos normais. A separação entre treinamento e validação utiliza uma proporção de 80% para o treinamento. O conjunto de validação inclui as trajetórias normais remanescentes e, de forma complementar, trajetórias anômalas.

Inicia-se a fase de treinamento. Este passo é configurado para trinta épocas, utilizando um tamanho de lote (*batch size*) de 128. Os otimizadores do tipo são aplicados, com taxas de aprendizado definidas em 0.0002 para o gerador e 0.0001 para o discriminador.

O algoritmo implementa a penalidade de gradiente, assegurando a estabilidade do discriminador. Este processo envolve a geração de um valor α aleatório entre 0 e 1, o cálculo de amostras interpoladas (mistura linear de dados reais e falsos) e a determinação da norma dos gradientes do discriminador sobre essas amostras interpoladas.

Para cada lote de dados, é realizada a convolução unidimensional utilizando Conv1d, recebendo trajetórias em sequência de 20 e uma dimensão de características de 5, transformada para 64, prosseguindo para 128 e mais uma vez para 256 canais. Essa nova matriz passa por um processo de *flattening*, com a saída escalar final.

Figura 23 – Fluxograma da implementação de WGAN com Pytorch e camadas lineares



Fonte: Os autores (2025).

O discriminador é atualizado por três iterações, onde a função de perda do discriminador é calculada pela distância de Wasserstein, a diferença entre a média dos *scores* reais e falsos, e é somada ao termo de penalidade de gradiente. Após o treino do discriminador, o gerador é treinado por uma iteração.

A estrutura do gerador se inicia com três camadas lineares consecutivas:

1. A camada 1 projeta a entrada de sessenta e quatro para cento e vinte e oito dimensões, realizando a normalização dos lotes.
2. A camada 2 realiza a transformação de cento e vinte e oito para duzentas e cinquenta e seis dimensões, realizando também a normalização dos lotes.
3. A camada 3 mapeia a saída para o produto do comprimento da sequência (as sequências tem um tamanho igual a 20) pela dimensão das *features*. A saída final desta camada utiliza é formatada para uma matriz tridimensional, representando as trajetórias sintéticas.

Esta função de perda do gerador é definida pelo negativo da média dos *scores* falsos, visando maximizar o *score* atribuído pelo discriminador às amostras geradas. Os gradientes de ambas as redes são aplicados com *clipping*. O salvamento do modelo de desempenho superior é condicionado à melhoria na distância de Wasserstein.

A detecção de anomalias é baseada exclusivamente na saída do discriminador. A pontuação de anomalia é calculada pela inversão do *score* do discriminador (negativo do *score* de saída), visto que *scores* discriminadores menores indicam trajetórias mais desviantes do padrão normal aprendido.

4.4.2 Camadas LSTM com Pytorch

Posteriormente, a segunda abordagem, representada na Figura 24, aprimora o desempenho por meio da introdução de modelagem temporal de camadas *Long Short-Term Memory* (LSTM). Esta versão corresponde à abordagem que incorpora aprimoramentos arquitetônicos e estratégias de dados em relação à implementação inicial, buscando o desempenho superior. O pacote utilizado para o desenvolvimento é o PyTorch.

O processamento de dados inicia pela leitura e organização dos rótulos de anomalias. O sistema carrega um arquivo CSV contendo o identificador do vídeo, o tempo de início e o tempo de fim de cada evento anômalo. Esses registros são convertidos em uma estrutura de dicionário que associa cada vídeo aos intervalos correspondentes de anomalia. Além disso, o sistema estima a duração total de cada vídeo e calcula uma pontuação de severidade para os eventos registrados.

A etapa seguinte contempla o carregamento das trajetórias previamente processadas. O sistema localiza e lê arquivos no formato .npy, aplicando um limite

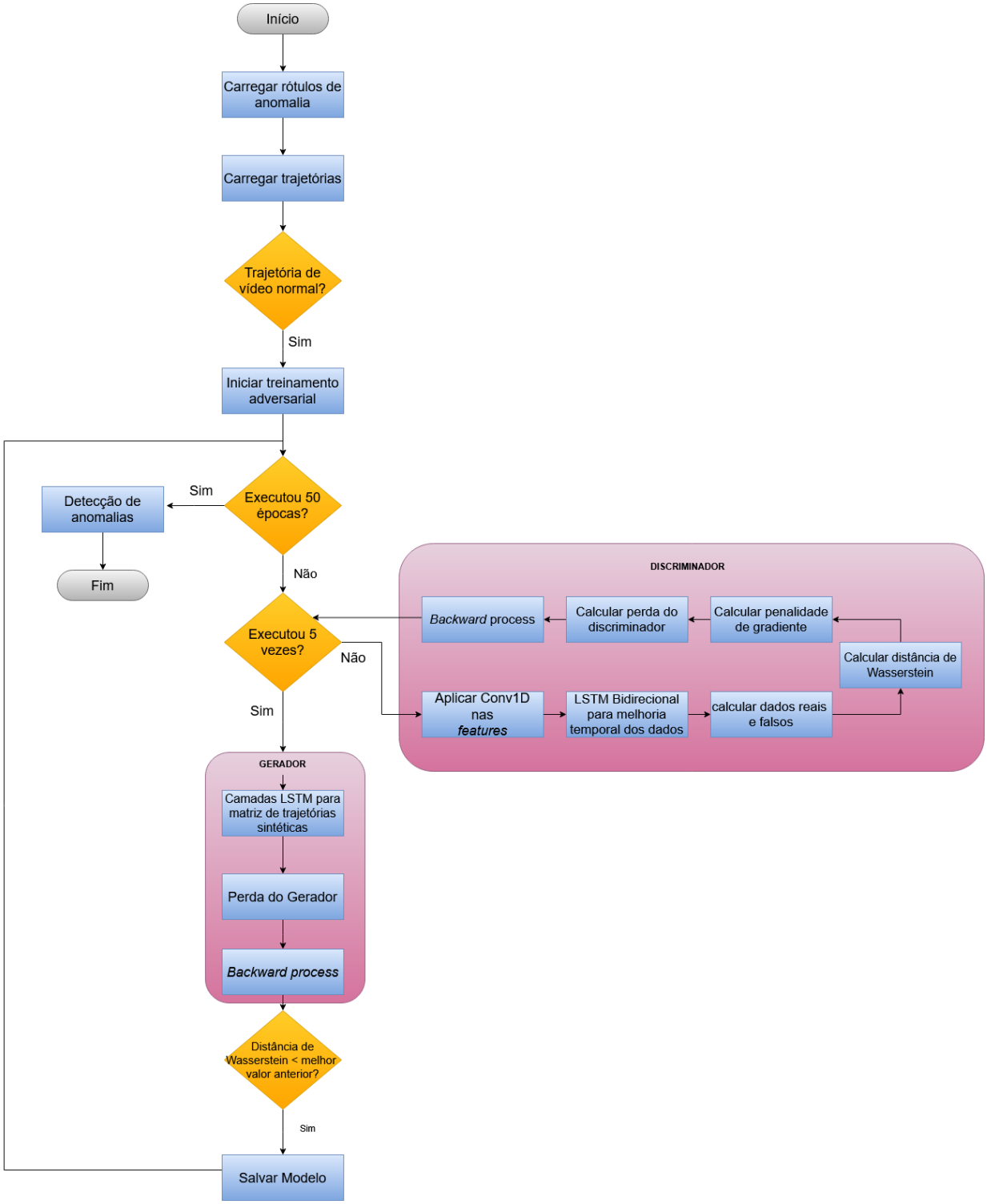
máximo de quinhentas trajetórias por vídeo para otimização de memória. Para o treinamento da rede adversarial, são selecionadas exclusivamente trajetórias provenientes de vídeos normais. A divisão entre conjuntos de treinamento e validação segue a proporção de 80% para o treinamento e 20% para testes. O conjunto de validação é balanceado segundo uma proporção específica de amostras anômalas (*balance ratio*), incluindo até duzentas trajetórias anômalas por vídeo e limitando-se a três vídeos contendo anomalias. A composição final garante que aproximadamente 30% das trajetórias da validação correspondam a padrões anômalos, assegurando um cenário de avaliação mais representativo.

A arquitetura do gerador utiliza modelagem temporal com LSTM. A entrada consiste em um vetor de ruído de dimensão 128, transformado inicialmente por uma sequência de camadas lineares, normalização, função de ativação e *dropout*, resultando em um vetor projetado para dimensão 512. Esse vetor é expandido ao comprimento da sequência (vinte passos temporais) e processado por uma LSTM com 256 unidades ocultas e duas camadas empilhadas. A projeção de saída aplica novamente camadas lineares, normalização e ativação, finalizando com uma camada linear que mapeia para a dimensão das features da trajetória. A saída final é organizada em uma matriz tridimensional representando trajetórias sintéticas.

O treinamento segue a configuração clássica de WGAN-GP, conduzido ao longo de cinquenta épocas com tamanho de lote igual a 64. São utilizados otimizadores com taxas de aprendizado de 0.0001 para o gerador e 0.0004 para o discriminador. Mantém-se a proporção de cinco atualizações do discriminador para cada atualização do gerador. A função de perda do discriminador corresponde à distância de Wasserstein acrescida da penalidade de gradiente, enquanto a perda do gerador é o negativo do escore médio atribuído às amostras sintéticas. As atualizações dos gradientes incluem clipping, e o modelo é salvo sempre que apresenta melhoria na distância de Wasserstein.

A etapa de detecção de anomalias utiliza uma estratégia de pontuação combinada. O sistema integra a pontuação do discriminador de forma invertida, pois valores menores indicam maior desvio do padrão normal, e um escore de reconstrução baseado na menor distância entre uma trajetória real e todas as trajetórias sintéticas geradas no lote.

Figura 24 - Fluxograma da implementação de WGAN com Pytorch e camadas LSTM



Fonte: Os autores (2025).

4.4.3 Camadas LSTM com TensorFlow Keras

Com o objetivo de validar a generalização do modelo e permitir comparações entre *frameworks*, foi desenvolvida esta implementação paralela utilizando Keras do pacote TensorFlow, com um *fluxograma* de seu funcionamento na Figura 25. A metodologia baseia-se em aplicar também camadas LSTM e conjuntos de dados com balanceamento.

O processamento dos dados inicia-se com o carregamento, localizando os arquivos .npy contendo as trajetórias previamente extraídas de cada vídeo do *dataset*.

A divisão entre treinamento e validação utiliza 15% das trajetórias normais para compor a validação. O conjunto de treinamento é formado exclusivamente por trajetórias de vídeos normais, enquanto o conjunto de validação integra as trajetórias normais remanescentes e, adicionalmente, trajetórias classificadas como anômalas.

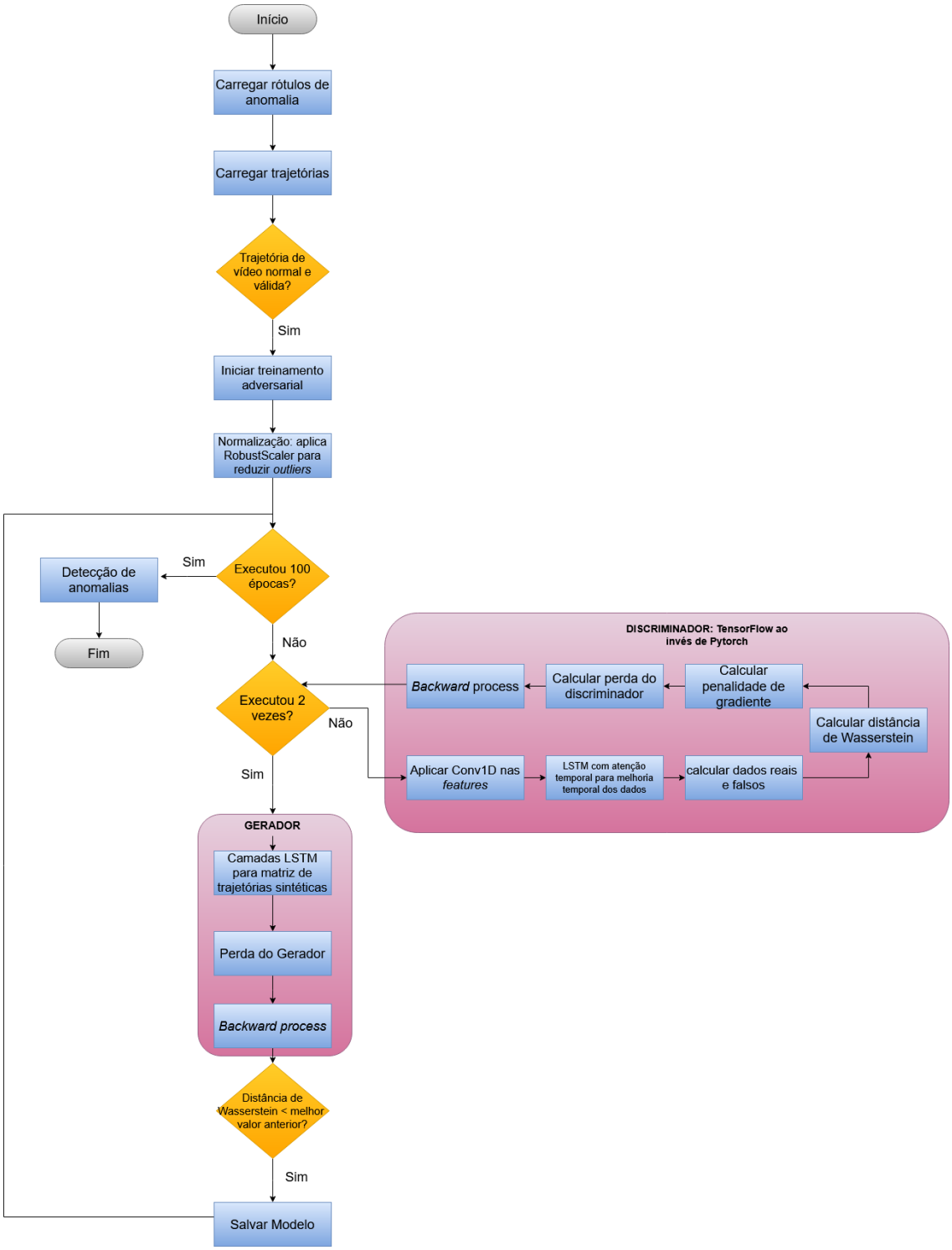
O módulo de filtragem de qualidade aplica verificações sequenciais em cada trajetória, a fim de eliminar ruídos e inconsistências. São descartadas trajetórias que apresentem valores nulos ou infinitos. Em seguida, realiza-se uma checagem de estacionalidade, onde trajetórias cuja soma das variâncias das posições em x e y seja inferior a 1×10^{-6} são desconsideradas. O algoritmo também ignora trajetórias que apresentem velocidades acima de 10 quadros por segundo, ou desvio padrão de velocidade maior que 5.

Para reduzir a influência de valores discrepantes, a normalização é realizada com o RobustScaler, garantindo maior estabilidade em trajetórias com alta variabilidade.

A arquitetura do gerador é composta por camadas lineares e uma sequência de camadas LSTM, responsáveis pela modelagem das dependências temporais. A dimensão do vetor de ruído de entrada é definida em 128.

O discriminador possui uma arquitetura híbrida que combina convoluções de uma dimensão (*Conv1D*) com kernel de tamanhos 3 e 5, cujas saídas são concatenadas e encaminhadas a uma camada LSTM, seguida por uma camada linear final. Essa combinação visa capturar tanto padrões locais quanto dependências temporais em múltiplas escalas.

Figura 25 - Fluxograma da implementação de WGAN com TensorFlow e camadas LSTM



Fonte: Os autores (2025).

A fase de treinamento é configurada para 100 épocas, utilizando tamanho de lote igual a 64. São empregados otimizadores com taxas de aprendizado de 0.0002 para o gerador e 0.0001 para o discriminador. O processo de otimização segue um

regime alternado, no qual o discriminador é atualizado por duas iterações para cada lote de dados, enquanto o gerador é atualizado uma única vez.

A função de perda do discriminador baseia-se na distância de Wasserstein, definida pela diferença entre a média dos scores atribuídos às amostras reais e às geradas. A penalidade foi calculada usando os mesmos parâmetros aplicados na implementação com camadas LSTM e *framework* Pytorch.

A função de perda do Gerador corresponde ao negativo da média dos scores falsos, buscando aumentar a pontuação atribuída às amostras geradas. Os gradientes de ambas as redes passam por clipping para evitar explosões numéricas e garantir convergência estável.

Inicia-se a etapa de detecção de anomalias é realizada por meio da combinação da pontuação fornecida pelo discriminador e a distância de reconstrução entre trajetórias reais e geradas. O *score* do discriminador atua como indicador principal, sendo que valores menores refletem trajetórias que se afastam do comportamento normal aprendido pelo modelo. A combinação dessas métricas resulta em uma pontuação final de anomalia. Trajetórias cujo valor combinado excede um limite pré-definido são classificadas como anômalas, permitindo a identificação automática de padrões atípicos no tráfego.

4.4.4 Sintese da evolução arquitetural

A evolução arquitetural do sistema de detecção de anomalias ocorreu em três diferentes etapas sucessivas, demandadas por ajustes estruturais e pela necessidade de melhorar os resultados obtidos.

A primeira implementação, de camadas lineares utilizando PyTorch, foi estruturado o gerador por meio de camadas lineares organizadas em uma arquitetura *feedforward*. O discriminador adotou uma configuração baseada em convoluções unidimensionais. O treinamento ocorreu ao longo de trinta épocas com lote de 128, e a pontuação de anomalia derivou exclusivamente da saída do discriminador, sem integração de métricas auxiliares.

A segunda implementação, de camadas LSTM com o *framework* PyTorch, introduziu modelagem temporal para elevar a capacidade do sistema de representar dependências sequenciais. Essa implementação adicionou as camadas LSTM no gerador, permitindo capturar relações de ordem entre os elementos das séries. O

discriminador utilizou uma estrutura híbrida composta por Conv1D e LSTM bidirecional. O treinamento estendeu-se a cinquenta épocas com lote de 64, e os conjuntos de dados foram balanceados para reduzir assimetrias entre as classes.

A terceira e última implementação, com camadas LSTM e *framework* TensorFlow, avaliou a generalização do modelo ao comparar os resultados obtidos entre as bibliotecas PyTorch e TensorFlow. O gerador integrou camadas lineares com LSTM para a modelagem de sequências, enquanto o discriminador manteve a estrutura híbrida baseada em Conv1D e LSTM. O processamento aplicou normalização com RobustScaler, reduzindo o impacto de valores que destoam no conjunto de dados. O treinamento foi conduzido por cem épocas com lote de 64.

Em conjunto, estas três implementações evidenciam uma transição gradual de estruturas mais diretas para arquiteturas fundamentadas em modelagem temporal e processamento em múltiplas escalas, culminando na análise entre diferentes bibliotecas Python como etapa de validação da consistência e estabilidade do sistema.

Para sintetizar as diferentes abordagens realizadas para o sistema de detecção de anomalias, a Tabela 2 abaixo apresenta o resumo de cada arquitetura desenvolvida:

Tabela 2 - Sintese da evolução arquiteturial

Etapa	Pacote	Componente Central
Versão Linear (PyTorch)	PyTorch	Camadas Lineares
Versão LSTM (PyTorch)	PyTorch	Camadas LSTM
Versão TensorFlow.Keras	TensorFlow.Keras	Gerador Dense + LSTM

Fonte: Os autores (2025).

4.5 Pós-processamento e avaliação

O módulo final é responsável pelo refinamento dos resultados e pela avaliação quantitativa do sistema. O *anomaly scoring* é ajustado para reduzir falsos positivos, e os eventos detectados são analisados com base em continuidade temporal e consistência espacial.

As métricas empregadas para a avaliação são *F1-Score* e AUROC (Área sob a Curva ROC), permitindo mensurar tanto a precisão quanto a capacidade discriminativa do modelo na distinção entre comportamentos normais e anômalos.

5 RESULTADOS

Este capítulo apresenta a discussão dos resultados obtidos nas etapas de extração e processamento de trajetórias, seguidas da avaliação quantitativa das diferentes implementações da rede adversarial Wasserstein GAN com Gradiente de Penalidade (WGAN-GP). Todos os experimentos foram realizados sobre os vídeos de tráfego da *Track* 4 - *Traffic Anomaly Detection* do *AI City Challenge* 2021.

5.1 Extração de trajetórias

A etapa de extração de trajetórias foi executada sobre um conjunto de 100 vídeos de tráfego urbano, cada um contendo cenas com diferentes níveis de fluxo veicular, iluminação e ângulos de câmera. Cada vídeo foi processado individualmente por meio do módulo de rastreamento, responsável por associar as detecções quadro a quadro, formando trilhas temporais correspondentes aos veículos observados.

Durante o processamento, cada detecção recebeu um identificador único de trilha (track_id), e suas informações espaciais, temporais e de confiança foram registradas em formato JSON. Cada arquivo resultante contém tanto os metadados do vídeo quanto o conjunto completo de trajetórias detectadas.

O formato de saída padronizado para os arquivos JSON contempla os seguintes campos principais:

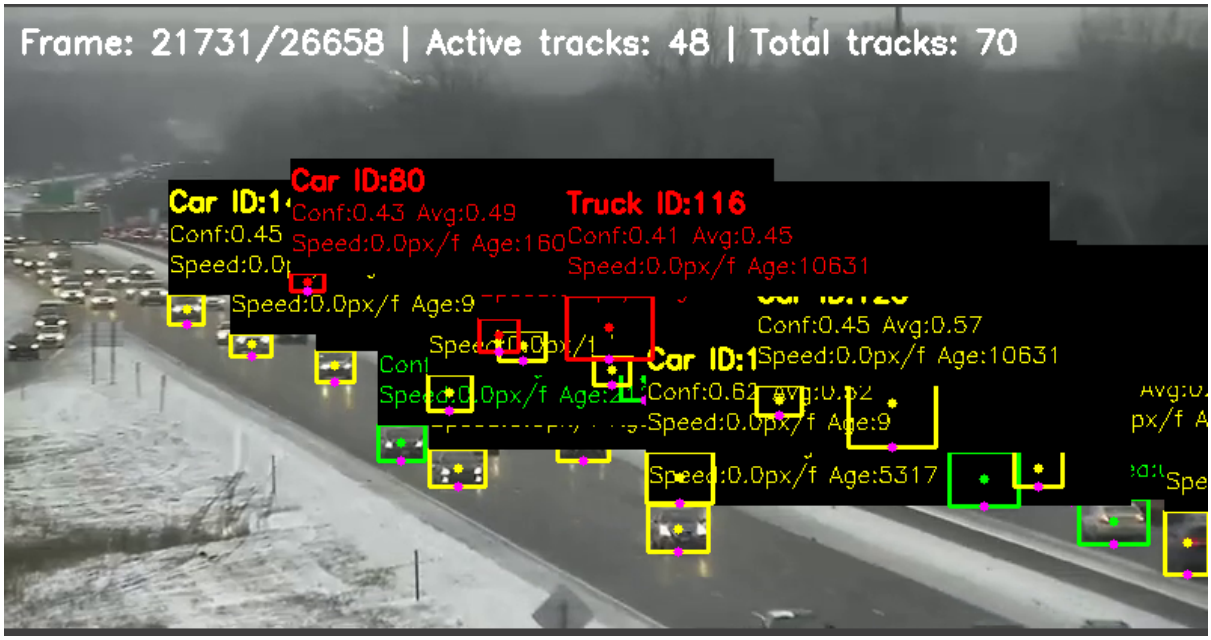
- total_frames: número total de quadros processados no vídeo;
- fps: taxa de quadros por segundo;
- video_path: caminho de origem do vídeo processado;
- total_tracks: quantidade total de trajetórias válidas detectadas;
- min_track_length: comprimento mínimo considerado aceitável para validação de trajetórias;
- tracks: lista de objetos que descrevem, para cada identificador de trajetória, as informações por quadro, incluindo *bounding box*, posição central, velocidade instantânea, confiança média e classe veicular.

As detecções e *bounding boxes* se mostram confiáveis em vídeos apresentaram condições favoráveis à detecção e rastreamento, sendo plenamente aproveitáveis nas etapas seguintes. Entre esses, destacam-se os vídeos 11 e 14,

cujas condições de iluminação, contraste e estabilidade da câmera favoreceram a extração de trajetórias completas.

O vídeo 11, representado na Figura 26, demonstra desempenho semelhante, com detecção quase integral dos veículos no campo de visão. Apenas os veículos mais distantes da câmera não foram reconhecidos pelo sistema de rastreamento. As trilhas resultantes mostraram continuidade satisfatória e baixa taxa de interrupção.

Figura 26 - Demonstração de vídeo 11 detectando veículos

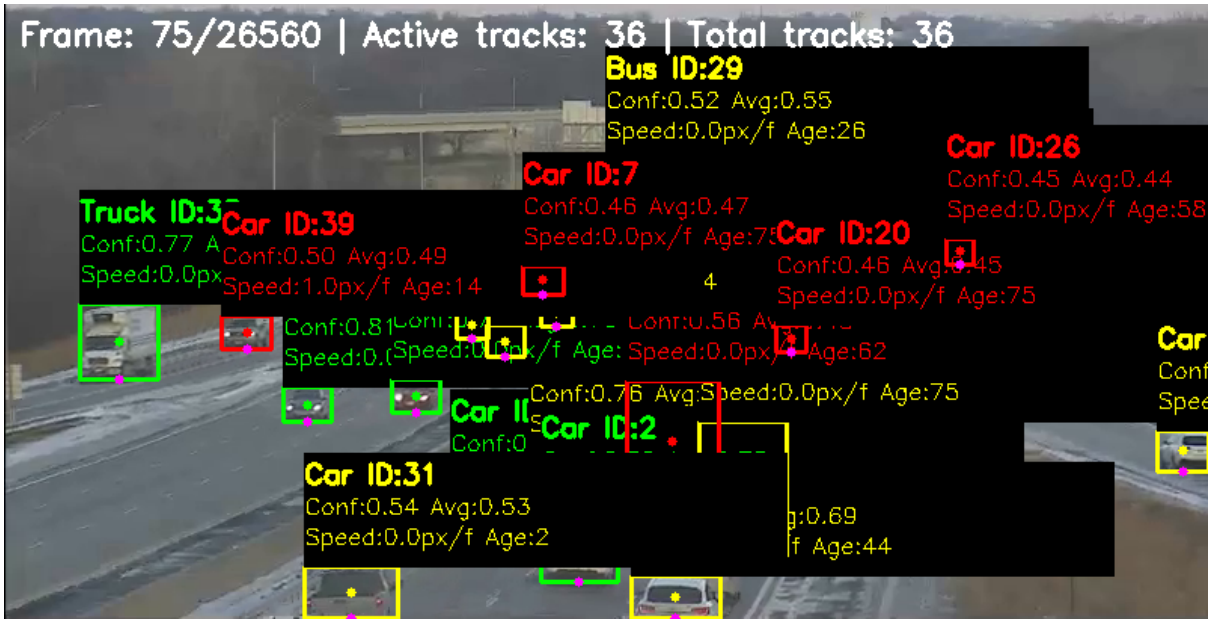


Fonte: Os autores (2025).

O vídeo 14 apresenta imagens de alta definição e iluminação uniforme, possibilitando uma detecção consistente dos veículos ao longo da cena.

Na Figura 27, verifica-se que o sistema foi capaz de identificar a maior parte dos veículos presentes no quadro, mesmo com pequenas variações de confiabilidade durante os deslocamentos.

Figura 27 - Demonstração de vídeo 14 detectando veículos



Fonte: Os autores (2025).

Contudo, dos 100 vídeos processados, houveram vídeos com extrações de trajetórias que prejudicaram as etapas subsequentes, em razão de limitações associadas à distância da câmera, baixa luminosidade, instabilidade de gravação ou ofuscamento por faróis.

O vídeo 21, por exemplo, apresenta uma câmera posicionada a longa distância da via, resultando em borrões e falta de nitidez nas imagens. A extração de trajetórias foi afetada principalmente pela baixa visibilidade dos veículos mais distantes.

Na Figura 28, observa-se que o sistema conseguiu detectar corretamente o veículo situado sobre o viaduto, na região mais próxima da câmera. Contudo, o caminhão presente na mesma cena, mas em área mais distante, não foi identificado.

A Figura 29 ilustra outro exemplo de falha de detecção, em que um veículo mais afastado permanece não rastreado ao longo de toda a sequência.

Figura 28 28 - Demonstração de vídeo 21 com baixa visibilidade detectando um veículo

Fonte: Os autores (2025).

Figura 29 29 - Demonstração de vídeo 21 com baixa visibilidade não detectando um veículo

Fonte: Os autores (2025).

O vídeo 31 apresenta iluminação reduzida e forte ofuscamento causado pelos faróis dos automóveis, dificultando o reconhecimento dos contornos veiculares.

Na Figura 30, é possível observar a baixa visibilidade geral da cena. As detecções ocorreram predominantemente em veículos mais próximos à câmera, como exemplificado na Figura 31, o que comprometeu a continuidade e a consistência das trajetórias formadas.

Figura 30 - Demonstração de vídeo 31 com baixa visibilidade e com ofuscamento

Fonte: Os autores (2025).

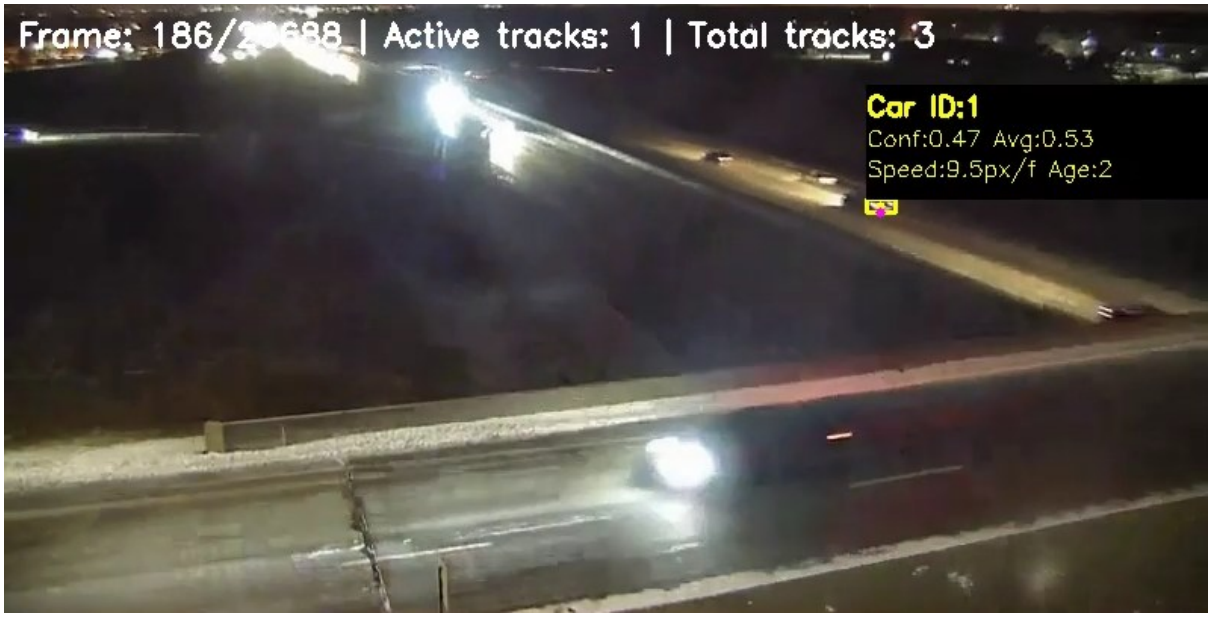
Figura 31 - Demonstração de vídeo 31 não detectando um veículo

Fonte: Os autores (2025).

O vídeo 45 apresenta configuração semelhante, com a câmera posicionada em um ponto elevado e distante da pista, além de registrar baixa luminosidade e reflexos intensos das luzes veiculares.

Na Figura 32, observa-se o enquadramento distante e o predomínio de regiões escuras. A maioria das detecções ocorreu sobre o viaduto localizado acima da via principal, conforme demonstrado na Figura 33, resultando em trilhas fragmentadas e incompletas.

Figura 32 32 - Demonstração de vídeo 45 com baixa visibilidade e com ofuscamento



Fonte: Os autores (2025).

Figura 33 33 - Demonstração de vídeo 45 detectando carro próximo a câmera

Fonte: Os autores (2025).

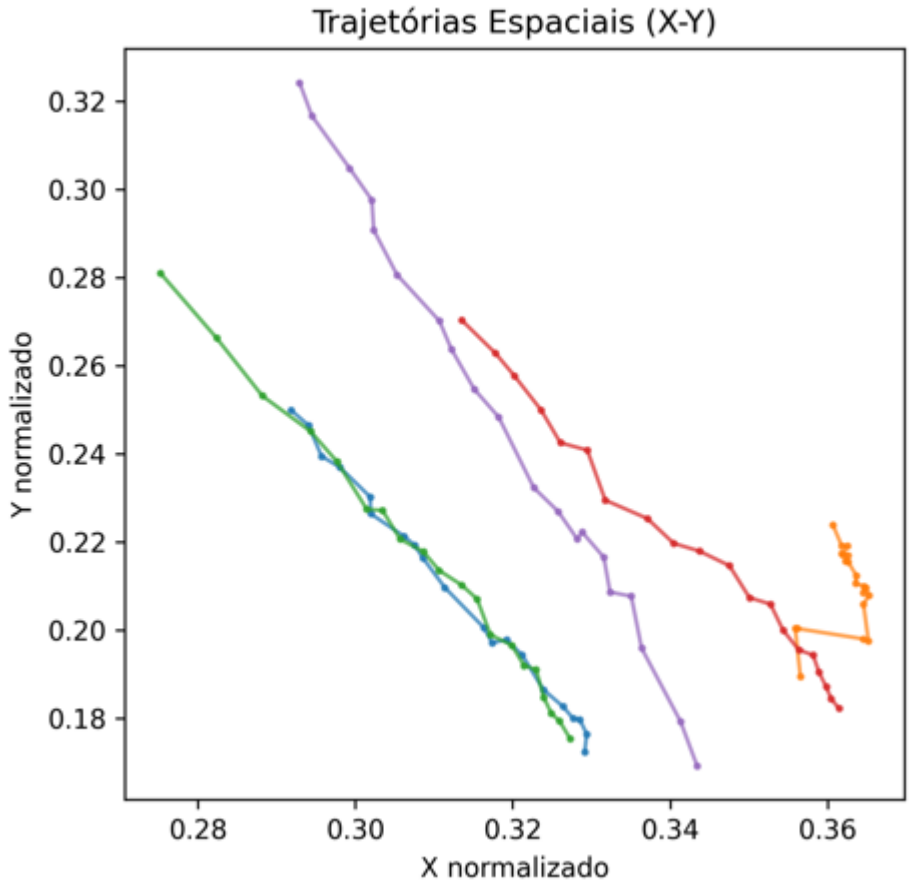
5.2 Processamento de trajetórias

O módulo de processamento de trajetórias foi responsável pela normalização e transformação dos dados extraídos na etapa anterior. Cada trajetória foi interpolada temporalmente para um comprimento fixo de 20 pontos, permitindo padronização para posterior análise em redes neurais.

Durante o processamento, foram extraídas cinco características principais: posição (x, y), velocidade, aceleração e direção.

A normalização, como demonstrada na Figura 34 abaixo, mantendo as coordenadas espaciais no intervalo [0,1] e as direções normalizadas para o intervalo [-1,1].

Figura 34 - Demonstração de normalização de trajetórias para o vídeo 1



Fonte: Os autores (2025).

Adicionalmente, filtros de qualidade foram aplicados para remover trajetórias inválidas, estacionárias ou com valores extremos fora do limite estatístico global.

O resultado foi salvo em arquivos NumPy, contendo matrizes tridimensionais na forma (trajetória × tempo × característica). Esses arquivos representam a base de entrada para o treinamento e a avaliação dos modelos WGAN-GP.

5.3 Avaliação quantitativa

A avaliação quantitativa é realizada sobre as métricas de classificação binária de *precision*, *recall*, *F1-Score*, *accuracy*, AUC-ROC e AUC-PR. A análise abrange as três principais variantes do modelo Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP), visando a mensuração da capacidade discriminativa na distinção entre comportamentos normais e anômalos. A entrada para os modelos

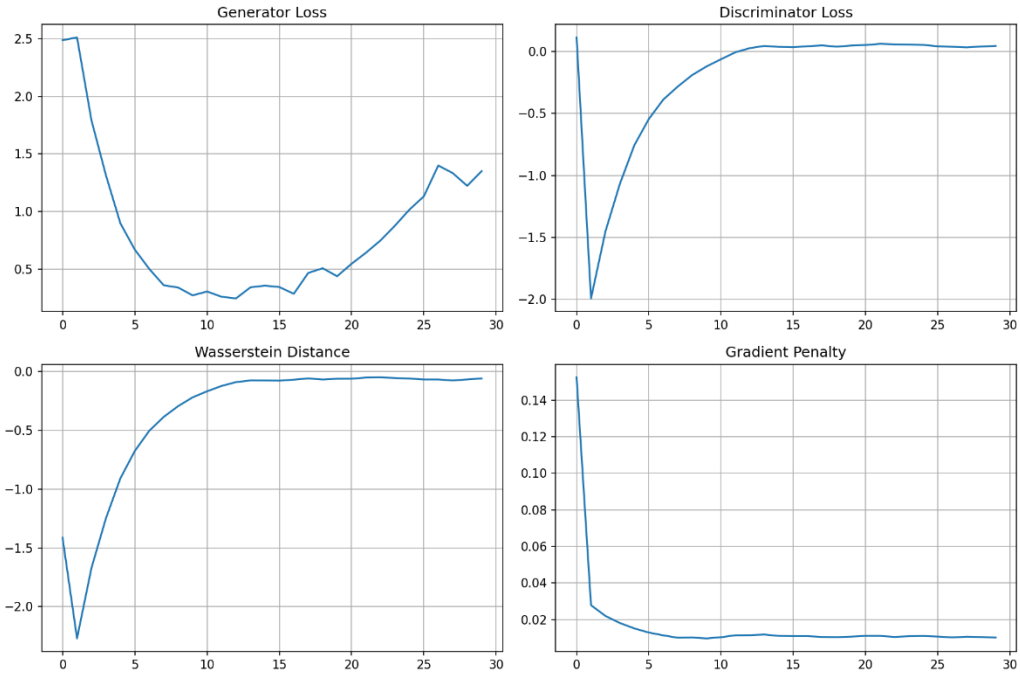
consiste em matrizes tridimensionais, resultantes do processamento de trajetórias, padronizadas para vinte pontos temporais e cinco características cinemáticas.

Os resultados apresentados a seguir correspondem às três principais variantes do modelo WGAN-GP: uma com camadas lineares, outra com camadas LSTM e uma terceira implementada em TensorFlow/Keras para análise comparativa entre bibliotecas.

5.3.1 Implementação linear (PyTorch)

A primeira abordagem implementou uma arquitetura WGAN-GP utilizando PyTorch. Esta versão aplicou camadas lineares na rede geradora e uma arquitetura Conv1D na rede discriminadora, priorizando a otimização da implementação. O treinamento, demonstrado pela Figura 35, foi estabelecido para trinta épocas, utilizando um tamanho de lote (*batch size*) de 128, com taxas de aprendizado definidas em 0.0002 para o gerador e 0.0001 para o discriminador. A detecção de anomalias nesta variante dependeu exclusivamente da pontuação de saída do discriminador, invertida para que scores menores indicassem maior desvio do padrão normal.

Figura 35 - *Training history* para a implementação com Pytorch e camadas lineares



Fonte: Os autores (2025).

O desempenho desta implementação linear apresentou valores reduzidos para a detecção de anomalias. A *F1-Score* atingiu 0.0946, e o AUC-ROC resultou em 0.4399. Este desempenho indica uma limitação intrínseca da arquitetura em camadas unicamente lineares para capturar as dependências temporais complexas inerentes aos dados de trajetórias.

Tabela 3 - Resultados da implementação de Pytorch Linear

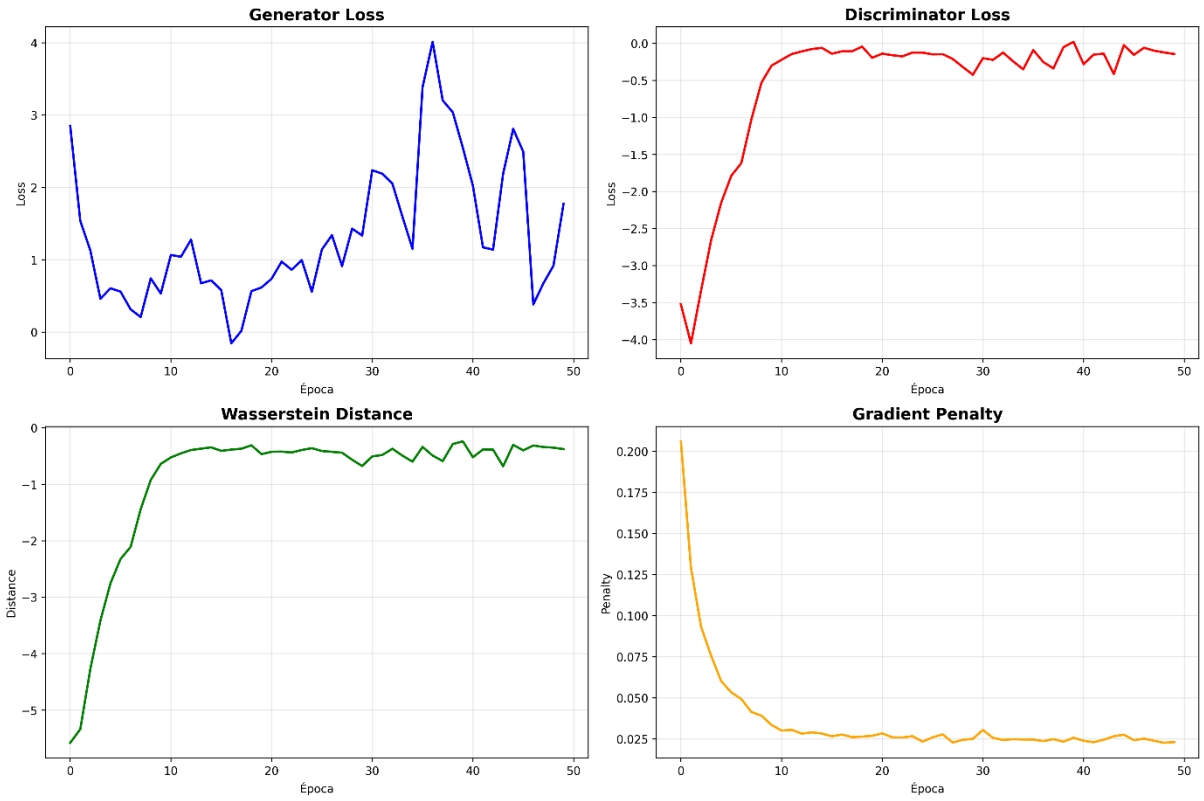
Métrica	**Valor**
Threshold	1.4720
Precision	0.1094
Recall	0.0833
F1-Score	0.0946
Accuracy	0.7905
AUC-ROC	0.4399
AUC-PR	0.1131

Fonte: Os autores (2025).

5.3.2 Implementação LSTM (PyTorch)

A segunda variante incorporou o aprimoramento da modelagem temporal por meio da introdução de camadas *Long Short-Term Memory* (LSTM). O gerador utilizou uma LSTM com duzentas e cinquenta e seis unidades ocultas e duas camadas empilhadas, processando um vetor de ruído de dimensão 128. O discriminador adotou uma arquitetura híbrida de Conv1D com LSTM bidirecional para apreender dependências temporais. O treinamento, demonstrado na Figura 36, foi estendido para cinquenta épocas, com tamanho de lote de 64, utilizando taxas de aprendizado de 0.0001 para o gerador e 0.0004 para o discriminador. A validação utilizou um conjunto de dados balanceado, contendo aproximadamente 30% de trajetórias anômalas.

Figura 36 - *Training history* para a implementação com Pytorch e camadas LSTM



Fonte: Os autores (2025).

A pontuação de anomalia nesta arquitetura combinou duas métricas: o score invertido do discriminador (70%) e um score de reconstrução (30%) baseado na menor distância entre uma trajetória real e as amostras sintéticas geradas. Esta metodologia demonstrou maior capacidade em representar padrões dinâmicos. A *F1-Score* alcançou 0.2897, e o AUC-ROC resultou em 0.6329. A melhoria significativa nas métricas AUC-ROC e *F1-Score*, em comparação com a versão linear, valida a importância da inclusão de camadas recorrentes para o processamento de sequências temporais.

Tabela 4 - Resultados da implementação de Pytorch com LSTM

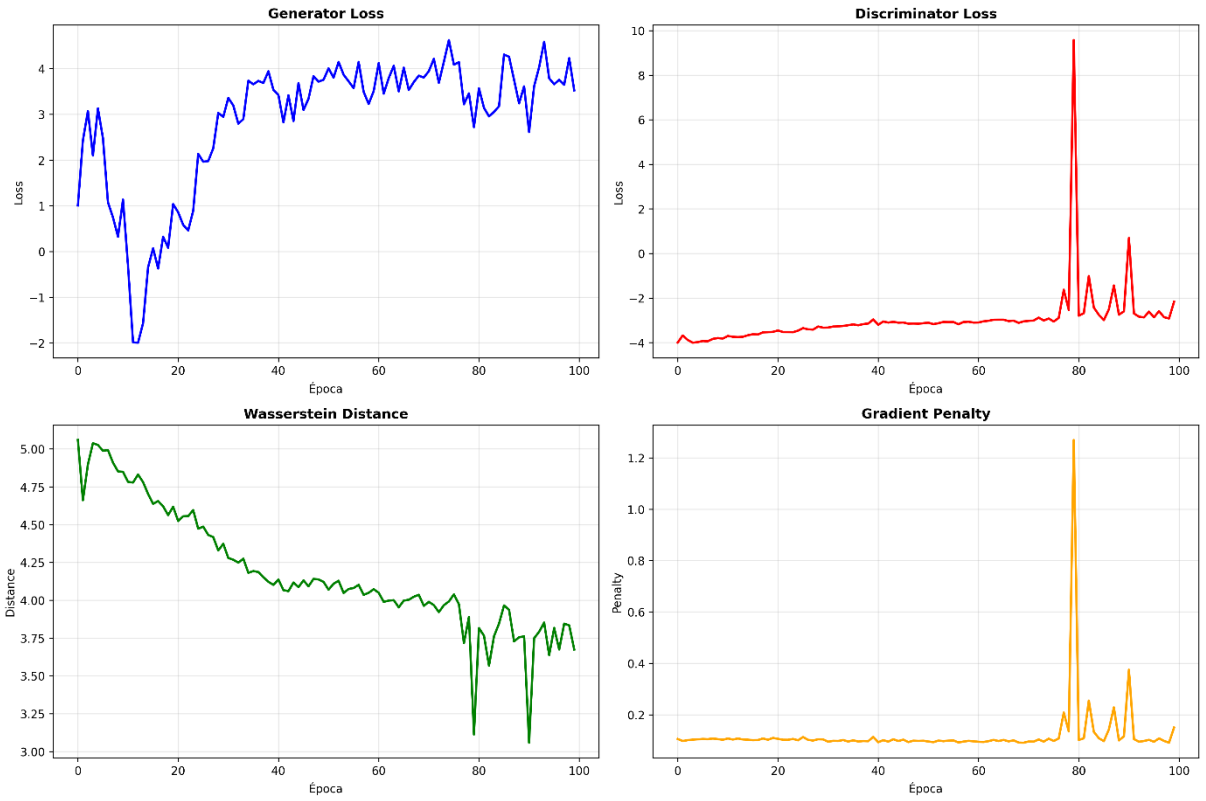
Métrica	**Valor**
Threshold	0.6390
Precision	0.4345
Recall	0.2172
F1-Score	0.2897
Accuracy	0.6804
AUC-ROC	0.6329
AUC-PR	0.4034

Fonte: Os autores (2025).

5.3.3 Implementação TensorFlow Keras

Com o objetivo de validar a generalização do modelo e permitir a análise entre bibliotecas, uma implementação paralela foi desenvolvida utilizando Keras do pacote TensorFlow. Esta versão utilizou um gerador composto por camadas densas e LSTM, e um discriminador com arquitetura híbrida de Conv1D e LSTM, visando capturar padrões locais e temporais em múltiplas escalas. O pré-processamento aplicou o RobustScaler para a normalização, mitigando a influência de outliers na alta variabilidade das trajetórias. O treinamento foi configurado para 100 épocas e está demonstrado na Figura 37.

Figura 37 - *Training history* para a implementação com Keras e camadas LSTM



Fonte: Os autores (2025).

Esta implementação intermediária alcançou *F1-Score* de 0.2239 e AUC-ROC de 0.5535. O desempenho se situou em um ponto intermediário entre as implementações LSTM (PyTorch) e Linear (PyTorch), reforçando a conclusão de que a modelagem temporal é um fator determinante para a eficácia do sistema.

Tabela 5 - Resultados da implementação de TensorFlow Keras

Métrica	**Valor**
Threshold	0.0714
Precision	0.3606
Recall	0.1624
F1-Score	0.2239
Accuracy	0.6249
AUC-ROC	0.5535
AUC-PR	0.3706

Fonte: Os autores (2025).

5.4 Análise dos resultados das implementações WGAN-GP

A comparação entre as três implementações evidencia diferenças significativas nos comportamentos de aprendizado e generalização.

O modelo LSTM (PyTorch) apresentou o desempenho superior entre as variantes arquitetônicas analisadas. A capacidade de modelar sequências temporais inerentes às camadas LSTM demonstrou ser crucial para distinguir as trajetórias normais das anômalas, um requisito fundamental para a detecção de anomalias em tráfego.

Entretanto, as métricas obtidas indicam a necessidade de refinamentos no sistema global. O desempenho do sistema de detecção de anomalias sofreu limitações significativas devido à propagação de erros oriundos do módulo de extração de trajetórias. Falhas no rastreamento, causadas por condições adversas como baixa luminosidade, ofuscamento por faróis e distância elevada da câmera, resultaram em trilhas fragmentadas ou inconsistentes. Trajetórias inválidas ou ruidosas comprometem a qualidade dos dados de treinamento e validação, impactando a capacidade do WGAN-GP de aprender a distribuição normal de forma robusta.

5.5 Análise comparativa com os competidores do *AI city challenge*

A fim de contextualizar o desempenho do modelo proposto, uma comparação com a tabela de classificação do *AI City Challenge Track 4.* A Tabela 6 apresenta os resultados obtidos pelos 9 mais bem colocados no desafio.

Tabela 6 – Classificação *AI city challenge track* 4

Classificação	ID da equipe	Nome da equipe	Pontuação
1	76	BaiduVIS&SIAT	0.9355
2	158	BD	0.922
3	92	WHU-IIP	0.9197
4	90	SIS Lab	0.8597
5	153	Titan Mizzou	0.5686
6	48	BUPT-MCPRL2	0.289
7	26	Attention Please!	0.2184
8	154	Alchera	0.1418
9	12	CET	0.1401

Fonte: Adaptado de Zhao et al. (2021)

A métrica utilizada para avaliar os modelos foi o *S4 Score*, descrito no artigo de Zhao et al. (2021), calculada a partir do *F1-Score* e *RMSE* conforme equação 5.5.1.

$$S4 = F1 * (1 - RMSE) \tag{5.5.1}$$

Foi calculada a pontuação da implementação LTSM via *PyTorch*, o modelo com os melhores resultados. O *F1-score* foi computado junto dos resultados apresentados na tabela 6. O cálculo do RMSE foi realizado para as predições de verdadeiros positivos desse modelo. Assim, foi calculado o S4 Score apresentado na Tabela 7.

Tabela 7 – Resultados obtidos

F1	RMSE	Pontuação S4
0.2897	0.3115	0.1995

Fonte: Os autores (2025)

Ao comparar os resultados obtidos na Tabela 7 e a classificação do *AI City Challenge Track* 4 (Tabela 6), nota-se que o modelo implementado alcançaria a oitava posição na classificação oficial.

6 CONCLUSÃO

O trabalho desenvolveu um sistema para a detecção de anomalias em vias urbanas. A pesquisa propôs uma abordagem baseada em Redes Adversárias Generativas Wasserstein (WGAN) com Penalidade de Gradiente (WGAN-GP), aprendizado de máquina e visão computacional. A metodologia empregou a base de dados da *track* 4 do *AI City Challenge* de 2021 e dados sintéticos derivados para o treinamento. A sequência de desenvolvimento iniciou com a análise do contexto da segurança viária e a revisão de técnicas computacionais relevantes. Posteriormente, o sistema foi estruturado em quatro módulos: rastreamento de veículos, processamento de trajetórias, detecção de anomalias com WGAN-GP e pós-processamento e avaliação de resultados. Implementou-se três variações arquiteturais do modelo WGAN-GP para detecção de anomalias: camadas lineares, camadas LSTM com PyTorch e camadas LSTM com TensorFlow Keras.

Considerando os objetivos específicos deste trabalho, a pesquisa realizou a análise da teoria relacionada a modelos de aprendizado de máquina e métricas de avaliação, por meio do estudo de conceitos de visão computacional, redes neurais, WGAN e métricas como *F1-Score* e AUROC. O banco de dados foi definido pelo conjunto da *track* 4 do *AI City Challenge* de 2021. O trabalho realizou a implementação dos modelos, explorando arquiteturas com camadas lineares e camadas LSTM nos pacotes PyTorch e TensorFlow Keras. O plano experimental envolveu a extração e o processamento das trajetórias, seguida pelo treinamento e avaliação dos modelos de aprendizado de máquina. A avaliação utilizou métricas de desempenho como *F1-Score* e AUC-ROC.

A implementação com camadas LSTM utilizando PyTorch apresentou o desempenho mais elevado entre as variações arquiteturais, alcançando *F1-Score* de 0.2897 e AUC-ROC de 0.6329. A avaliação quantitativa, utilizando a métrica S4 *Score*, resultou em uma pontuação de 0.1995. Este resultado demonstra que o modelo alcançaria a oitava posição na classificação oficial da *track* 4 do *AI City Challenge* de 2021. A arquitetura baseada em WGAN-GP demonstrou ser aplicável para a detecção de anomalias em trajetórias. O trabalho confirma a capacidade do modelo de aprender a distribuição de dados normais e de usar a geração de dados sintéticos para o treinamento, o que é útil em situações de escassez de dados anômalos. Contudo, os resultados obtidos indicaram que o modelo apresentou desempenho satisfatório

apenas em cenários com vídeos estáveis e baixo nível de oclusões, devido às limitações do sistema.

6.1 Limitações

Os principais fatores que foram identificados como limitadores do desempenho do sistema de detecção de anomalias foram a propagação de erros do módulo de extração de trajetórias. Ocorreram falhas na detecção em vídeos com condições adversas, especificamente baixa luminosidade, ofuscamento por faróis e distância elevada da câmera.

Vídeos também com falhas devido a momentos corrompidos e falhas também se mostraram potenciais limitantes e podem ter cooperado para o mal desempenho do *framework* proposto neste trabalho.

6.2 Trabalhos futuros

A partir das limitações identificadas, propõem-se nesta seção direções para trabalhos futuros. Recomenda-se o aprimoramento do módulo de rastreamento de veículos, visando aumentar a resiliência do sistema em condições de baixa visibilidade e ofuscamento. Sugere-se a pesquisa por algoritmos de detecção e rastreamento mais precisos para mitigar as falhas de extração.

Adicionalmente, propõe-se a otimização da arquitetura WGAN-GP. Por fim, indica-se a realização de testes para a resolução de problemas complexos, como oclusões parciais de veículos e a eventual fusão com dados de outros sensores para enriquecer a análise das trajetórias.

REFERÊNCIAS

ARJOVSKY, Martin; CHINTALA, Soumith; BOTTOU, Léon. Wasserstein GAN. 26 jan. 2017.

AZFAR, Talha *et al.* Deep Learning-Based Computer Vision Methods for Complex Traffic Environments Perception: A Review. **Data Science for Transportation**, v. 6, n. 1, p. 1, 8 abr. 2024.

BRASIL. MINISTÉRIO DO DESENVOLVIMENTO REGIONAL. **Ministro afirma que investir em mobilidade urbana é garantir a qualidade de vida da população**, 2014. Disponível em: https://www.gov.br/mdr/pt-br/noticias/ministro-afirma-que-investir-em-mobilidade-urbana-e-garantir-a-qualidade-de-vida-da-populacao.

CABRAL DE AZEVEDO, Rogério; ENSSLIN, Leonardo. **METODOLOGIA DA PESQUISA PARA ENGENHARIAS**.

CARRARA, Fabio *et al.* Combining GANs and AutoEncoders for Efficient Anomaly Detection. 16 nov. 2020.

CHANDOLA, Varun; BANERJEE, Arindam; KUMAR, Vipin. Anomaly detection. **ACM Computing Surveys**, v. 41, n. 3, p. 1–58, 30 jul. 2009.

CHEN, Jingyuan *et al.* **Dual-Modality Vehicle Anomaly Detection via Bilateral Trajectory Tracing**. Disponível em: https://arxiv.org/abs/2106.05003.

CRESWELL, Antonia *et al.* Generative Adversarial Networks: An Overview. **IEEE Signal Processing Magazine**, v. 35, n. 1, p. 53–65, jan. 2018.

DAWANI, Jay. **Hands-on mathematics for deep learning : build a solid mathematical foundation for training efficient deep neural networks**. Packt Publishing, 2020.

EVERINGHAM, Mark *et al.* The Pascal Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**, v. 88, n. 2, p. 303–338, 9 jun. 2010.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern Recognition Letters**, v. 27, n. 8, p. 861–874, fev. 2006.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. Upper Saddle River, N.J.: Prentice Hall, 2008.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. MIT Press, 2016.

GOODFELLOW, Ian J. *et al.* Generative Adversarial Networks. 10 jun. 2014.

HAYKIN, S. S. **Neural Networks and Learning Machines**. Pearson, 2009.

HE, Yihui *et al.* Bounding Box Regression With Uncertainty for Accurate Object Detection. *In*: IEEE, jun. 2019.

INSTITUTO DE PESQUISA ECONÔMICA APLICADA. **Estudo aponta aumento de 13,5% em mortes no trânsito**, 2023. Disponível em: https://www.ipea.gov.br/portal/categorias/45-todas-as-noticias/noticias/13899-estudo-aponta-aumento-de-13-5-em-mortes-no-transito.

LUO, Wenhan *et al.* Multiple object tracking: A literature review. **Artificial Intelligence**, v. 293, p. 103448, abr. 2021.

MINISTÉRIO DOS TRANSPORTES DO BRASIL. **OMS lança plano para reduzir acidentes e mortes no trânsito até 2030**, 2021. Disponível em: https://www.gov.br/transportes/pt-br/assuntos/noticias/2021/10/oms-lanca-plano-para-reduzir-acidentes-e-mortes-no-transito-ate-2030#:~:text=A%20Organiza%C3%A7%C3%A3o%20Mundial%20da%20Sa%C3%BAde.

MINISTÉRIO DOS TRANSPORTES DO BRASIL. **Investimento em rodovias é essencial para preservar vidas, diz secretário nacional de Trânsito**, 2023. Disponível em: https://www.gov.br/transportes/pt-br/assuntos/noticias/2023/06/investimento-em-rodovias-e-essencial-para-preservar-vidas-diz-secretario-nacional-de-transito.

MIRZA, Mehdi; OSINDERO, Simon. **Conditional Generative Adversarial Nets**. arXiv, nov. 2014. Disponível em: http://arxiv.org/abs/1411.1784.

MITCHELL, T. M. **Machine Learning**. McGraw-Hill Education, 1997.

NAPHADE, Milind *et al.* The 5th AI City Challenge. 25 abr. 2021.

PANG, Bo *et al.* TubeTK: Adopting Tubes to Track Multi-Object in a One-Step Training Model. 10 jun. 2020.

PRAMODITHA, Rukshan. **The Concept of Artificial Neurons (Perceptrons) in Neural Networks | Towards Data Science**. Disponível em: https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc/.

PRINCE, S. J. D. **Computer Vision: Models Learning and Inference**. Cambridge University Press, 2012.

RUSSELL, S. J.; NORVIG, P.; DAVIS, E. **Artificial Intelligence: A Modern Approach**. Prentice Hall, 2010.

SABUHI, Mikael *et al.* **Applications of Generative Adversarial Networks in Anomaly Detection: A Systematic Literature Review**. **IEEE Access**Institute of Electrical and Electronics Engineers Inc., 2021.

SILVA, Fernando Nunes da. Mobilidade urbana: os desafios do futuro. **Cadernos Metrópole**, v. 15, n. 30, p. 377–388, dez. 2013.

SITARZ, Mikołaj. Extending F1 metric, probabilistic approach. **Advances in Artificial Intelligence and Machine Learning**, v. 3, n. 2, p. 1025–1038, fev. 2022.

STANFORD. **CS231n Convolutional Neural Networks for Visual Recognition**. Disponível em: https://cs231n.github.io/neural-networks-1/. Acesso em: 25 jan. 2025.

SZELISKI, Richard. **Computer Vision: Algorithms and Applications 2nd Edition**, 2021. Disponível em: https://szeliski.org/Book.

VILLANI, Cédric. **Optimal Transport**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. v. 338

WORLD HEALTH ORGANIZATION. **Road traffic injuries**, 2023. Disponível em: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

WORLD INTELLECTUAL PROPERTY ORGANIZATION. **What is Intellectual Property?**, 2023. Disponível em: https://www.wipo.int/about-ip/en/.

WU, Jie *et al.* **Box-Level Tube Tracking and Refinement for Vehicles Anomaly Detection**.

YADAV, Shashank Singh; VIJAYAKUMAR, Vaidehi; ATHANESIOUS, Joshan. **Detection of Anomalies in Traffic Scene Surveillance**. IEEE, 2018.

ZHANG, Aston *et al.* Dive into Deep Learning. 21 jun. 2021.

ZHANKAZIEV, S. V. *et al.* Predicting Traffic Accidents Using the Conflict Coefficient. *In*: Institute of Electrical and Electronics Engineers Inc., 2022.

ZHAO, Yuxiang *et al.* **Good Practices and A Strong Baseline for Traffic Anomaly Detection**.

ZHAO, Yuxiang *et al.* Practices and A Strong Baseline for Traffic Anomaly Detection. *In*: IEEE, jun. 2021b. Disponível em: https://ieeexplore.ieee.org/document/9522708/.

ZIVKOVIC, Z. Improved adaptive Gaussian mixture model for background subtraction. *In*: IEEE, 2004.