UFES

Universidade Federal do Espírito Santo
Centro Tecnológico - Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Reidentificação Baseada em Filtro de Correlação Discriminativo para Rastreamento de Múltiplos Objetos em Câmeras de Videomonitoramento

Augusto Abling

Vitória-ES, Abril 2025

Augusto Abling

Reidentificação Baseada em Filtro de Correlação Discriminativo para Rastreamento de Múltiplos Objetos em Câmeras de Videomonitoramento

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica na área de concentração Robótica, Controle e Automação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientadora: Profa. Dra. Raquel Frizera Vassallo

Vitória-ES

Abril 2025

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

---

Abling, Augusto, 1991-
A152r Reidentificação baseada em filtro de correlação discriminativo para rastreamento de múltiplos objetos em câmeras de videomonitoramento / Augusto Abling. - 2025.
108 f. : il.

Orientadora: Raquel Frizera Vassallo.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Cidades inteligentes. 2. Sistemas inteligentes de veículos rodoviários. 3. Sistemas de reconhecimento de padrões. 4. Processamento eletrônico de dados em tempo real. 5. Videovigilância. I. Frizera Vassallo, Raquel. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

Augusto Abling

Reidentificação Baseada em Filtro de Correlação Discriminativo para Rastreamento de Múltiplos Objetos em Câmeras de Videomonitoramento

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica na área de concentração Robótica, Controle e Automação.

Trabalho aprovado. Vitória-ES, 01 de abril de 2025:

Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo
Orientadora

Profa. Dra. Thais Pedruzzi do Nascimento
Universidade Federal do Espírito Santo
Examinadora Interna

Prof. Dr. Bruno Legora Souza da Silva
Universidade Federal do Espírito Santo
Examinador Interno

Prof. Dr. Douglas Almonfrey
Instituto Federal do Espírito Santo
Examinador Externo

Prof. Dr. Flávio Garcia Pereira
Instituto Federal do Espírito Santo
Examinador Externo

Vitória-ES

Abril 2025

AGRADECIMENTOS

A realização desta dissertação só foi possível graças à contribuição direta e indireta de algumas pessoas e instituições às quais gostaria de expressar minha sincera gratidão.

Primeiramente, agradeço aos meus pais, Dalva Teresinha Abling e Astor Abling, pelo amor incondicional, pelo incentivo constante e pela crença em meu potencial. Sou igualmente grato ao meu irmão, Adriano Abling, que sempre foi um exemplo de responsabilidade e dedicação ao trabalho. O apoio de vocês foi fundamental para que eu chegasse até aqui.

Um agradecimento especial vai para a minha namorada, Larrissa Ribeiro Rodrigues. Sua paciência, compreensão e apoio ao longo desta jornada foram essenciais. Sua presença se tornou um verdadeiro alicerce em momentos desafiadores.

Agradeço também à minha orientadora, Raquel Frizera Vassallo, pela orientação valiosa, pela confiança em meu trabalho e pela generosidade ao compartilhar seu conhecimento. O aprendizado que obtive no Laboratório de Visão Computacional (LabVisio), sob sua gestão, foi fundamental para enriquecer o conteúdo científico desta pesquisa.

Por fim, quero expressar minha gratidão à empresa Atman Systems, que me proporcionou desafios significativos em minha área de estudo, contribuindo de maneira decisiva para meu aprendizado e experiência. As oportunidades que tive lá foram fundamentais para a construção deste trabalho.

"O estudo profundo da natureza é a fonte mais fértil de descobertas matemáticas".

– Jean-Baptiste Joseph Fourier (1822)

RESUMO

Este estudo tem como objetivo desenvolver, testar e analisar a utilização de um filtro de correlação discriminativo como módulo de reidentificação de objetos, integrado ao rastreamento de múltiplos objetos, para uso em câmeras de videomonitoramento com foco em processamento em tempo real. O estudo se insere no contexto de cidades inteligentes e Sistemas de Transporte Inteligente (ITS, do inglês *Intelligent Transportation Systems*), onde a reidentificação e o rastreamento de objetos são fundamentais para a criação de tecnologias avançadas. A metodologia adotada incluiu a implementação de um filtro de correlação discriminativo, modificado para a tarefa de reidentificação, seguido de testes para avaliar o desempenho do algoritmo em cenários desafiadores, presentes em conjuntos de dados amplamente reconhecidos nos desafios de visão computacional. Os resultados demonstraram que o filtro de correlação proposto se aproxima, em precisão, das abordagens baseadas em redes neurais, sem a necessidade de treinamento prévio para contextos específicos. Conclui-se que a integração deste módulo de reidentificação com o rastreamento de múltiplos objetos oferece uma solução equilibrada para melhorar a precisão do rastreamento, a um custo computacional menor em comparação às redes neurais, contribuindo para o avanço de tecnologias em cidades inteligentes e ITS.

Palavras-chaves: Filtro de correlação discriminativo. Reidentificação de objetos. Rastreamento de múltiplos objetos. Visão computacional. Câmeras de videomonitoramento. Cidades inteligentes. Sistemas de transporte inteligente. Processamento em tempo real.

ABSTRACT

This study aims to develop, test, and analyze the use of discriminative correlation filter as a module for object re-identification, integrated with multiple object tracking for use in surveillance cameras with a focus on real-time processing. The study is set in the context of smart cities and Intelligent Transportation Systems (ITS), where object re-identification and tracking are fundamental for the creation of advanced technologies. The adopted methodology includes the implementation of a modified discriminative correlation filter for the re-identification task, followed by tests to evaluate the algorithm's performance in challenging scenarios present in widely recognized datasets in computer vision challenges. The results showed that the proposed correlation filter approaches the accuracy of neural network-based approaches without the need for prior training for specific contexts. Therefore, we may conclude that the integration of this re-identification module with multi-object tracking offers a balanced solution to improve tracking accuracy at a lower computational cost compared to neural networks, contributing to the advancement of technologies in smart cities and ITS.

Keywords: Discriminative correlation filter. Object re-identification. Multiple object tracking. Computer vision. Surveillance cameras. Smart cities. Intelligent transportation systems. Real-time processing.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AssA	Association Accuracy
CCOT	Continuous Convolution Operators Tracking (CCOT)
CLEAR	Classification of Events, Activities, and Relationships
CMC	Cumulative Matching Characteristic
CNN	Convolutional Neural Network
DCF	Discriminative Correlation Filter
DetA	Detection Accuracy
DFT	Discrete Fourier Transform
ECO	Efficient Convolution Operators
FFT	Fast Fourier Transform
FLOPs	Floating Point Operations
FPS	Frames Per Second
GT	Ground Truth
HOG	Histogram of Oriented Gradients
HOTA	Higher Order Tracking Accuracy
IBM	International Business Machines Corporation
IDP	Identification Precision
IDR	Identification Recall
IDSW	ID Switches
IFFT	Inverse Fast Fourier Transform
IoT	Internet of Things
IoU	Intersection over Union
ITS	Intelligent Transportation Systems

JDE	Joint Detection and Embedding
KNN	K-Nearest Neighbor
LBP	Local Binary Patterns
LocA	Localization Accuracy
KCF	Kernelized Correlation Filter
mAP	mean Average Precision
MOSSE	Minimum Output Sum of Squared Error
MOT	Multiple Object Tracking
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
NLP	Natural Language Processing
PSR	Peak-to-Sidelobe Ratio
SORT	Simple Online and Realtime Tracking
SSD	Single Shot Detector
SSE	Sum of Squared Error
SVM	Support Vector Machine
ViT	Vision Transformers
VOT	Visual Object Tracking
YOLO	You Only Look Once

LISTA DE SÍMBOLOS

$\sum$	Operador somatório
$\in$	Pertinência a um conjunto
σ	Desvio padrão
μ	Média aritmética
∞	Infinito
π	Constante Pi (aproximadamente 3,14159)
e^x	Função exponencial de x
$\mathcal{F}$	Transformada de Fourier
$*$	Operador de convolução
$\star$	Operador de correlação
H	Filtro de correlação (Kernel) no domínio da frequência
Z	Imagem de consulta no domínio da frequência
X	Imagem da galeria no domínio da frequência
G	Mapa de resposta esperado da correlação (pico gaussiano)

SUMÁRIO

1 INTRODUÇÃO

Em cidades inteligentes, a sinergia entre tecnologias avançadas e *Big Data* está revolucionando a gestão urbana. Para enfrentar os desafios impostos pela urbanização, Bollier (1998) sugeriu uma rede abrangente e a aplicação de tecnologias da informação, proposta que foi adotada pela empresa *International Business Machines Corporation* (IBM) em sua visão de Planeta Inteligente em 2008, segundo Liao e Chen (2022). Atualmente, o conceito de cidade inteligente se consolidou como uma estratégia fundamental para o desenvolvimento urbano, incorporando inteligência artificial, plataformas móveis, telecomunicações e outras inovações tecnológicas.

Os dados coletados por meio de diversas estratégias tecnológicas, que incluem sensoriamento remoto e sistemas de Internet das Coisas (IoT), são utilizados para gerenciar recursos, ativos e serviços de forma eficiente, otimizando as operações em toda a cidade. Em 2022, algumas áreas de maior tendência para a formação de cidades inteligentes foram identificadas, com a mobilidade inteligente liderando, seguida pela participação do cidadão por meio de plataformas digitais, segurança pública e gestão de energia inteligente (Pooja G. et al., 2022).

No planejamento para a transformação das cidades em ambientes inteligentes, os sistemas de transporte inteligente (ITS) se destacam como elementos essenciais. Esses sistemas combinam tecnologias avançadas e soluções de engenharia para otimizar o tráfego de veículos e pedestres, com o objetivo de melhorar a segurança, reduzir a emissão de poluentes e promover a mobilidade urbana. A pesquisa nessa área tem avançado rapidamente, impulsionada pela disponibilidade de ferramentas de computação em nuvem que permitem o processamento, análise e visualização de grandes volumes de dados. Isso viabiliza a utilização de informações que antes eram consideradas impraticáveis, ampliando significativamente o potencial para a gestão e otimização do transporte urbano (MONTOYA-TORRES et al., 2021).

Dentro das tecnologias de sensoriamento em cidades inteligentes, há uma crescente tendência em se utilizar câmeras de monitoramento para criar um sistema de vigilância abrangente (KOSTIC et al., 2022). Isso permite o monitoramento simultâneo de diversas áreas, oferecendo uma visão geral do comportamento urbano e facilitando a tomada de decisões rápidas para resolver problemas emergentes.

Para coletar dados de trânsito a partir das múltiplas câmeras que compõem o sistema de monitoramento, é necessário rastrear os elementos que formam o fluxo de tráfego nas cenas,

que podem incluir ruas, avenidas e praças. Sistemas inteligentes baseados em câmeras, através da visão computacional, têm a capacidade de fornecer informações relevantes e automatizar processos que seriam executados operacionalmente com auxílio da visão humana. No entanto, o desenvolvimento de aplicações em tempo real, que forneçam dados atualizados continuamente, enfrenta várias limitações e desafios (OLADIMEJI et al., 2023).

A maioria das cidades ainda não possui a infraestrutura adequada para lidar com o grande volume de dados gerados, resultando em redes de comunicação ineficientes que podem criar obstáculos no processamento em tempo real, especialmente para vídeos de câmeras remotas (CHEN; CHEN, 2018). Problemas como ruídos, perda de informações e travamentos podem ocorrer durante a transferência de *frames* dos vídeos, seja por saturação da rede ou falhas de comunicação.

Entre os principais desafios na área de visão computacional se encontram a detecção, rastreamento e reidentificação de múltiplos objetos. As dificuldades mencionadas, associadas à transmissão de vídeo, demandam o uso de abordagens robustas, leves e consistentes, para manter a continuidade e precisão na identificação dos objetos rastreados, evitando troca ou perda de identidade (LUO et al., 2021b).

A reidentificação de objetos é um campo da visão computacional dedicado a resolver problemas relacionados à identidade de objetos rastreados. Além de travamentos de vídeo, outros desafios como oclusão e objetos similares podem criar identidades duplicadas ou alterar a identidade dos objetos. Manter a identidade dos objetos rastreados é crucial para obter informações contextuais, como monitoramento do fluxo de tráfego e extração de métricas ou determinação da rota completa percorrida por veículos ou pessoas na cidade (ZAKRIA et al., 2021).

Entre as abordagens possíveis, Bolme et al. (2010) destacam a utilização de filtros de correlação discriminativos, tradicionalmente empregados em tarefas de rastreamento visual, mas com potencial subexplorado para a reidentificação de objetos com baixo custo computacional.

Embora metodologias que utilizam modelos de reidentificação baseados em redes neurais tenham avançado na melhoria da precisão, o custo computacional associado a essas abordagens ainda é elevado, tornando o processamento em tempo real desafiador e exigindo hardware de alta capacidade (THOMPSON et al., 2022). Isso pode aumentar significativamente o custo final de implantação e ainda assim não garantir alta confiabilidade no rastreamento de múltiplos objetos.

Portanto, a busca por avanços contínuos em ITS envolve não apenas o uso de tecnologias modernas e precisas, mas também a preocupação com sua eficiência e viabilidade de

implantação prática. Isso permite uma maior adesão, independentemente do nível de desenvolvimento das cidades.

Assim, o trabalho aqui proposto se concentra em estudar, testar e analisar uma abordagem alternativa para a reidentificação de objetos, combinada ao rastreamento múltiplo, utilizando filtros de correlação discriminativos em vídeos de câmeras de monitoramento. O objetivo é oferecer uma identificação precisa dos elementos rastreados a um custo computacional menor, em comparação com métodos baseados em redes neurais convolucionais (CNNs).

1.1 Motivação e Justificativa

Quando se fala em cidades inteligentes, há de se achar um equilíbrio para que as tecnologias tragam evolução, mas também confiabilidade, tudo isso a um custo acessível para os órgãos responsáveis pelos investimentos. Portanto, o estudo de técnicas que possam atingir um resultado satisfatório a um custo computacional menor se torna atrativo.

Filtros de correlação já foram utilizados no passado para realizar tarefas que geralmente envolvem o rastreamento de objetos, mas não é comum ver sua aplicação especificamente definida para reidentificação. Exemplo disso é o uso de filtro de correlação discriminativo como metodologia para rastreamento de múltiplos objetos proposto por Henriques et al. (2015). Filtros de correlação geralmente se destacam pelo equilíbrio entre precisão e velocidade de processamento, através de estratégias como processamento de imagem no domínio da frequência, que podem trazer benefícios computacionais. Outro ponto positivo é a sua lógica de treinamento *online*, ou seja, em tempo de processamento, não exigindo nenhum treinamento exaustivo prévio, como os modelos de redes neurais necessitam para treinamento supervisionado.

O filtro de correlação é uma técnica já conhecida na visão computacional para o Rastreamento Visual de Objetos (VOT - Visual Object Tracking), que é o rastreamento de objetos pelas características visuais apenas, mas há uma escassez de trabalhos que analisem essa abordagem como uma tarefa de reidentificação isolada do rastreamento de objetos, ou seja, utilizar o filtro de correlação apenas para identificar a similaridade entre imagens de forma generalizada.

Dentro desse contexto, este trabalho procura trazer uma contribuição analítica para os desafios de visão computacional relacionados à reidentificação e rastreamento de múltiplos objetos em tempo real.

Desta forma, espera-se, após testes e análises, que seja possível determinar se a utilização da técnica de filtro de correlação discriminativo como solução para reidentificação de

objetos é viável, frente aos desafios já mencionados, comparando também o desempenho com modelos de redes neurais profundas.

1.2 Metodologia Proposta

Neste trabalho será desenvolvida uma metodologia para a utilização de técnicas já existentes, tanto para reidentificação de objetos quanto para rastreamento de múltiplos objetos, incluindo pequenas modificações que permitam a integração entre ambas as tarefas de visão computacional.

Primeiramente, será estudada e adaptada uma das abordagens de filtro de correlação para ser utilizada no processo de reidentificação de forma isolada, ou seja, apenas cumprindo a tarefa de medir a similaridade entre imagens de interesse, extraídas de *datasets* voltados para a reidentificação de pessoas e veículos, ajustando-a de forma que seja possível comparar com abordagens que utilizam modelos baseados em redes neurais.

Depois desta primeira etapa, será feita a escolha do algoritmo de rastreamento de múltiplos objetos, buscando os melhores resultados conforme métricas de rastreio, lembrando da necessidade de processamento em tempo real. A partir da abordagem de rastreamento selecionada, deve-se elaborar um módulo capaz de ser acoplado ao algoritmo escolhido, para estender suas capacidades e melhorar a consistência da identificação dos objetos através da reidentificação.

1.3 Problemas e Desafios

A reidentificação de objetos em vídeos é um desafio complexo devido às dificuldades em definir similaridade e dissimilaridade entre objetos em imagens digitais (ZHOU et al., 2019). A busca por um equilíbrio entre a precisão de reidentificação e baixo custo computacional é fundamental para o sucesso dessa tarefa.

Os benefícios de uma reidentificação consistente podem elevar o nível dos analíticos de vídeo no contexto de cidades inteligentes, melhorando o desempenho e a confiabilidade do rastreamento de múltiplos objetos e em múltiplas câmeras.

No entanto, a disponibilidade de *datasets* que representem adequadamente o cenário de câmeras de monitoramento, sem aplicar nenhum viés através das imagens, é escassa. Esses *datasets* podem conter imagens muito genéricas ou de qualidade superior à encontrada no cenário real. Portanto, os problemas encontrados em situações reais podem não ser devidamente reproduzidos por esses *datasets*. O mesmo é válido para conjuntos de vídeos que possuem boa resolução, qualidade de imagem e sem travamentos, o que foge de muitas

situações comuns de câmeras de monitoramento reaproveitadas para aplicações em visão computacional, que geralmente não fornecem vídeos de alta qualidade e estabilidade.

Um dos pontos que torna os filtros de correlação interessantes, apesar de complexos, é a estratégia de processamento no domínio da frequência. Embora essa abordagem otimize o cálculo denso de correlação, ela também dificulta o entendimento e o controle do processo realizado, sendo mais complicado visualizar as manipulações ocorridas durante esses cálculos, diferentemente do processamento de imagens no domínio do espaço (SOLOMON; BRECKON, 2011).

Há também o desafio de encontrar métricas de similaridade robustas para as respostas do filtro de correlação, ou seja, computar o valor de similaridade através da imagem-resposta da correlação.

Como já mencionado, o foco da pesquisa está na utilização de filtros de correlação com rastreamento de múltiplos objetos em tempo real para câmeras de monitoramento. Isso traz dificuldades inerentes à movimentação dos objetos na cena durante o rastreamento, como oclusão, mudança de posição, mudança no formato e direção.

1.4 Delimitação de Escopo

O escopo deste trabalho é definido pela busca por uma solução que forneça equilíbrio entre precisão e velocidade de processamento, para reidentificação e rastreamento de objetos em tempo real. Dessa forma, não se busca superar os níveis de precisão obtidos por modelos baseados em redes neurais, nem atingir a máxima eficiência computacional, mas sim propor uma abordagem intermediária, com desempenho satisfatório e viabilidade prática.

A estratégia a ser adotada com o uso de filtro de correlação para resolver o problema de reidentificação deve-se limitar a algoritmos conhecidos na literatura. A intenção não é elaborar novas abordagens dentro do assunto filtros de correlação, mas explorar as suas vantagens para encontrar uma nova solução que possa trazer um bom compromisso entre precisão e velocidade em problemas de reidentificação, tornando assim mais objetivo o desenvolvimento do trabalho proposto.

Para a experimentação, serão utilizados *datasets* amplamente reconhecidos no campo da visão computacional, compostos por imagens e vídeos específicos para reidentificação e rastreamento de múltiplos objetos, focando principalmente em pessoas e veículos, que são o foco principal do videomonitoramento pretendido.

Os algoritmos a serem adotados devem estar alinhados com os objetivos propostos, seguindo o mesmo raciocínio lógico na escolha das opções existentes. O mesmo é válido para as

métricas de similaridade utilizadas no processo de reidentificação entre imagens.

Durante os experimentos com os *datasets* reunidos, para reidentificação de objetos, serão utilizadas amostras de imagens contendo os objetos de interesse. Dependendo do *dataset*, poderá haver avaliação em múltiplas câmeras. No entanto, no caso do rastreamento de múltiplos objetos, os testes ocorrerão em sequências de imagens que compõem um vídeo de apenas uma câmera por vez, sem associação entre câmeras.

1.5 Objetivos

Com base na introdução sobre a situação problema e a proposta de trabalho, os objetivos a seguir são formalizados para clarificar o caminho a ser adotado nesta dissertação.

1.5.1 Objetivo Geral

Desenvolver, testar e analisar um filtro de correlação como módulo de reidentificação de objetos, integrado ao rastreamento de múltiplos objetos, com foco em câmeras de monitoramento e processamento em tempo real.

1.5.2 Objetivos Específicos

A seguir, estão listados os objetivos específicos que visam alcançar o objetivo geral proposto, abrangendo as etapas de reidentificação e rastreamento de múltiplos objetos:

1. Avaliar a viabilidade do filtro de correlação como uma solução equilibrada para a reidentificação de objetos.
2. Avaliar a integração do filtro de correlação ao rastreamento de múltiplos objetos, visando estender a capacidade de manter uma identificação consistente.
3. Comparar a solução desenvolvida com outras existentes baseadas em redes neurais.

1.6 Estrutura da Dissertação

O Capítulo 1 oferece uma introdução ao contexto do problema e à solução proposta, abordando também os principais desafios e objetivos do trabalho.

O Capítulo 2 fornece uma revisão dos assuntos e trabalhos relacionados à proposta, oferecendo o embasamento teórico essencial para o entendimento dos capítulos subsequentes.

Para atingir os objetivos especificados e abordar de forma detalhada as tarefas de reidentificação e rastreamento de objetos, os capítulos de desenvolvimento foram organizados conforme descrito na sequência.

No Capítulo 3, é apresentada a experimentação focada na reidentificação como uma tarefa isolada. Este capítulo visa avaliar a capacidade de medir a similaridade entre objetos em conjuntos de imagens de pessoas e veículos, utilizando o processamento com filtros de correlação.

O Capítulo 4 trata da etapa complementar, na qual a abordagem de reidentificação com filtro de correlação é integrada ao algoritmo de rastreamento de múltiplos objetos escolhido. Este capítulo compara a metodologia proposta com abordagens que utilizam modelos de redes neurais profundas, previamente treinados para imagens de pessoas e veículos.

Por fim, o Capítulo 5 apresenta uma análise dos resultados obtidos na experimentação com reidentificação e rastreamento de múltiplos objetos. Este capítulo discute as particularidades de cada abordagem e sua aplicabilidade.

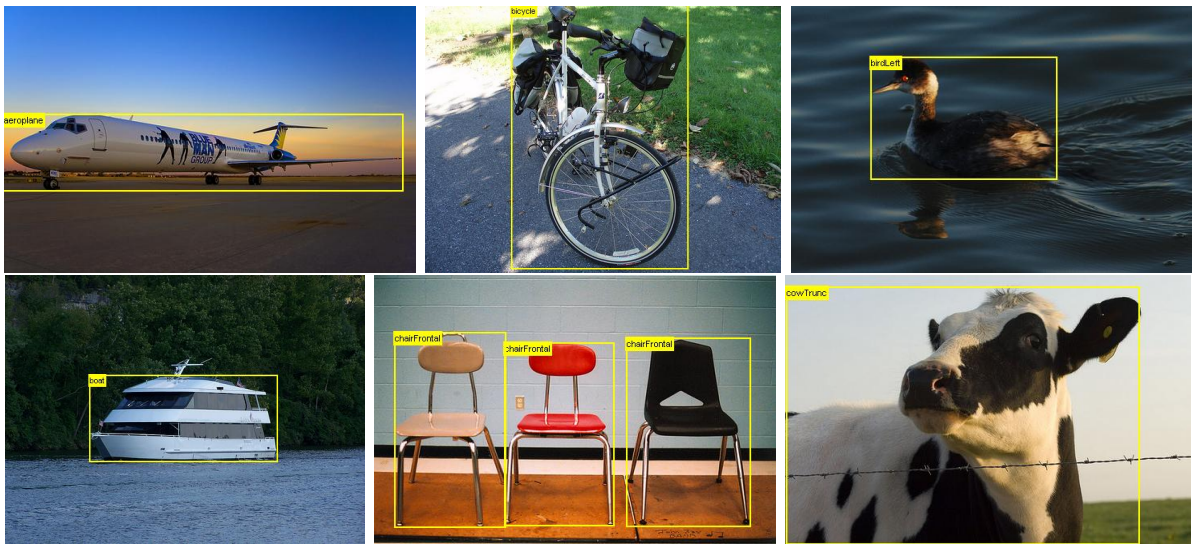
2 FUNDAMENTAÇÃO TEÓRICA

Através deste capítulo de embasamento teórico, será possível obter os principais conceitos sobre os tópicos mais relevantes em visão computacional e demais áreas de estudo envolvidas neste trabalho. Seguindo essa proposta, o conteúdo descreve primeiramente os conceitos de três grandes desafios dentro da visão computacional, que são a detecção, rastreamento e reidentificação de objetos, passando também um pouco do histórico da evolução das abordagens propostas. Após isso, o filtro de correlação discriminativo deve ser abordado de forma a se ter o conhecimento essencial para compreensão do seu funcionamento e aplicação. Por fim, alguns trabalhos envolvendo os temas reidentificação, rastreamento de múltiplos objetos e filtros de correlação serão citados para situar a teoria mencionada nos cenários de atuação.

2.1 Detecção de Objetos

A detecção de objetos em visão computacional é um campo da inteligência artificial e processamento de imagens que visa identificar e localizar objetos específicos em imagens ou vídeos. Seu propósito é reconhecer automaticamente padrões visuais e diferenciar entre diversas classes de objetos, como carros, pessoas e animais, exemplificado na Figura 1.

Figura 1 – Detecção de objetos no *dataset Pascal VOC*.



Fonte: *Visual Object Classes Challenge* 2012 (VOC2012), disponível em http://host.robots.ox.ac.uk/pascal/VOC/voc2012/.

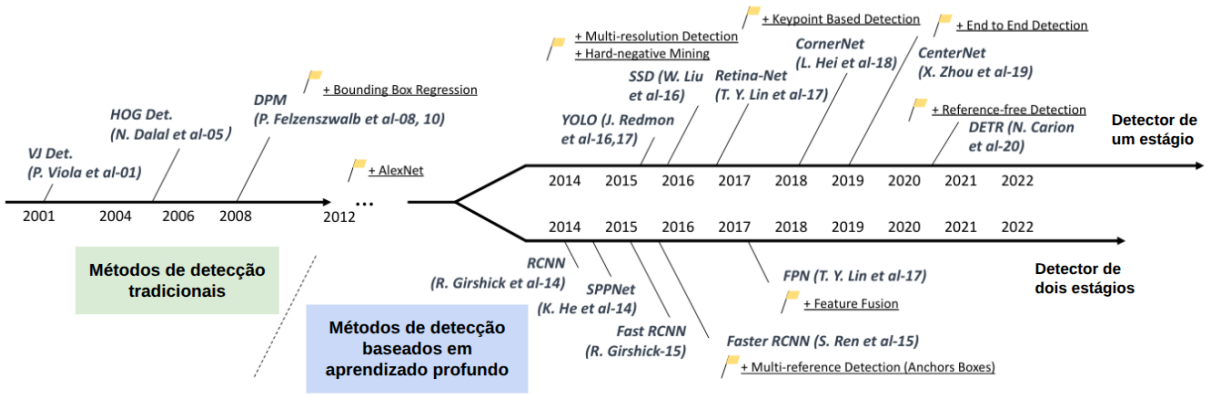
Algumas abordagens clássicas foram amplamente utilizadas ao longo dos anos, como o algoritmo *Viola-Jones*, proposto em 2001 por Paul Viola e Michael Jones, baseado na

extração de características *Haar-Like* e classificador *Adaboost*, onde seu principal destaque é o processamento em tempo real após seu modelo treinado, sendo muito utilizado no reconhecimento facial para ajuste de foco automático em câmeras digitais (VIOLA; JONES, 2001). Da mesma forma, *Support Vector Machines* (SVMs) surgiram para facilitar o processo de treinamento e proporcionar resultados mais precisos, também realizado por meio de treinamento supervisionado, utilizando, por exemplo, as características de cores e bordas através de Histogramas de Gradientes Orientados (HOG) para detectar os padrões esperados em diferentes classes de objetos, conforme a solução proposta por Dalal e Triggs (2005).

Embora o algoritmo Viola-Jones (VIOLA; JONES, 2001) e as SVMs (CORTES; VAPNIK, 1995) tenham sido amplamente utilizados com sucesso em muitas aplicações de detecção de objetos, eles tendem a ter desempenho inferior em comparação às abordagens baseadas em redes neurais convolucionais modernas. As CNNs geralmente alcançam melhores resultados em termos de precisão e robustez, especialmente em cenários desafiadores com variações de iluminação, oclusão e objetos de diferentes tamanhos. No entanto, as abordagens clássicas ainda podem ser úteis em cenários específicos onde os recursos computacionais são limitados ou quando se requer desempenho em tempo real com recursos mais modestos (ZAIDI et al., 2021; ZOU et al., 2023).

Após o trabalho proposto por Krizhevsky, Sutskever e Hinton (2012), as redes neurais se tornaram o padrão adotado para resolver problemas de detecção de objetos, atingindo o estado da arte para esse desafio. Uma boa representação dos marcos históricos dos trabalhos propostos nessa área pode ser vista na Figura 2, feita por Zou et al. (2023).

Figura 2 – Marcos da detecção de objetos.



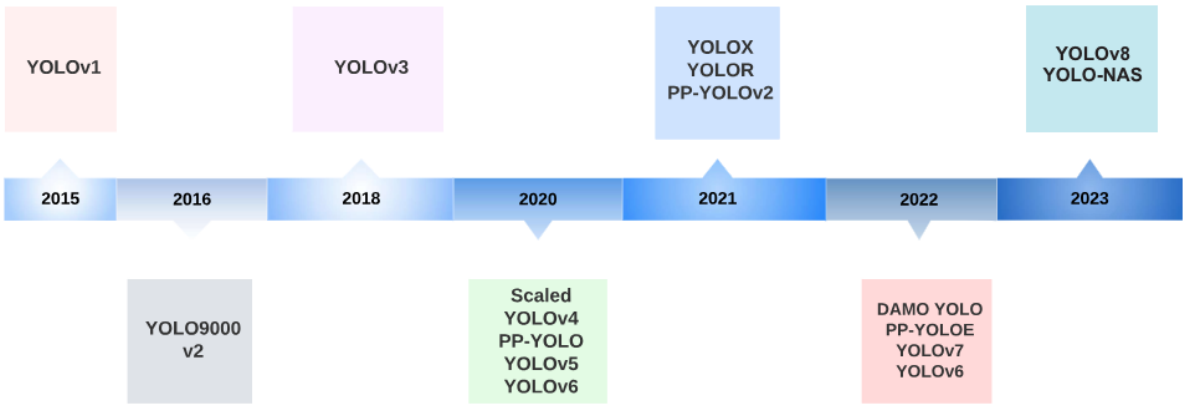
Fonte: Adaptado de Zou et al. (2023).

O campo das redes neurais convolucionais, aplicadas às soluções de visão computacional, ganhou destaque especialmente com o desenvolvimento de arquiteturas como a *Faster R-CNN*, *You Only Look Once* (YOLO) e *Single Shot Detector* (SSD) (REN et al., 2016;

LIU et al., 2016; REDMON et al., 2016). Essas redes são treinadas em grandes conjuntos de dados rotulados para aprender a detectar objetos com precisão e eficiência. Nessas abordagens, cada objeto detectado recebe uma pontuação de confiança (*confidence score*), que indica a probabilidade de a predição estar correta. Essa pontuação é usada para filtrar detecções com baixa confiança, reduzindo falsos positivos e melhorando a precisão geral do sistema.

No que se refere à arquitetura YOLO, seu principal diferencial em relação a outras abordagens de detecção de objetos é a capacidade de realizar predições em tempo real por meio de uma única passagem pela rede neural, o que a torna extremamente rápida e eficiente. Por essa razão, segundo Terven, Cordova-Esparza e Romero-Gonzalez (2023), a YOLO obteve grande destaque na área e originou diversas versões com melhorias significativas ao longo do tempo. Essa evolução é ilustrada na Figura 3, que apresenta o histórico das principais versões até 2023, sendo a YOLOv12 (TIAN; YE; DOERMANN, 2025) a proposta mais recente.

Figura 3 – Linha do tempo das diferentes versões propostas da arquitetura YOLO.



Fonte: Terven, Cordova-Esparza e Romero-Gonzalez (2023).

A capacidade de identificar e localizar objetos automaticamente em imagens e vídeos desempenha um papel fundamental em muitas tecnologias emergentes, contribuindo para avanços significativos em áreas como segurança, saúde, transporte e entretenimento. Com isso, uma ampla gama de aplicações práticas se tornou possível graças a essas técnicas propostas, incluindo vigilância por vídeo, veículos autônomos, sistemas de assistência ao motorista, reconhecimento facial, classificação de imagens e muito mais.

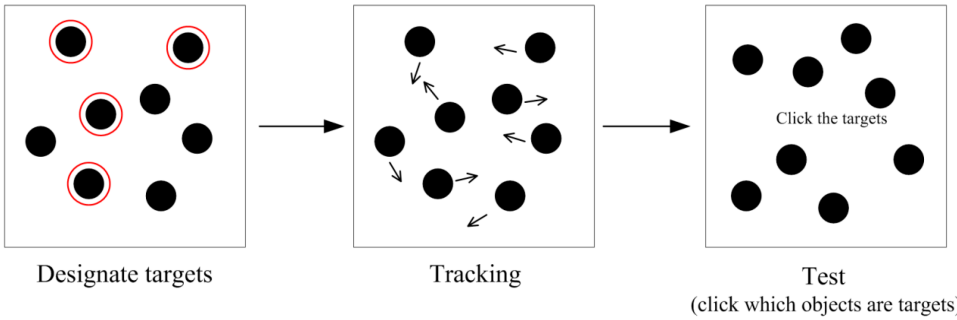
É importante destacar que, apesar de detectarem a presença e a localização dos objetos, os algoritmos de detecção não atribuem uma identidade persistente aos objetos ao longo do tempo. Ou seja, em uma sequência de vídeo, uma mesma pessoa pode ser detectada em quadros consecutivos, mas cada detecção é tratada de forma independente. Por essa

razão, técnicas de rastreamento (*tracking*) são frequentemente utilizadas em conjunto com a detecção para manter a identidade dos objetos entre quadros, associando detecções ao longo do tempo com base em critérios espaciais, temporais e visuais (LUO et al., 2021b).

2.2 Rastreamento de Múltiplos Objetos

A percepção visual humana sobre o rastreamento de objetos em movimento é um tema central em estudos que antecedem certos conceitos explorados na visão computacional. Em 1989, Pylyshyn propôs a teoria do rastreamento de objetos visuais, argumentando que este é um processo essencial na percepção visual humana. Ele sustentou a ideia de que tal processo opera de maneira contínua e automática, visando manter o controle do movimento ao longo do tempo sem perder a identidade de cada objeto, como demonstra a Figura 4, onde os objetos possuem a mesma característica visual e a identidade deve ser mantida baseada apenas no movimento e percepção humana (PYLYSHYN; STORM, 1988). Este trabalho influenciou significativamente a compreensão da percepção visual e serviu como base para muitas pesquisas subsequentes sobre o rastreamento de objetos.

Figura 4 – Experimento realizado por Pylyshyn e Storm (1988) sobre a percepção visual humana para o rastreamento de múltiplos objetos.



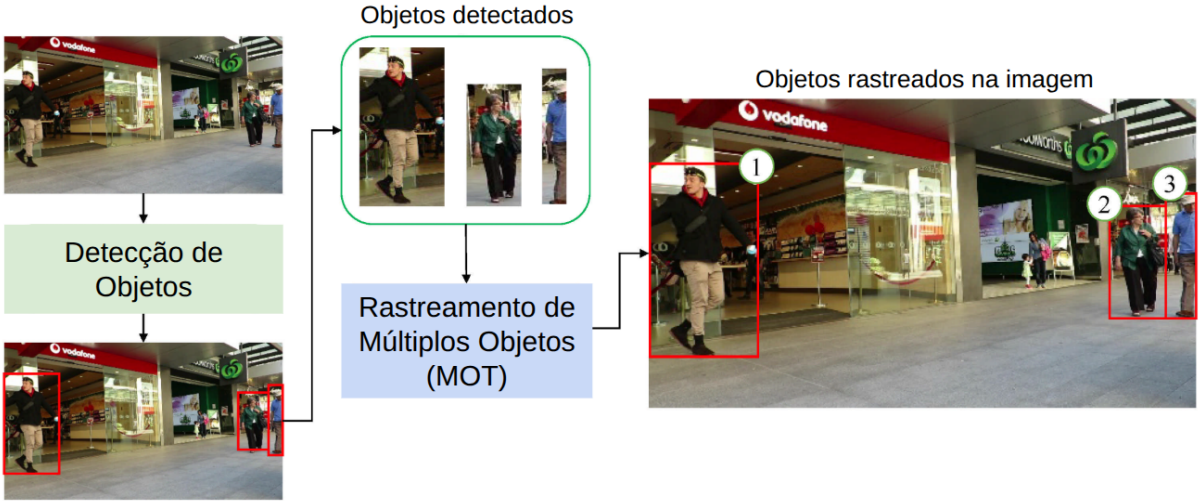
Fonte: Hyona, Li e Oksama (2019).

Ao longo das décadas seguintes, ocorreram avanços consideráveis no campo do rastreamento de objetos em visão computacional, impulsionados pelo crescente interesse em aplicações práticas, como vigilância por vídeo, monitoramento de tráfego, realidade aumentada, entre outros. Em visão computacional, o rastreamento de múltiplos objetos (MOT) visa rastrear e identificar vários objetos simultaneamente em cenas complexas, permitindo uma compreensão mais aprofundada de como esses objetos interagem entre si e com o ambiente ao longo do tempo (LUO et al., 2021b).

Nesse contexto, destaca-se o paradigma de rastreamento por detecção (tracking-by-detection), como ilustrado na Figura 5, no qual objetos são primeiro detectados de forma independente em cada quadro do vídeo, por um algoritmo de detecção de objetos,

e posteriormente esses objetos são associados ao longo do tempo por um algoritmo de rastreamento (PARK et al., 2021).

Figura 5 – Rastreamento de múltiplos objetos conforme o paradigma *tracking-by-detection*.



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

Um marco significativo foi o desenvolvimento de métodos de rastreamento baseados em modelos, como o filtro de Kalman (KALMAN, 1960) e o filtro de partículas (GORDON; SALMOND; SMITH, 1993). Estes métodos combinam informações de modelos de movimento e observações visuais para estimar a posição e trajetória de objetos ao longo do tempo, prevendo o estado futuro do objeto e corrigindo-o com base em novas medições. Embora eficazes em muitos cenários, esses métodos frequentemente enfrentam desafios em ambientes complexos para rastreamento de múltiplos objetos, sendo o filtro de Kalman indicado para cenários mais comuns de videomonitoramento e, quando há muitos objetos na cena, o filtro de partículas pode obter melhores resultados (MARRON et al., 2007).

O rastreamento de objetos em visão computacional evoluiu ao longo dos anos com uma variedade de abordagens, cada uma com suas próprias vantagens e limitações. Inicialmente, destacam-se o Mean Shift e o CAM Shift, que buscam encontrar o centro de massa para o conjunto de pixels da referência alvo a cada imagem nova. Embora eficazes em rastrear objetos com padrões de cor distintos, esses métodos podem ser sensíveis a mudanças na iluminação e oclusões (BRADSKI, 1998; COMANICIU; MEER, 2002).

Outra técnica fundamental é o fluxo óptico, que calcula o movimento dos pixels entre frames consecutivos. Este pode ser implementado usando *keypoints* ou com uma abordagem *grid-based* (LUCAS; KANADE, 1981; HORN; SCHUNCK, 1981). Embora útil para capturar o movimento global dos objetos, o fluxo óptico pode ser impreciso em cenas complexas com movimentos rápidos ou oclusões.

Em 2016, foi introduzido o *Simple Online and Realtime Tracking* (SORT) utilizando apenas recursos espaciais, como Interseção pela União (IoU) e filtro de Kalman, seguido pelo DeepSORT em 2017, que combina técnicas de rastreamento com uma rede neural para melhorar a identificação dos objetos (BEWLEY et al., 2016; WOJKE; BEWLEY; PAULUS, 2017). Essas abordagens evidenciam a evolução do rastreamento de múltiplos objetos, combinando técnicas clássicas com métodos modernos baseados em redes neurais profundas, um movimento semelhante ao observado na área de detecção de objetos (LUO et al., 2021b; PARK et al., 2021; WANG; SONG; HWANG, 2024).

Métodos como o *Joint Detection and Embedding* (JDE), Tracktor/Tracktor++ e posteriormente FairMOT, integraram o processo de detecção e reidentificação de objetos para o rastreamento em uma estrutura unificada, como mencionam Wang, Song e Hwang (2024). Tais métodos adicionam uma camada à rede neural com foco em aprender as diferenças entre os objetos para manter a identificação correta, reaproveitando a extração de características já realizada pelo detector de objetos e, assim, otimizando o processamento como um todo.

Seguindo uma estratégia simples, similar ao funcionamento do SORT, porém com uma visão diferente em relação ao aproveitamento das detecções fornecidas pelo algoritmo de detecção de objetos, Zhang et al. (2022) introduziram o algoritmo de rastreio chamado ByteTrack, que vem se equiparando ou até superando algoritmos complexos com resultados positivos, a um custo computacional reduzido, se comparado a metodologias que utilizam redes neurais. Mais informações sobre o ByteTrack serão descritas a seguir, mostrando o porquê dessa abordagem ser relevante para o trabalho aqui proposto.

Essas são apenas algumas das muitas abordagens que têm sido desenvolvidas ao longo dos anos para o rastreamento de múltiplos objetos em visão computacional, demonstrando um progresso contínuo em direção a sistemas mais precisos, eficientes e robustos.

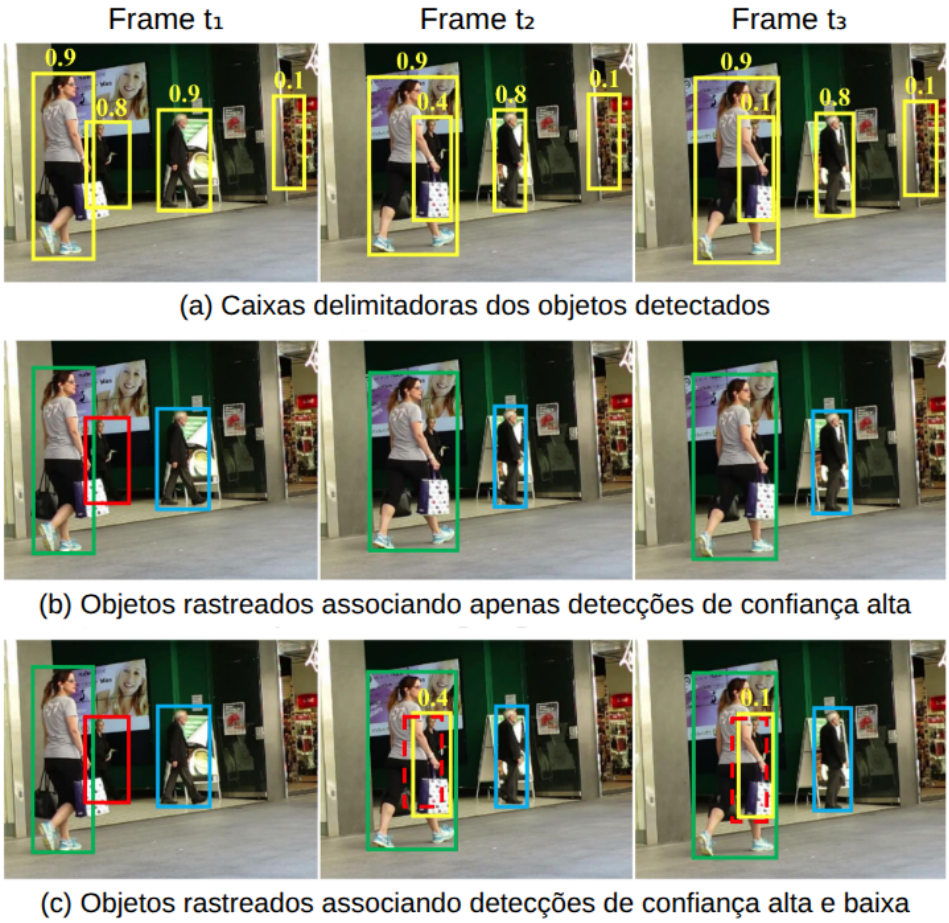
2.2.1 ByteTrack

A maioria das abordagens de rastreamento de múltiplos objetos segue uma estrutura lógica similar, baseada no rastreio-por-detecção, onde recebem os objetos de interesse detectados contendo as informações de confiança de cada detecção. O valor de confiança é geralmente utilizado para eliminar incertezas na identificação dos objetos, ignorando todas as amostras de confiança baixa antes de passar no algoritmo de rastreio. Porém, dentro desses casos de incerteza, muitos objetos detectados com confiança baixa representam verdadeiros positivos em condições mais complexas, como oclusão (ZHANG et al., 2022).

Zhang et al. (2022) enxergam uma perspectiva diferente para propor uma solução simples e eficaz que melhora os resultados tanto na acurácia de rastreamento quanto na identificação

dos objetos. Ao invés de descartar detecções de confiança baixa, aproveita praticamente todas as detecções fornecidas e destina os objetos de confiança alta para o fluxo de rastreio como de costume e as detecções de confiança baixa para a manutenção da identificação dos objetos que já estão sendo rastreados, como demonstra a Figura 6.

Figura 6 – Comparação do rastreamento considerando apenas detecções de confiança alta (b) e todas detecções, de confiança alta e baixa (c).



Fonte: Adaptado de Zhang et al. (2022)

A abordagem segue a base do algoritmo SORT proposto por Bewley et al. (2016), justamente pela simplicidade e eficiência computacional, porém possibilitando não apenas a utilização de similaridades provindas de informação espacial, como o cálculo de IoU, mas também através de métodos de reidentificação, na utilização de modelos que forneçam a similaridade visual entre os objetos, como proposto no DeepSORT por Wojke, Bewley e Paulus (2017). Segundo Zhang et al. (2022), a associação após a matriz de similaridade gerada também pode seguir os cálculos do método Húngaro (KUHN, 1955), designando as detecções para atualizar os objetos já rastreados.

Zhang et al. (2022) também implementam a solução proposta integrada a outras abordagens de rastreamento existentes, melhorando o resultado atingido de cada caso. Por exemplo,

aplicado ao FairMOT sobre o *dataset* MOT17 de validação, sem a associação *BYTE*, atinge resultados de 69.1% para acurácia de rastreamento de múltiplos objetos (MOTA), 72.8% na precisão de identificação (IDF1) e 299 trocas de identificação (IDSW). Já com a associação BYTE para aproveitar todas as detecções, os resultados melhoram para 70.4% e 74.2% respectivamente, assim como as trocas de identificação caem para 232.

Portanto, o ByteTrack é uma abordagem que atingiu o estado da arte em rastreamento de múltiplos objetos de forma simples, apenas aproveitando recursos de detecção que já estavam presentes, mas sendo ignorados. Pela sua simplicidade, desempenho computacional e flexibilidade de implementação, será uma abordagem de rastreamento essencial no desenvolvimento e estudo do trabalho proposto nesta dissertação.

2.2.2 Métricas de Desempenho para Rastreamento Múltiplo

Para avaliar o desempenho do processo de rastreamento de múltiplos objetos, várias métricas podem ser computadas, porém as mais relevantes serão descritas a seguir, onde a resposta fornecida deve indicar o quão consistente e confiável está sendo o rastreio, através da correta identificação dos objetos e suas localizações.

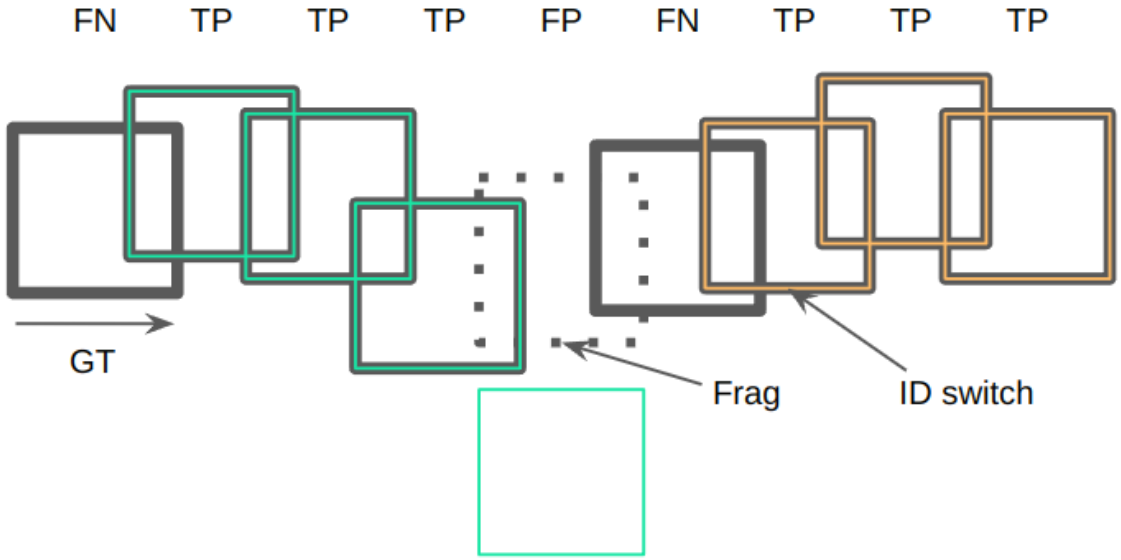
2.2.2.1 CLEAR: MOTA e MOTP

Na dificuldade em determinar formas de avaliar o rastreamento de múltiplos objetos, Bernardin e Stiefelhagen (2008) se basearam em *workshops* internacionais voltados para avaliação de diversos desafios relacionados ao rastreamento, chamados de *Classification of Events, Activities, and Relationships* (CLEAR), ocorridos nos anos de 2006 e 2007. As principais métricas implementadas foram de *Multiple Object Tracking Accuracy* (MOTA) e *Multiple Object Tracking Precision* (MOTP), onde Bernardin e Stiefelhagen (2008) demonstram a relevância dessas métricas para estimar de forma precisa a localização dos objetos e a acurácia em reconhecer e rotular corretamente os objetos ao longo do tempo.

Antes de apresentar os cálculos das métricas MOTP e MOTA, é necessário compreender alguns conceitos fundamentais e outras métricas complementares que enriquecem a análise e contextualizam melhor os resultados. A metodologia adotada baseia-se na identificação de hipóteses de localização dos objetos e na quantificação dos erros atrelados às associações incorretas entre as trajetórias estimadas e as verdadeiras. Como demonstra a Figura 7, quando a estimativa de rastreio acerta na hipótese de localização, ela pode ser considerada um verdadeiro positivo, caso contrário, será um falso positivo. Já quando perde o rastreamento do objeto existente, a estimativa é considerada um falso negativo. Além disso, em relação à identificação do objeto, surgem duas métricas relevantes, a fragmentação ocorrida durante os frames em que o objeto fica sem rastreio após já ter sido iniciado (*Frag*) e a quantidade de trocas de identificação (*ID Switches*, IDSW) que ocorrem quando

uma nova identificação é criada para o mesmo objeto de forma errônea (MILAN et al., 2016; DENDORFER et al., 2020b).

Figura 7 – Problemas de associação entre as caixas delimitadoras (*bounding boxes*) de um objeto detectado e rastreado ao longo de 9 frames de exemplo. TP: *True Positive* (Verdadeiro Positivo), FP: *False Positive* (Falso Positivo) e FN: *False Negative* (Falso Negativo).



Fonte: Próprio autor.

MOTA mede a acurácia geral do rastreio múltiplo e pode ser calculada conforme a seguinte fórmula:

$$\text{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{2.1}$$

onde:

- FN_t é o número de falsos negativos no *frame* t;
- FP_t é o número de falsos positivos no *frame* t;
- $IDSW_t$ é o número de trocas de identidade no *frame* t;
- GT_t é o número de objetos verdadeiros no *frame* t;
- MOTA pode variar de $-\infty$ a 1, onde 1 indica uma acurácia perfeita.

Já MOTP mede a precisão de localização dos objetos rastreados em relação às suas verdadeiras posições ou verdade absoluta (*ground truth*), a fórmula para calcular MOTP é:

$$\text{MOTP} = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t} \tag{2.2}$$

onde:

- $d_{i,t}$ é a distância entre o objeto rastreado i e a verdade absoluta no *frame* t;
- c_t é o número de correspondências entre objetos rastreados e objetos verdadeiros no *frame* t.

MOTP varia de 0 a 1, onde valores mais próximos de 0 indicam maior precisão de localização.

A distância dos objetos pode ser calculada através da distância Euclidiana ou com IoU (1 - IoU) das caixas delimitadoras dos objetos.

2.2.2.2 Identificação: IDF1 Score

O IDF1 tem o foco em avaliar a preservação da identidade dos objetos rastreados ao longo da sequência de frames, considerando os acertos e erros ocorridos durante as associações para determinar uma relação entre a precisão e revocação (*recall*) de identificação (MILAN et al., 2016; DENDORFER et al., 2020b; RISTANI et al., 2016a).

A precisão de identificação (IDP) é a proporção de identificações corretas sobre todas as identificações feitas pelo algoritmo.

$$\text{IDP} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFP}} \tag{2.3}$$

onde:

- IDTP (*True Positives*) são as correspondências corretas entre as detecções e os objetos verdadeiros;
- IDFP (*False Positives*) são as detecções incorretamente associadas a qualquer identidade.

A revocação de identificação (IDR) é a proporção de identificações corretas sobre todas as identificações verdadeiras.

$$\text{IDR} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}} \tag{2.4}$$

onde:

- IDFN (*False Negatives*) são os objetos verdadeiros que não foram detectados ou corretamente associados pelo algoritmo.

Diferentemente das métricas pelo método CLEAR feitas *frame* a *frame*, o IDF1 acontece identidade por identidade (RISTANI et al., 2016a). O IDF1 pode ser encontrado conforme a fórmula abaixo:

$$\text{IDF1} = \frac{2 \times \text{IDTP}}{2 \times \text{IDTP} + \text{IDFP} + \text{IDFN}} \tag{2.5}$$

O IDF1 resulta em um valor de equilíbrio entre IDP e IDR. O IDP foca na qualidade das correspondências feitas de forma precisa, enquanto o IDR foca na capacidade do algoritmo de detectar e identificar corretamente com baixo índice de erros.

2.2.2.3 HOTA

Mais recentemente, construído com base na métrica geral do MOTA, porém com a proposta de lidar com as limitações da métrica anterior, o método *Higher Order Tracking Accuracy* (HOTA) é uma métrica que leva em consideração a precisão de diferentes ordens, ou seja, diferentes níveis de avaliação para as correspondências entre as trajetórias rastreadas e as trajetórias verdadeiras no longo prazo, capturando melhor a complexidade do rastreamento de múltiplos objetos e fornecendo submétricas para análises mais aprofundadas (LUITEN et al., 2020).

A HOTA se decompõe em algumas submétricas que avaliam diferentes aspectos do rastreamento, são elas:

- *Detection Accuracy*(DetA): Mede a acurácia da detecção de objetos;
- *Association Accuracy* (AssA): Avalia a capacidade do algoritmo de manter a identidade dos objetos ao longo do tempo.
- *Localization Accuracy* (LocA): Mede a acurácia da localização dos objetos detectados;

Para calcular a métrica HOTA, são necessários alguns cálculos de relação entre Verdadeiro Positivo (TP), Falso Positivo (FP) e Falso Negativo (FN), assim como feito para MOTA, obtidos a partir do processo de rastreamento. Também é necessário calcular a correspondência entre os objetos, podendo novamente utilizar a distância de interseção

sobre união (IoU) entre objetos rastreados e objetos reais. Para considerar um par (c), como uma correspondência válida, é necessário definir um limiar de IoU.

A métrica HOTA para um limiar de IoU definido pode ser obtida por:

$$\text{DetA} = \frac{|TP|}{|TP| + |FN| + |FP|} \tag{2.6}$$

$$\text{AssA} = \frac{\sum_{(c)\in TP} \frac{|TPA(c)|}{|TPA(c)|+|FNA(c)|+|FPA(c)|}}{|TP|} \tag{2.7}$$

$$\text{HOTA} = \sqrt{\text{DetA} \times \text{AssA}} \tag{2.8}$$

Onde TPA, FNA e FPA são cálculos adicionais, que podem ser consultados no trabalho de Luiten et al. (2020). Representam, respectivamente, as associações entre verdadeiros positivos, falsos negativos e falsos positivos.

Apesar de não estar diretamente incluída na equação da HOTA, a métrica *Localization Accuracy* (LocA) é discutida por Luiten et al. (2020) como um complemento essencial para a análise de desempenho, pois revela o quão precisamente os objetos foram localizados nas detecções corretas. Ela é calculada por.

$$\text{LocA} = \frac{\sum_{(c)\in|TP|} \text{IoU}(c)}{|TP|} \tag{2.9}$$

A métrica HOTA proporciona uma visão mais holística do desempenho de algoritmos de rastreamento de múltiplos objetos, considerando tanto a acurácia da detecção quanto a associação ao longo do tempo e localização dos objetos (LUITEN et al., 2020).

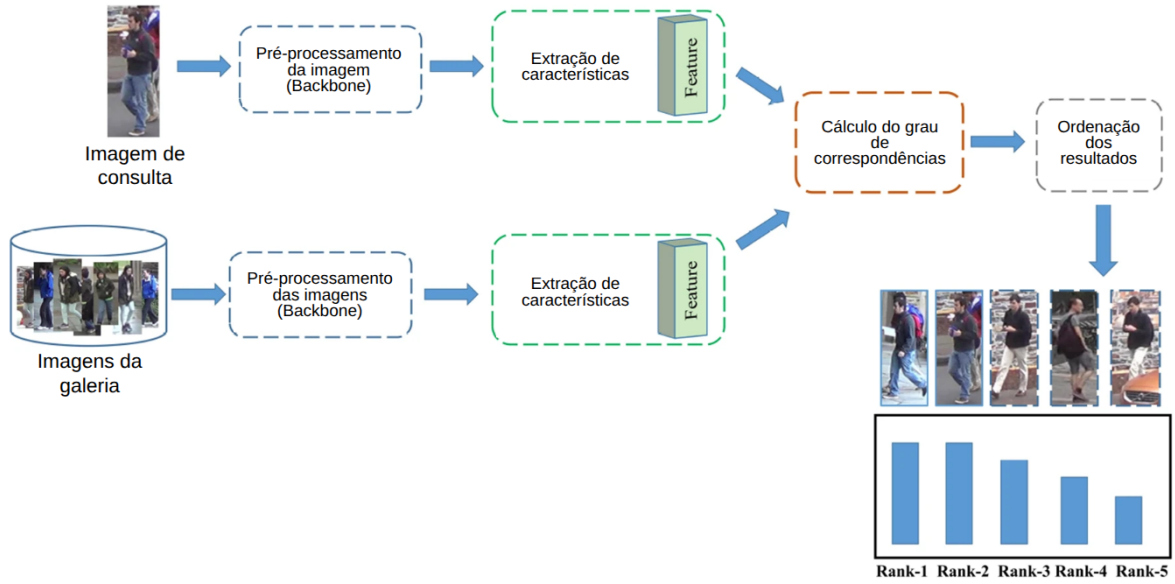
2.3 Reidentificação de Objetos

Soluções emergentes em centros urbanos baseadas em visão computacional vêm se tornando cada vez mais frequentes, auxiliando em tarefas como monitoramento de tráfego e segurança pública. Nesses casos, o desafio do reconhecimento correto dos elementos presentes na cena se torna essencial para a efetividade de um sistema inteligente. A reidentificação de objetos tem se tornado um tema presente em pesquisas recentes, buscando não apenas manter um bom rastreio em uma única câmera, mas também encontrar o elemento com a maior similaridade entre câmeras não sobrepostas, uma tarefa complexa à medida que as câmeras são posicionadas em diferentes ângulos ou localidades, gerando assim

diferentes perspectivas do mesmo objeto observado e rastreado ao longo de sua trajetória (ALMASAWA; ELREFAEI; MORIA, 2019; SUN et al., 2024; OLIVEIRA; MACHADO; TAVARES, 2021).

O processo de reidentificação normalmente se comporta como mostra a Figura 8. Uma das múltiplas imagens, referente aos objetos-alvo, é utilizada como ponto de prova para consulta (*query*) na base de imagens dos objetos previamente construída, também chamada de galeria (*gallery*). Através das características visuais discriminativas extraídas de cada imagem, conforme modelo ou abordagem adotada, as amostras são comparadas, buscando, através de métricas de similaridade, construir um *ranking* que indique a menor distância entre as características da imagem consulta e as imagens da galeria.

Figura 8 – Processo de reindentificação entre imagem de consulta e imagens da galeria, para formar o ranking de similaridade.



Fonte: Adaptado de Sun et al. (2024)

Siamese Network foi um dos principais trabalhos relacionados à tarefa de reidentificação, proposto por Bromley et al. (1993), cujo objetivo era realizar verificação de assinaturas pessoais através da comparação dos vetores de características extraídos por um par de redes neurais idênticas, porém com o dado de entrada diferente. Assim, através da medição de distância entre os resultados na saída, era computada a similaridade. Segundo Li, Chen e Zhang (2022), redes siamesas são adequadas para problemas de correspondência entre imagens, como recomendação de produtos.

Desde sua primeira proposta, inúmeros trabalhos surgiram com grandes avanços, estendendo as capacidades da rede siamesa simples, como por exemplo, Zheng et al. (2019), que utilizaram essa base para melhorar a localização espacial e invariância ao ponto de vista

para múltiplas câmeras, guiado pelo mecanismo de atenção, chamado *Consistent Attentive Siamese Network*.

Outra abordagem que surgiu para mudar o cenário dentro da visão computacional, foram *Vision Transformers* (ViTs), uma modificação dos *Transformers* que obtiveram sucesso em Processamento de Linguagem Natural (NLP), como menciona Ye et al. (2024). Essa é uma arquitetura de rede neural que dispensa convoluções e recorrências. Em vez disso, utiliza mecanismos de atenção para modelar as dependências globais entre entradas e saídas.

Já Zhou et al. (2019), propuseram uma arquitetura de rede para endereçar os problemas de escalas das características extraídas para reidentificação de pessoas, criando um bloco que acopla características homogêneas e heterogêneas, chamando o processo de *Omni-Scale feature learning* para compor a *Omni-Scale Network* (OSNet). Zhou et al. (2019) também exemplificaram uma das principais complexidades na tarefa de reidentificação, para relacionar corretamente imagens dentro de um conjunto que contenha, por exemplo, pessoas com aspecto visual muito parecido, como demonstra a Figura 9.

Figura 9 – Correspondências entre imagens no *dataset* Market1501 para a similaridade entre pessoas durante o processo de reindentificação. Para cada grupo de imagens, a primeira subimagem é a referência de consulta, enquanto a segunda e terceira representam as correspondências correta e incorreta .

Fonte: Zhou et al. (2019)

O tema reidentificação de objetos está em alta, com o objetivo de resolver os problemas complexos, limitados anteriormente pelas tecnologias existentes e técnicas passadas, mas cujas soluções atuais vêm sendo aprimoradas, baseadas em novas abordagens com um suporte computacional mais eficiente, possibilitando desta forma o treinamento e a inferência de modelos de reidentificação mais robustos (SUN et al., 2024).

2.3.1 Métricas de Desempenho para Reidentificação

Para avaliar o resultado obtido das correspondências, durante um processo de reidentificação, duas métricas são comumente utilizadas para essa tarefa: *Cumulative Matching Characteristic* (CMC) e *mean Average Precision* (mAP) (OLIVEIRA; MACHADO; TAVARES, 2021; ALMASAWA; ELREFAEI; MORIA, 2019; KARANAM et al., 2019).

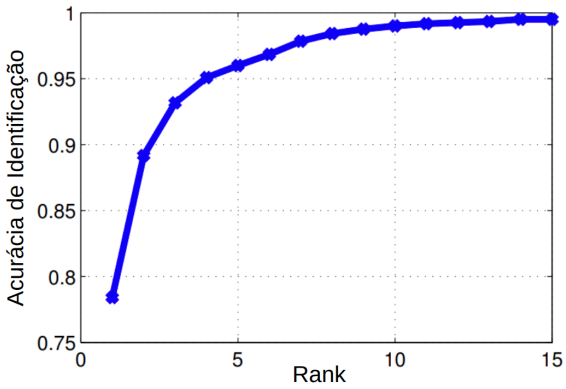
2.3.1.1 Cumulative Matching Characteristic (CMC)

Em *datasets* contendo grande número de imagens na galeria, é comum encontrar dificuldades para realizar a reidentificação exata, pois podem conter diversas amostras similares. Para fins de análise, uma reidentificação que não tenha sido totalmente precisa, mas que apresente resultados próximos da correspondência correta, ainda pode ser considerada satisfatória por fornecer informações relevantes sobre o desempenho do algoritmo avaliado. Conforme mencionam Oliveira, Machado e Tavares (2021), a métrica CMC fornece essa perspectiva dos resultados através da formação de um *ranking*, baseada na acurácia de reidentificação de cada imagem de consulta em relação a toda a galeria. Portanto, o cálculo de CMC para cada *ranking top-k* é dado por:

$$\text{Acc}_k = \begin{cases} 1 & \text{se o } \textit{ranking top-k} \text{ da galeria contiver a identidade da consulta,} \\ 0 & \text{caso contrário.} \end{cases} \tag{2.10}$$

Para construir a curva CMC, repete-se o cálculo para todas as imagens de consulta no conjunto de teste. Em seguida, calcula-se a média das acurácias obtidas para cada valor de k. A curva é então traçada representando o valor de k no eixo horizontal (*ranking*) e a taxa média de acerto acumulada no eixo vertical (acurácia), como ilustra a Figura 10. Essa curva caracteriza o comportamento do algoritmo ao longo de diferentes posições de ranking e permite avaliar até que ponto ele é capaz de recuperar corretamente as identidades, mesmo que não na primeira posição.

Figura 10 – Curva CMC.



Fonte: Adaptado de DeCann e Ross (2013)

2.3.1.2 Mean Average Precision (mAP)

A métrica mAP, definida pela Equação 2.11, é popular na avaliação de experimentos em visão computacional. Além de ser utilizada no desafio de detecção, também é empregada na reidentificação de objetos. Diferentemente do CMC, que considera como verdadeiro positivo qualquer amostra de consulta retornada como correta no ranking gerado, no mAP os valores são computados para considerar as falhas dentro do ranking. Primeiramente é feito o calculo de precisão média AP para cada amostra de consulta q. A seguir é feita a média em relação ao total de imagens de consulta Q para obter um valor único (OLIVEIRA; MACHADO; TAVARES, 2021; SUN et al., 2024).

$$\text{mAP} = \frac{\sum_{q=1}^{Q} \text{AP}(q)}{Q}. \tag{2.11}$$

Complementando, para encontrar AP do contexto sobre recuperação de informação, Manning, Raghavan e Schutze (2008) definem a precisão média como sendo a relação entre a precisão e revocação dos resultados. O cálculo também é realizado formando *ranking* para cada imagem de consulta, porém apenas os *top-k* relevantes são considerados. Então, para cada k relevante é calculada a precisão conforme a Equação 2.12 (SUN et al., 2024).

$$\text{AP} = \frac{1}{|R|} \sum_{k \in R} P(k) \tag{2.12}$$

onde:

- $|R|$ é o número total de posições relevantes (ou seja, o número de imagens relevantes);
- $P(k)$ é a precisão na posição k, dada por:

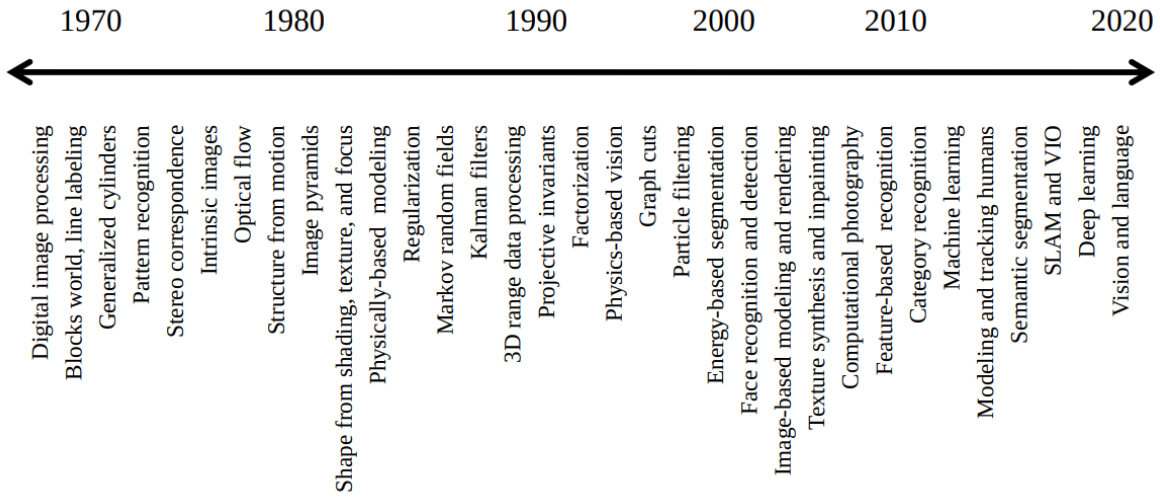
$$P(k) = \frac{\text{Número de imagens relevantes até a posição } k}{k}. \tag{2.13}$$

2.4 Processamento de Imagem no Domínio da Frequência

O processamento de imagens digitais constitui a base de conhecimento necessária para adentrar no campo da visão computacional, permitindo a compreensão dos fundamentos de interpretação e transformações de imagens digitais (SZELISKI, 2022). Como demonstra a linha do tempo na Figura 11, o processamento de imagens digitais foi um dos tópicos precursores nas pesquisas que envolvem visão computacional.

A maioria das operações realizadas em processamento de imagens digitais é comumente elaborada no domínio do espaço, o que facilita a compreensão das operações matriciais. No

Figura 11 – Linha do tempo dos tópicos mais ativos em visão computacional.



Fonte: Szeliski (2022)

entanto, essa abordagem pode trazer uma carga computacional mais intensa ao realizar sequências convolucionais, como as encontradas na maioria das redes neurais nas camadas de convolução (*Convolutional Layers*) (SZELISKI, 2022).

O processamento de imagens digitais no domínio da frequência surge como um ponto de vista diferente, trazendo possibilidades no tratamento de problemas complexos de forma otimizada. No entanto, conforme apontado por Solomon e Breckon (2011), muitos autores introduzem as transformações baseadas nos conceitos de Fourier diretamente por meio de definições matemáticas, o que pode representar uma dificuldade ao aprendizado, devido à complexidade teórica envolvida. Como resultado, o real significado prático da implementação da transformada de Fourier, por exemplo, tende a ser ofuscado.

Gonzalez e Woods (2010) explicam a evolução matemática densa dos conceitos apresentados por Jean-Baptiste Joseph Fourier, encontrados no texto biográfico de 1807 e publicado em 1822 no seu livro sobre "A Teoria Analítica do Calor", tratados com ceticismo na época, pois expressava que qualquer função complexa, sendo periódica e satisfazendo algumas condições matemáticas, pode ser representada pela simples soma de senos e cossenos de diferentes frequências, multiplicados por um coeficiente diferente para cada componente. Essa soma ficou conhecida como a série de Fourier (FOURIER, 1822). Posteriormente, através da transformada de Fourier, replicada para problemas de duas dimensões como imagens digitais para representar funções não periódicas, Gonzalez e Woods (2010) fornecem o contexto e formulação para conversões e tratativas entre o domínio do espaço e da frequência.

A série de Fourier pode ser expressa na notação de números complexos como demonstra a

Equação 2.15, utilizando a fórmula de Euler, conforme a Equação 2.14, onde a função $f(x)$ de uma variável contínua x e período T compõem a soma de senos e cossenos (GONZALEZ; WOODS, 2010).

$$e^{ix} = \cos(x) + i\sin(x) \tag{2.14}$$

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi nx}{T}} \tag{2.15}$$

onde *cn* são os coeficientes multiplicadores e que também constituem o espectro de Fourier, contendo os valores que representam a amplitude e fase das componentes das frequências da função original, dados por:

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(x)e^{-i\frac{2\pi nx}{T}}\,dx \quad \text{para } n = 0, \pm1, \pm2, \ldots \tag{2.16}$$

Já a transformada de Fourier é uma extensão da série de Fourier para representação de funções não periódicas, considerando para isso, que a função periódica tem um período espacial infinito (SOLOMON; BRECKON, 2011). A transformada de Fourier unidimensional pode ser dada como:

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}\,dx \tag{2.17}$$

e sua inversa como:

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega x}\,d\omega \tag{2.18}$$

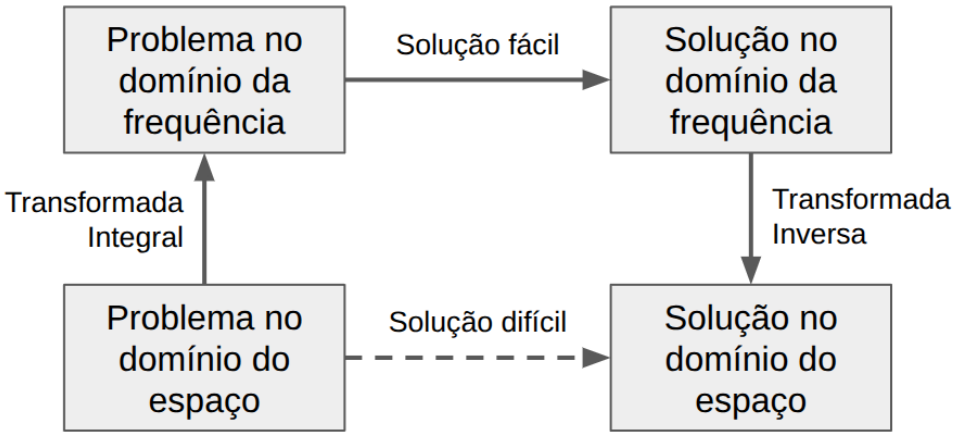
onde a frequência angular ω é:

$$\frac{2\pi}{T}. \tag{2.19}$$

Conforme destacado por Solomon e Breckon (2011), a análise no domínio da frequência através da transformada de Fourier é uma poderosa técnica matemática que consiste no cálculo de integrais para conversão entre o domínio do espaço e da frequência, possibilitando a representação equivalente e completa para esse ambiente abstrato. Os caminhos para realizar essas transformações podem ser sintetizados pelo esquema da Figura 12, que resume como o processamento no domínio do espaço pode ser reformulado de maneira mais

simples e eficiente no domínio da frequência, justificando, assim, a escolha por abordagens que operam nesse domínio para o processamento de imagens.

Figura 12 – Fluxo de transformações entre domínio do espaço e da frequência para simplificar problemas de processamento de imagem.



Fonte: Adaptado de Solomon e Breckon (2011)

Muitas funções importantes já possuem uma prévia relação da transformada de Fourier aplicada, como demonstra o exemplo de Solomon e Breckon (2011) para uma função gaussiana no domínio do espaço:

$$f(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{2.20}$$

sua equivalência transformada no domínio da frequência é:

$$F(k) = \sqrt{2\pi\sigma^2}\exp\left(-\frac{\sigma^2 w^2}{2}\right). \tag{2.21}$$

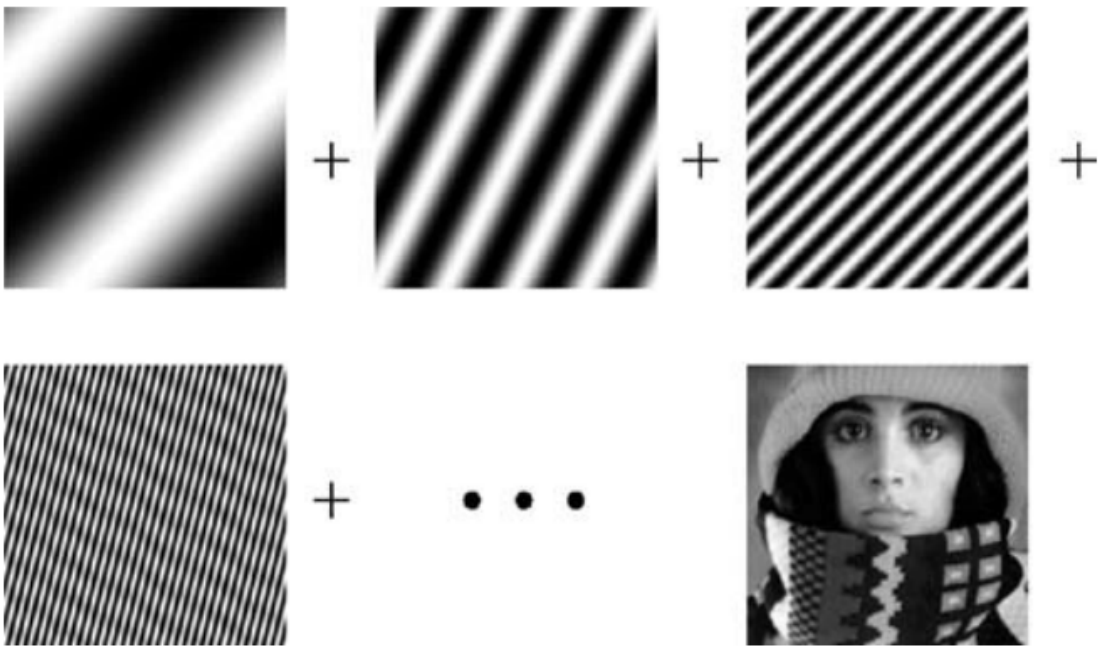
No caso bidimensional, Solomon e Breckon (2011) consideram, de forma análoga ao caso 1-D, como sendo a superposição ponderada de funções harmônicas 2-D, como demonstra o exemplo visual na Figura 13, resultando na imagem final pela inversa da transformada.

Como explica Szeliski (2022), a transformada de Fourier 2-D correspondente é:

$$F(\omega_x, \omega_y) = \iint_{-\infty}^{\infty} f(x, y)\, e^{-i(\omega_x x + \omega_y y)}\, dx\, dy. \tag{2.22}$$

Na sua forma discreta, conhecida como *discrete Fourier transform* (DFT), tem-se:

Figura 13 – Combinação de funções 2-D harmônicas para formar imagem no domínio do espaço.



Fonte: Solomon e Breckon (2011)

$$F(k_x, k_y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(x_m, y_n)\, e^{-i2\pi\left(\frac{k_x x_m}{M} + \frac{k_y y_n}{N}\right)} \tag{2.23}$$

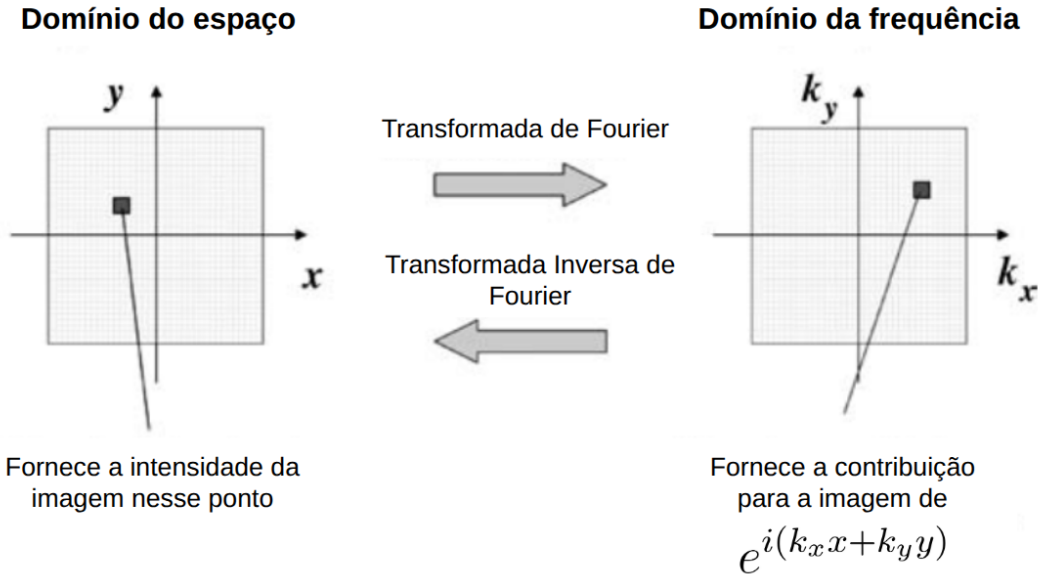
onde:

- k_x e k_y são os índices de frequência espacial discretos;
- M e N são o número de pixels em cada dimensão x e y, respectivamente, ou seja, a largura e altura da imagem.

O cálculo da transformada de Fourier não seria aplicável se fosse necessário computar, na força bruta, a transformação para cada elemento da imagem, requisitando cálculos de ordem $(MN)^2$, onde M e N são o número de pixels em cada dimensão da imagem. Foi através da *fast Fourier transform* (FFT), um algoritmo para cálculo eficiente da transformada discreta de Fourier (DFT), que foi possível reduzir a complexidade computacional para $MN \log_2(MN)$, tornando uma abordagem praticável (GONZALEZ; WOODS, 2010). As instruções e maiores detalhes sobre o desenvolvimento da FFT não serão descritos neste trabalho pela sua extensão, porém, em suma, o algoritmo aplica os cálculos de integrais de forma objetiva para amostragem digital e mantém a essência das propriedades mais relevantes (SOLOMON; BRECKON, 2011).

Um dos pontos mais importantes a se mencionar, para a compreensão dessa relação entre domínio espacial e da frequência, é o mapeamento entre os domínios no plano cartesiano, como demonstra a Figura 14, que explica os valores de intensidade que compõem uma imagem em pixels para suas contribuições nos valores complexos em cada ponto (k_x, k_y) para a função $e^{i(k_x x + k_y y)}$ (SOLOMON; BRECKON, 2011).

Figura 14 – Relação entre domínio do espaço e domínio da frequência.



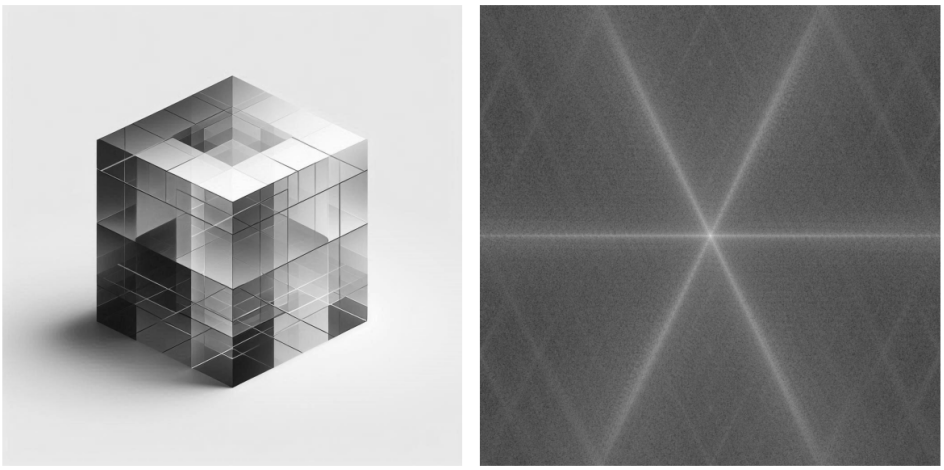
Fonte: Adaptado de Solomon e Breckon (2011)

Dessa forma, após centralizar as frequências baixas no centro e as frequências altas nas bordas, é possível representar o espectro de Fourier amostrado, separando as partes de magnitude e fase, mostradas na Figura 15.

Dentro do processamento digital de imagens, dois dos principais recursos base são as operações de convolução e correlação através da aplicação de *kernels* sobre a imagem, geralmente para tarefas de filtragem linear e reconhecimento de padrões, respectivamente. No domínio do espaço, o resultado da correlação é produzido pelo somatório do produto dos coeficientes do *kernel* aplicados ao pixel central e seus vizinhos abrangidos por esse kernel, para cada deslocamento de posição do *kernel* sobre a imagem inteira. Já a convolução realiza essa mesma operação, mas com o *kernel* previamente rotacionado em 180^o. No domínio da frequência, esse procedimento pode ser realizado diretamente através da multiplicação entre as transformadas resultantes de cada função, ressaltando assim uma das principais vantagens do processamento no domínio da frequência, devido à eficiência computacional nesses casos (GONZALEZ; WOODS, 2010; SOLOMON; BRECKON, 2011; SZELISKI, 2022).

Simplificando as operações matemáticas, a convolução e correlação de um *kernel* H de

Figura 15 – Imagem abstrata de um cubo perfurado com detalhes isométricos e a resposta para a magnitude do espectro de Fourier correspondente. As arestas compõem boa parte da representação da imagem, portanto ficam destacadas no espectro.



Fonte: Próprio autor.

tamanho (m, n) por uma imagem de entrada X, podem ser dadas de forma equivalente conforme a Tabela 1 entre domínio do espaço e da frequência.

Tabela 1 – Comparação das formulações de convolução e correlação nos domínios do espaço e da frequência. Para correlação no domínio da frequência é necessário utilizar o complexo conjugado do *kernel* para ter a inversão correspondente que não é aplicada na convolução, isso reproduz o espelhamento da função.

Operação	**Domínio do Espaço**	**Domínio da Frequência**
Convolução	$(X * H)(i,j) = \sum_{m,n} X(i-m, j-n) \cdot H(m,n)$	$\mathcal{F}\{X * H\} = \mathcal{F}\{X\} \cdot \mathcal{F}\{H\}$
Correlação	$(X \star H)(i,j) = \sum_{m,n} X(i+m, j+n) \cdot H(m,n)$	$\mathcal{F}\{X \star H\} = \mathcal{F}\{X\} \cdot \overline{\mathcal{F}\{H\}}$

Aplicações recentes tentam retomar o campo de estudo no domínio da frequência para extrair maior riqueza de detalhes e maior desempenho aliando essas abordagens a parte do processamento feito por redes neurais. Por exemplo, Stuchi et al. (2023) propõem o aprendizado de filtros discriminativos no domínio da frequência com redes neurais convolucionais para aplicações como classificação de textura, detecção da doença ocular de catarata e também análise de retina, demonstrando resultados positivos com um modelo mais leve se comparado às tradicionais CNNs. De forma similar, Rehman, Hussein e Sultani (2024) incorporam a extração de características baseadas na união de imagens tratadas

no domínio do espaço e da frequência, para reconhecer padrões de regiões malignas na prevenção do câncer de mama. Em sua conclusão, mencionaram que a combinação de características vindas dos domínios do espaço e da frequência proporcionou melhoria nos resultados.

Com isso é possível perceber que os trabalhos no domínio da frequência podem fornecer benefícios no desempenho de algoritmos baseados na identificação visual de padrões assim como outros processamentos em visão computacional. O que cria barreira muitas vezes é a complexidade na compreensão dos conceitos no domínio da frequência, que, quando aprofundados, podem ser mais difíceis e extensos se comparados às técnicas de trabalho no domínio do espaço.

2.5 Filtro de Correlação Discriminativo

O filtro de correlação discriminativo (DCF) geralmente é associado ao problema de rastreamento visual de objetos (VOT), sendo utilizado principalmente como abordagem no rastreio de objeto único, realizado por meio de treinamento supervisionado para aprendizado de um modelo de regressão linear, quando o objeto em seu estado inicial é conhecido. Dessa forma, é possível construir o modelo ao longo dos próximos frames que servem de base para consulta e comparação conforme o objeto se movimenta pela cena (JAVED et al., 2021).

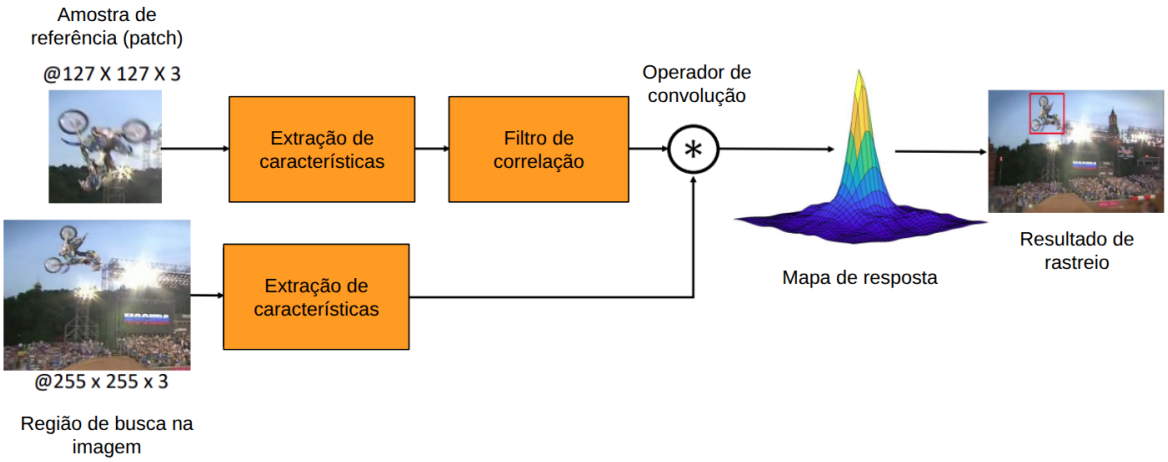
Conforme mencionado por Lukezic et al. (2018), o filtro de correlação discriminativo data do ano de 1980, com o trabalho de Hester e Casasent (1980), mas apenas se popularizou após ser utilizado nos desafios de rastreamento de objetos em visão computacional, principalmente após a publicação de Bolme et al. (2010) em 2010 e de Henriques et al. (2015) em 2015. Com a transformada de Fourier como um artifício para realizar múltiplas operações matriciais de forma eficiente no domínio da frequência, atingiram-se resultados de precisão satisfatória, a uma alta taxa de quadros por segundo (FPS), indicando sua viabilidade para processamento em tempo real.

O foco principal do DCF é encontrar um filtro de correlação (*kernel*), que discrimine bem uma imagem, através da aplicação de um sinal na entrada que produza na saída uma resposta bem definida. A aplicação de um *kernel*, previamente treinado com base em uma imagem conhecida do objeto, através da correlação ou convolução com a imagem-alvo, produz na saída algo como um pico gaussiano, tipicamente utilizado, onde esse pico representa a possível localização do objeto procurado (JAVED et al., 2021).

Um exemplo de sua utilização pode ser visualizado no *pipeline* da Figura 16, onde uma imagem de referência de 127x127 pixels é utilizada para gerar o filtro de correlação que

melhor represente as características do objeto. Esse filtro é então aplicado à imagem completa, permitindo localizar a posição do objeto por meio do pico gaussiano no mapa de resposta, o qual indica a região de maior correspondência, possibilitando a reidentificação do objeto mesmo sob variações de rotação.

Figura 16 – *Pipeline* clássico de rastreio visual utilizando DCF.



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

Detalhando um pouco mais sobre alguns trabalhos que foram marcos no uso de filtros de correlação, o *Minimum Output Sum of Squared Error* (MOSSE) proposto por Bolme et al. (2010), traz uma abordagem com processamento a alta taxa de quadros por segundo e robustez a variações de iluminação, escala, pose, deformações e oclusão parcial através da análise do mapa de resposta com *Peak-to-Sidelobe Ratio* (PSR), que será explicado na Seção 2.5.1. O conceito principal é encontrar um filtro que minimize a soma dos erros quadrados (SSE) entre a saída e a resposta desejada. No domínio da frequência, isso é expresso como:

$$\text{SSE} = \sum_i |X_i \cdot H^* - G_i|^2 \tag{2.24}$$

onde:

- G_i é a resposta desejada no domínio da frequência para a i-ésima amostra do treinamento do *kernel*(*patch*);
- H é filtro de correlação no domínio da frequência;
- H^* é o complexo conjugado de H;
- X_i é o i-ésimo *patch* da imagem no domínio da frequência.

Para encontrar um filtro H que minimiza o SSE em uma solução otimizada, é feita a derivada parcial da SSE em relação a H resultando na Equação 2.25, que consiste no somatório de amostras (*patches*) utilizadas para treinamento. No numerador é feita a correlação entre a resposta desejada e a imagem de referência, enquanto no denominador é feita a autocorrelação da imagem base (BOLME et al., 2010).

$$H^* = \frac{\sum_i (G_i \cdot X_i^*)}{\sum_i (X_i \cdot X_i^*)}. \tag{2.25}$$

Uma evolução do algoritmo MOSSE foi proposta por Henriques et al. (2015), denominada *Kernelized Correlation Filter* (KCF). Uma de suas principais contribuições foi o uso de matrizes circulantes para gerar, de forma eficiente, um grande número de amostras a partir de uma única imagem base. Essas amostras são usadas durante o treinamento *online*, isto é, o processo de atualização contínua do filtro de correlação ao longo do tempo, à medida que novos quadros são processados. Esse mecanismo permite que o rastreador se adapte dinamicamente às mudanças na aparência do objeto.

Outra melhoria importante foi a aplicação do truque do *kernel* (*kernel trick*) no domínio da frequência, que permite mapear os dados de entrada para um espaço de maior dimensionalidade, viabilizando a modelagem de relações não lineares entre as amostras, sem aumento significativo de custo computacional. Com isso, o filtro de correlação é capaz de operar em um espaço de características mais expressivo.

Além disso, o KCF introduziu suporte a múltiplos canais de imagem e incorporou a extração de características HOG (Histogram of Oriented Gradients) ao *pipeline* de rastreamento, aumentando a robustez frente a variações na textura e na forma dos objetos rastreados.

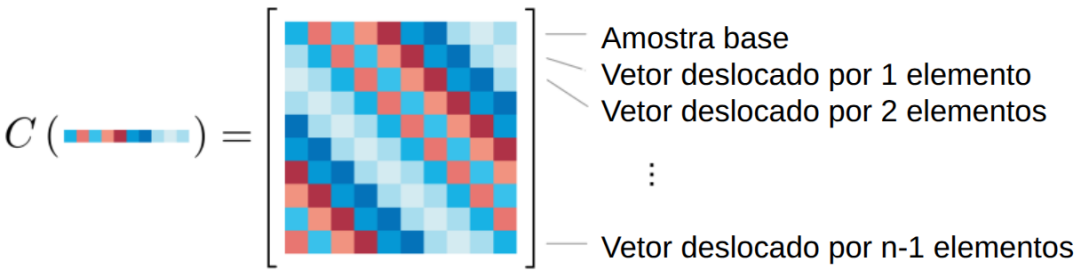
Henriques et al. (2015) destacam que, ao invés de realizar uma iteração sobre todos os deslocamentos cíclicos da imagem base, como demonstrado nas Figuras 17 e 18, a propriedade de diagonalização da Transformada Discreta de Fourier (DFT) pode ser utilizada para tornar esse processo mais eficiente.

Na prática, isso significa que uma matriz circulante, isto é, uma matriz em que cada linha é uma rotação cíclica da linha anterior, pode ser transformada, via DFT, em uma matriz diagonal no domínio da frequência. Como a convolução entre sinais no domínio do espaço corresponde a uma multiplicação no domínio da frequência, essa propriedade permite que todas as correlações entre a imagem base e seus deslocamentos cíclicos sejam computadas simultaneamente e de forma mais eficiente, apenas com multiplicações elemento a elemento no domínio da frequência, otimizando assim a criação de variações da imagem base e reduzindo significativamente o tempo computacional que seria necessário no domínio do espaço. Uma explicação teórica detalhada dessa propriedade pode ser encontrada, por

Figura 17 – Deslocamento cíclico vertical de uma imagem base. O KCF aplica o deslocamento vertical e horizontal para todas possíveis variações da imagem de referência (*patches*).

Fonte: Adaptado de Henriques et al. (2015).

Figura 18 – Composição de uma matriz circulante baseada em uma imagem vetorizada. As linhas são sequências de deslocamentos do vetor inicial em um elemento por vez.



Fonte: Adaptado de Henriques et al. (2015).

exemplo, em Gray (2006), que aborda como a DFT atua como base ortonormal para diagonalização de matrizes circulantes, ou seja, um conjunto de vetores ortogonais e normalizados.

Para um *kernel* gaussiano, conforme proposto por Henriques et al. (2015), é possível trabalhar inteiramente no domínio da frequência por meio da Transformada Rápida de Fourier (FFT). No treinamento, utiliza-se uma amostra inicial da imagem-alvo e gera-se uma matriz circulante a partir de deslocamentos cíclicos dessa amostra, o que permite formular diretamente os coeficientes do filtro como uma solução fechada no domínio da frequência. Desta forma, em vez de resolver um sistema linear de alta dimensão no domínio do espaço, obtém-se o vetor de pesos do filtro através de operações ponto a ponto no domínio da frequência, reduzindo a complexidade computacional. Na detecção, o filtro treinado é correlacionado com uma nova região candidata da imagem do *frame* em análise, que também é convertida para o domínio da frequência. Assim, o pico do mapa de resposta indica a posição estimada do objeto. Com essas otimizações matemáticas, o Algoritmo 1 pode ser aplicado de forma eficiente para executar tanto as etapas de treino quanto de detecção em tempo real.

Algoritmo 1: Treinamento e detecção com *Kernelized Correlation Filter* (KCF) utilizando *kernel* Gaussiano. Adaptado de (HENRIQUES et al., 2015).

Input:

x: *patch* de imagem para treinamento ($m \times n \times c$),
y: resposta desejada em forma de gaussiana ($m \times n$),
z: *patch* de imagem para teste ($m \times n \times c$),
σ: parâmetro do *kernel* gaussiano,
λ: parâmetro de regularização.

Output:

responses: mapa de resposta para detecção ($m \times n$).

Função `train`(x, y, σ, λ)**:**

$k \leftarrow$ `kernel_correlation`(x, x, σ);
$\alpha_f \leftarrow \text{FFT2}(y) \oslash (\text{FFT2}(k) + \lambda)$;
return α_f

Função `detect`(α_f, x, z, σ)**:**

$k \leftarrow$ `kernel_correlation`(z, x, σ);
$responses \leftarrow \text{Re}(\text{IFFT2}(\alpha_f \odot \text{FFT2}(k)))$;
return *responses*

Função `kernel_correlation`(x_1, x_2, σ)**:**

$C \leftarrow \text{IFFT2}(\sum_{i=1}^{c} \text{conj}(\text{FFT2}(x_1^i)) \odot \text{FFT2}(x_2^i))$;
$D \leftarrow ||x_1||^2 + ||x_2||^2 - 2 \cdot C$;
$k \leftarrow \exp\left(-\frac{1}{\sigma^2} \cdot \frac{|D|}{N}\right)$; `// N é o número de elementos em D`
return k

Além dos exemplos citados, como KCF que utiliza extração de características *Rawpixel* e HOG, outras abordagens com diversas técnicas para aprimorar os algoritmos de DCF, como *Continuous Convolution Operators Tracking* (CCOT) e *Efficient Convolution Operators* (ECO), utilizam outros recursos para representar as características das imagens, como *Color Name* e *Deep features* (JAVED et al., 2021).

Por oferecer algoritmos de processamento rápido no domínio da frequência e precisão considerável, muitas aplicações que exigem processamento em tempo real ou hardware limitado se beneficiam do filtro de correlação.

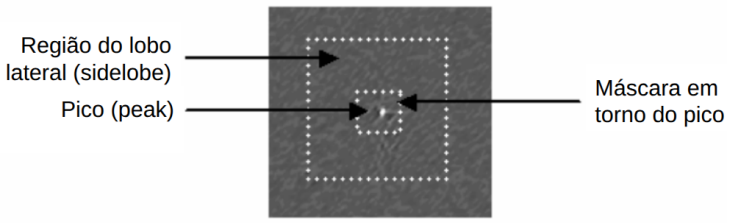
2.5.1 Peak-to-Sidelobe Ratio

Como visto anteriormente sobre filtros de correlação discriminativos, uma resposta de correlação é esperada para indicar a força do sinal produzido pela similaridade entre as imagens ou características comparadas.

Adebayo (2013) menciona que a métrica *Peak-to-Sidelobe Ratio* (PSR) é comumente utilizada para avaliar a qualidade da resposta de correlação, fornecendo uma decisão de classificação mais confiável. O PSR é definido como a razão entre o valor máximo da resposta de correlação R_{max} e a média dos valores da resposta em uma região ao redor desse pico $\mu_{R_{sidelobes}}$, levando em consideração as variações nessa região através do desvio padrão $\sigma_{R_{sidelobes}}$, excluindo a área próxima ao pico. A expressão matemática é explicada pela Equação 2.26 e pode ser visualmente representada pela Figura 19 (SAWIDES; KUMAR; KHOSLA, 2004).

$$\text{PSR} = \frac{R_{max} - \mu_{R_{sidelobes}}}{\sigma_{R_{sidelobes}}}. \tag{2.26}$$

Figura 19 – Região para estimativa do PSR.



Fonte: Adaptado de Sawides, Kumar e Khosla (2004).

Um valor alto de PSR sugere que o pico da resposta é claramente distinguível dos "lobos laterais", indicando uma maior confiabilidade tanto de correlação quanto de localização. Por outro lado, um PSR baixo pode resultar de uma resposta de correlação que contenha múltiplos picos ou ruídos, dificultando a identificação precisa do objeto alvo. Isso pode indicar que os sinais de entrada, no caso, imagens, são diferentes ou que há oclusão, algo que enfraquece a definição do sinal resultante.

2.6 Extração de Características Clássicas

A extração de características clássica faz parte dos pilares fundamentais em tarefas de visão computacional, servindo como base para o reconhecimento, rastreamento e análise de imagens. As características são informações extraídas de uma imagem que capturam aspectos relevantes para a tarefa específica, como bordas, texturas, formas ou cores. Algumas das abordagens clássicas mais conhecidas serão descritas a seguir de forma superficial para compreensão da aplicabilidade de cada uma.

2.6.1 Rawpixel

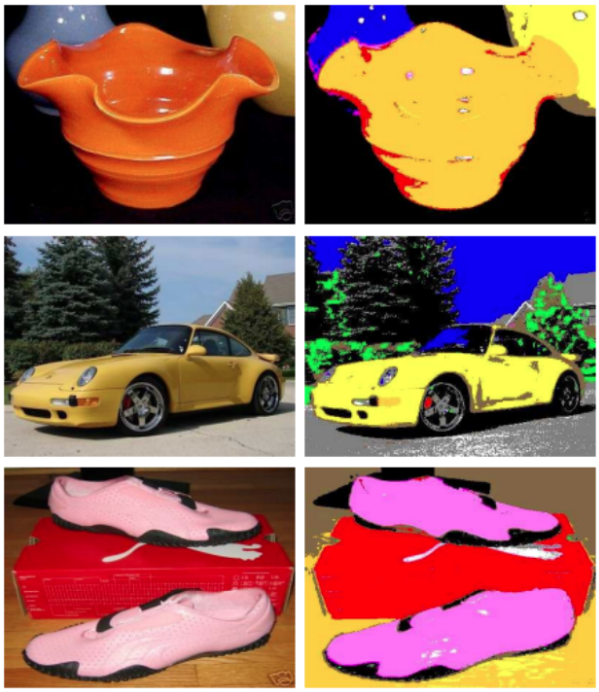
Rawpixel não se trata de uma abordagem para extrair características propriamente dita, mas sim baseia-se no uso direto dos valores dos pixels da imagem. Apesar de simples, essa

abordagem pode ser poderosa em certos contextos, especialmente quando combinada com métodos de aprendizado que conseguem identificar padrões diretamente dos dados brutos, como é feito em algoritmos de correlação, como mencionado por Henriques et al. (2015). No entanto, o uso desses dados diretamente pode ser sensível a variações de iluminação, ângulo e ruído, o que pode limitar sua eficácia em situações mais complexas.

2.6.2 Color Name

Essa técnica converte os valores de cor dos pixels em uma representação de mais fácil interpretação, utilizando um conjunto fixo de nomes de cores pré-definidos, como vermelho, amarelo, azul e outras cores, como ilustrado na Figura 20. A técnica consiste em uma forma de discretização semântica para realizar o mapeamento de cores baseado em como as pessoas normalmente descrevem as cores (WEIJER et al., 2009).

Figura 20 – Discretização de imagens utilizando Color Name. A primeira coluna contém as imagens originais e a segunda coluna representa as características de cor extraídas.



Fonte: Weijer et al. (2009).

Por exemplo, um mapeamento simplificado pode definir regiões do espaço RGB para cores como:

- Vermelho: R $>$ 200, G $<$ 80, B $<$ 80
- Verde: R $<$ 80, G $>$ 200, B $<$ 80

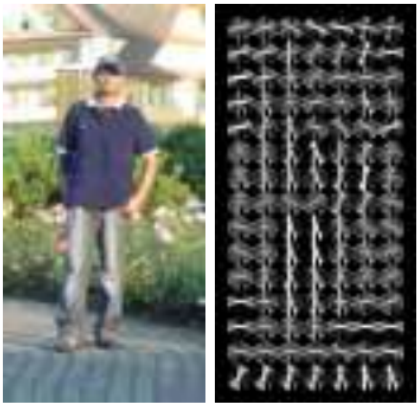
- Azul: R < 80, G < 80, B > 200
- Amarelo: R > 200, G > 200, B < 100

Essa representação facilita a comparação entre imagens ao reduzir a sensibilidade a variações de cor que são perceptivamente irrelevantes, como pequenas mudanças de tonalidade. A abordagem é especialmente útil em tarefas de reconhecimento e classificação de objetos, onde a cor desempenha um papel discriminativo importante, como na identificação de roupas, alimentos ou veículos (WEIJER et al., 2009).

2.6.3 HOG

Histogram of Oriented Gradients é uma técnica que captura as distribuições de gradientes de intensidade na imagem. Basicamente, a imagem é dividida em pequenas células, e para cada célula é computado um histograma de direções de gradientes, onde o resultado dessas direções pode ser indicado como mostrado na Figura 21. HOG é especialmente eficaz na detecção de objetos, pois captura informações sobre as bordas e formas presentes na imagem, o que o torna robusto a variações de iluminação e pequenas deformações (DALAL; TRIGGS, 2005).

Figura 21 – Imagem de teste e a representação das características HOG computadas.



Fonte: Dalal e Triggs (2005).

2.6.4 LBP

Local Binary Patterns (LBP) é um descritor que captura informações de textura, comparando cada pixel com seus vizinhos imediatos, gerando um padrão binário que representa a textura local, como demonstra o exemplo na Figura 22. A simplicidade e a eficácia do LBP o fizeram ser uma escolha popular para tarefas de reconhecimento facial e análise de textura, onde as variações locais de textura são cruciais (OJALA; PIETIKäINEN; HARWOOD, 1996; BABU et al., 2022).

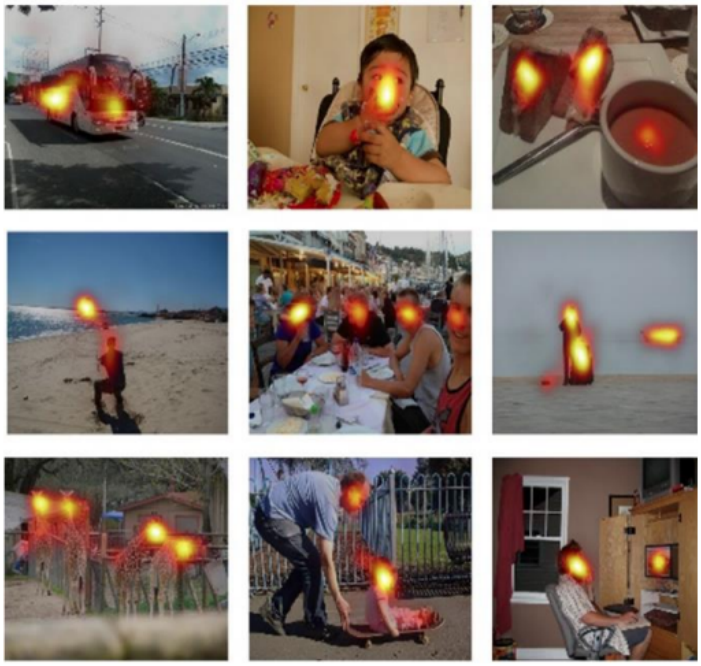
Figura 22 – Extração de características faciais com LBP para reconhecimento de emoções.

Fonte: Adaptado de Babu et al. (2022).

2.6.5 Saliency Maps

Saliency Maps são representações visuais que indicam as regiões de uma imagem ou cena que são mais suscetíveis de atrair a atenção humana, como mostram as imagens representadas na Figura 23. Essa técnica é baseada no conceito de que, ao observar uma cena, o olho humano não processa todas as informações visuais de maneira uniforme, mas sim foca automaticamente nas áreas mais "salientes", aquelas que se destacam devido ao contraste, cor, textura, movimento ou outras características visuais, como ilustrado nas segmentações feitas para cada imagem da Figura 24 (ZHAO; KOCH, 2011; BORJI et al., 2019; ULLAH et al., 2020).

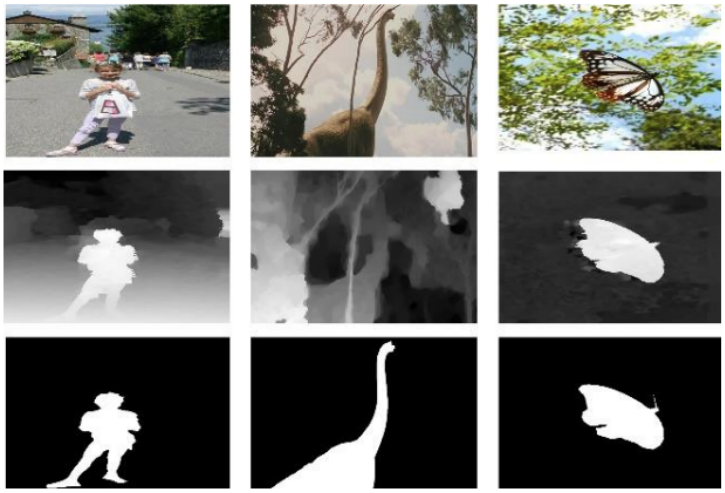
Figura 23 – Predição de atenção do olho humano em algumas imagens.



Fonte: Ullah et al. (2020).

As abordagens *saliency-based* são fundamentais em uma ampla gama de aplicações, desde

Figura 24 – Mapas de saliência aplicados à segmentação de imagem. A primeira linha contém as imagens originais, a segunda a segmentação obtida pela saliência detectada e a terceira coluna o *ground truth*.



Fonte: Ullah et al. (2020).

segmentação de imagens, reconhecimento de objetos, assistência visual, análise de conteúdo de mídia e foco automático de câmera, destacando informações visuais essenciais para otimizar processos em diversas tecnologias.

2.6.6 Considerações

Cada uma dessas técnicas oferece uma perspectiva única sobre os dados de imagem, permitindo que modelos de visão computacional capturem diferentes aspectos visuais relevantes. A escolha da técnica de extração de características depende fortemente do contexto e do tipo de informação que se deseja extrair da imagem, sendo muitas vezes benéfico combinar múltiplas abordagens para alcançar uma representação mais robusta e abrangente.

2.7 Trabalhos Relacionados

Muitos trabalhos são desenvolvidos especializados em diversos desafios da visão computacional, focados em cada segmento ou, em alguns casos, na interseção entre os mesmos, como a proposta deste trabalho, que visa reunir uma abrangência de múltiplos conhecimentos base, porém com foco maior na reidentificação e rastreamento de objetos.

A utilização de filtros de correlação discriminativos (DCFs) tem sido explorada há alguns anos em pesquisas sobre rastreamento visual de objetos. No entanto, mesmo em contextos nos quais ocorre reidentificação de forma implícita, essas abordagens geralmente não tratam os DCFs como uma solução independente voltada exclusivamente para o problema

de reidentificação, dissociada do rastreamento. A seguir, são apresentados trabalhos que abordam tanto a reidentificação quanto o rastreamento de objetos, utilizando metodologias variadas. Entre eles, destacam-se algumas propostas que integram DCFs para lidar de forma conjunta com os desafios dessas duas tarefas.

2.7.1 Soluções para Reidentificação com Metodologias Clássicas

A reidentificação de objetos cumpre um papel importante nas tarefas de visão computacional. Nos últimos anos, esse desafio da área tem se intensificado com a atenção de novas pesquisas, porém, aplicações específicas para videomonitoramento utilizando técnicas clássicas estão presentes há mais tempo entre os métodos empregados.

Exemplo disso é o trabalho proposto por Farenzena et al. (2010), intitulado de "*Person Re-identification by Symmetry-Driven Accumulation of Local Features*" (SDALF), que propõe uma abordagem para reidentificação de pessoas com base na simetria do corpo humano. Essa metodologia foca na extração e acúmulo de características locais de uma maneira que leva em conta a estrutura simétrica do corpo. Para formar o conjunto de características, foram utilizados recursos como histograma de cores ponderado, regiões de cores com máxima estabilidade e recorrência nos padrões de textura.

Baseados nos conceitos de atenção da visão humana para detectar regiões salientes em imagens, Zhao, Ouyang e Wang (2013) aplicam o aprendizado não supervisionado de regiões salientes com *K-Nearest Neighbor* (KNN) e como segunda opção *One-class SVM* (OCSVM). Desta forma, não é necessário rotular previamente um *dataset* de forma exaustiva, podendo executar a reidentificação automática de pessoas com resultados melhorados por meio de um treinamento simples e eficiente, conforme comparações feitas com abordagens predecessoras.

Já Liao et al. (2015) propõem a representação de características denominada *Local Maximal Occurrence* (LOMO), que maximiza a ocorrência horizontal de características locais para estabilizar a representação contra mudanças de perspectiva. Além disso, o método aplica transformações para lidar com variações de iluminação e propõe uma métrica de aprendizado para medir de forma discriminativa a similaridade entre as características extraídas. Os experimentos, realizados em quatro *datasets* desafiadores de reidentificação de pessoas, demonstraram que o método proposto melhora as taxas de identificação *rank-1* em até 31,55%.

Li et al. (2017) trazem a reidentificação para a aplicação de busca seletiva de pessoas presentes em um conjunto de imagens provenientes de cenários reais. Sua proposta, *Correlation-based Identity Filter* (CIF), utiliza os conceitos de filtro de correlação para encontrar um filtro discriminativo baseado em amostras positivas da pessoa de referência

e amostras negativas de qualquer outra pessoa ou imagens de fundo. A abordagem utiliza extração de características por histograma de cores e HOG. Os autores concluem que, apesar das dificuldades encontradas, como similaridade entre as pessoas, variações de iluminação, aparência e oclusão, o método proposto trouxe bons resultados, sendo um modelo extremamente leve e eficiente para busca de pessoas em uma imagem.

2.7.2 Soluções para Reidentificação com CNNs

Novos estudos sobre a reidentificação de pessoas surgiram e algumas soluções que incorporam a extração de características com redes neurais convolucionais para a construção de modelos de reidentificação se tornaram mais comuns.

O artigo proposto por Li et al. (2014) aborda os desafios da reidentificação de pessoas em imagens de câmeras não sobrepostas, onde variações de iluminação, pose, ângulo de visão, diferentes resoluções de imagem e oclusões aumentam a complexidade do reconhecimento de padrões. A proposta inovadora do artigo é a rede neural *Filter Pairing Neural Network* (FPNN), que automaticamente aprende características ideais para a reidentificação, em vez de depender de características manuais (*handcraft features*), anteriormente mencionadas como características clássicas. A FPNN lida de forma conjunta com desalinhamentos, transformações fotométricas e geométricas, oclusões e desordem de fundo, maximizando a eficácia de cada componente através de uma arquitetura profunda. A abordagem foi avaliada em um grande conjunto de dados contendo 13.164 imagens de 1.360 pedestres, capturadas com caixas delimitadoras detectadas automaticamente, refletindo um cenário mais próximo de aplicações práticas em comparação com conjuntos de dados anteriores, que utilizavam imagens recortadas manualmente.

Zheng, Yang e Hauptmann (2016) oferecem uma revisão abrangente do campo da reidentificação de pessoas, destacando sua evolução desde os primeiros métodos baseados em características manuais até as abordagens mais recentes que utilizam aprendizado profundo em grandes conjuntos de dados. O texto classifica os métodos de reidentificação em duas categorias principais: baseados em imagens e em vídeos, que se enquadram nos conceitos de reidentificação *single-shot* e *multi-shot*. O estudo também discute a história da reidentificação de pessoas, sua relação com a classificação de imagens e sua origem através do rastreamento em múltiplas câmeras, que agora é tratado como uma tarefa independente, além de apontar direções futuras e críticas, como a otimização para processamento em grandes volumes de dados, a eficiência em tempo real e formas unificadas de avaliação de sistemas de reidentificação.

A reidentificação de pessoas é um dos tópicos mais comuns nos estudos dessa subárea da visão computacional, porém, mais recentemente, outros segmentos desse desafio vêm sendo elaborados. Amiri, Kaya e Keceli (2024) proporcionam uma visão detalhada das técnicas

de aprendizado profundo aplicadas à reidentificação de "veículos", um campo crucial para o desenvolvimento em ITS (*Intelligent Transportation Systems*) e iniciativas de cidades inteligentes. O estudo classifica os métodos de reidentificação de veículos em abordagens supervisionadas e não supervisionadas, revisa as pesquisas existentes em cada categoria e apresenta conjuntos de dados e critérios de avaliação relevantes. Dentro dos métodos supervisionados, a maioria dos modelos foca na obtenção de características visuais distintas das imagens de veículos, tratando o problema como uma tarefa de classificação, enquanto outros priorizam o aprendizado profundo de métricas com funções de perda específicas. O objetivo do artigo proposto é servir como um guia completo para pesquisadores e profissionais, promovendo o progresso na utilização de modelos de aprendizado profundo para a reidentificação de veículos.

2.7.3 Soluções para Rastreamento com DCF e Metodologias Clássicas

O artigo de Lukezic et al. (2018) aborda o problema desafiador do rastreamento de objetos de curto prazo, onde os filtros de correlação discriminativos têm mostrado excelente desempenho. A principal contribuição do trabalho é a introdução de alguns conceitos de confiabilidade no rastreamento com DCF. O mapa de confiabilidade espacial ajusta o suporte do filtro para as partes do objeto mais adequadas para rastreamento, permitindo ampliar a região de busca e melhorar o rastreamento de objetos. Com apenas dois conjuntos de características padrão simples, HOG e *Color Name*, o método DCF com Confiabilidade de Canal e Espacial (CSR-DCF) alcança ótimos resultados nos *benchmarks* VOT 2016, VOT 2015 e OTB100, rodando quase em tempo real em uma CPU.

Zhu et al. (2021) reforçam a limitação de muitos rastreadores que são computacionalmente custosos. Nesses casos, a utilização de filtros de correlação se torna adequada para sistemas de recursos mais limitados, porém uma das partes do algoritmo que restringe sua robustez é o esquema de atualização do filtro ao longo do rastreio. O trabalho propõe um esquema de atualização de amostras com treinamento mais eficiente e adaptativo, utilizando um algoritmo de *hashing* de diferenças para medir a distância entre amostras e determinar se uma nova amostra deve ser atualizada ou descartada com base na confiabilidade dos resultados de rastreamento. Além disso, o método é estendido para rastreamento de longo prazo, introduzindo um novo mecanismo de discriminação de estado de rastreamento para identificar falhas e retomar o rastreamento após um período maior.

Segundo Yadav e Payandeh (2022), diferentemente dos modelos de redes neurais, que na maioria dos casos necessitam de grande conjunto de dados para treinamento, abordagens baseadas em filtros de correlação, como o KCF, utilizam recursos fornecidos durante o rastreio para realizar o treinamento *online* dos modelos discriminativos de cada objeto. Apesar de eficazes para rastreamento em tempo real, apresentam limitações significativas em

cenários de oclusões e alvos fora do campo de visão. O trabalho introduz um rastreador de correlação desenvolvido com base no KCF e aprimorado com informações de profundidade (RGB-D) para "redetecção" de alvos. Os resultados experimentais são avaliados em tempo real, usando o sensor *Microsoft Kinect V2*, e a comparação entre o algoritmo de rastreamento KCF (RGB) e a versão proposta (RGB-D) demonstram que esta abordagem melhora a acurácia em situações complexas como oclusão, porém adiciona uma carga de processamento maior, com o KCF original rodando a 141 FPS e o proposto a 10 FPS.

2.7.4 Soluções para Rastreamento com DCF e CNNs

A integração entre DCF e redes neurais convolucionais surge para aumentar a robustez do algoritmo em cenários mais complexos, onde é mais difícil correlacionar as características específicas do objeto-alvo, necessitando de aprofundamento na análise de características.

Uma modalidade de integração mais colaborativa entre as duas tecnologias é utilizar uma CNN para extrair características ao invés de extratores *handcraft* e alimentar o DCF para correlacionar as informações. Valmadre et al. (2017) propõem uma rede siamesa para extrair as características, sendo um dos caminhos voltados para o treinamento do filtro de correlação com base em uma imagem de referência, onde posteriormente é feita a correlação com a imagem teste para encontrar a similaridade. De forma parecida, para aplicação em veículos autônomos, Zhao et al. (2018) também alimentam o filtro de correlação através de características extraídas de uma CNN, porém o estudo é feito para aprimorar a etapa de detecção com *Single-Shot Detector* (SSD), onde a própria rede que detecta fornece os atributos para o treinamento do DCF, que irá fornecer dados para realizar a reidentificação dos objetos já rastreados.

Fernandez-Sanjurjo, Mucientes e Brea (2021) seguem uma proposta mais independente para rastreamento de múltiplos objetos em tempo real, focada em sistemas embarcados com suporte a GPU. Sua abordagem se baseia em três recursos para computar a similaridade espacial e visual. Primeiramente, utiliza os vetores de características extraídos pela rede de detecção YOLOv3 e retém essa informação para cada objeto rastreado associado. Também reúne a localização e caixas delimitadoras do detector e as preditas pelo Filtro de Kalman em conjunto com as predições do filtro de correlação KCF. Dessa forma, são computadas as distâncias entre os vetores de características visuais com a distância de Mahalanobis. Para as características espaciais adquiridas com o filtro de Kalman e KCF, é computado o *Intersection over Union* (IoU). Os resultados demonstraram que foi possível obter melhorias no rastreio testado no *dataset* MOT16, atingindo a marca de 51,1% para MOTA e uma média de 25 FPS rodando em uma placa NVIDIA Jetson TX2.

2.7.5 Considerações

Com os trabalhos vistos até o momento, considerando também a trajetória da fundamentação teórica, é possível notar a reincidência de aplicações que utilizam DCF como uma abordagem mais eficiente computacionalmente, em virtude dos pontos negativos de se utilizar redes neurais mais pesadas, alternativa ideal para *hardwares* que possuem alguma limitação ou para sistemas que necessitam de processamento em tempo real.

Pode-se notar que, apesar de transitar pelos assuntos de reidentificação e rastreamento de objetos, filtro de correlação é, na maioria dos estudos, implementado no rastreio visual de objetos, sendo esses para rastreio único apenas. O trabalho de Li et al. (2017) se aproxima em relação à reidentificação proposta nesse trabalho de dissertação, ao se utilizar um filtro de correlação especificamente para tarefa de reidentificação de pessoas, porém a abordagem não explora muito esse campo e traz uma solução mais clássica e voltada para detecção de padrões em uma imagem inteira, na busca pela pessoa referenciada durante o treinamento do filtro.

O artigo de Fernandez-Sanjurjo, Mucientes e Brea (2021) traz uma solução que também se aproxima da proposta deste trabalho em relação à integração de um DCF ao rastreamento, nesse caso já buscando uma aplicação mais realista para rastreamento de múltiplos objetos. Uma combinação mais independente entre metodologias já existentes de detecção e rastreamento simples, aliados à parte de reidentificação com o KCF, se aproxima da ideia aqui proposta para unir um módulo de reidentificação DCF, simbolizando essa independência dos processos. Caso o detector ou rastreador simples mude para uma versão melhorada, isso não irá impactar fortemente o *pipeline* de reidentificação.

Portanto, apesar de próximos, não foram encontrados trabalhos que relacionem o uso de DCFs diretamente para o desafio de reidentificação como uma tarefa completamente isolada, ou seja, apenas com a finalidade de computar a similaridade entre imagens dos objetos-alvo, o que requer um estudo focado no desafio da visão computacional de reidentificação. Assim como os demais, a implementação do filtro de correlação também será integrada ao rastreamento de objetos para analisar o comportamento nas duas tarefas principais e se há aplicabilidade voltada para videomonitoramento visando o processamento em tempo real.

3 REIDENTIFICAÇÃO DE OBJETOS BASEADA EM FILTRO DE CORRELAÇÃO

Neste capítulo, serão descritos os modelos de arquiteturas propostas para experimentação e quais estratégias foram adotadas para adequar o filtro de correlação discriminativo com foco em reidentificação. Importante mencionar que, até aqui, os algoritmos ou modelos ainda não foram associados com rastreamento de objetos diretamente. Também será feita a comparação com um algoritmo que utiliza redes neurais convolucionais em diferentes cenários. Como o contexto deste trabalho envolve câmeras de videomonitoramento, os *datasets* usados são voltados para identificação de pessoas e veículos. Ao final deste capítulo, serão feitas algumas considerações sobre os resultados obtidos.

3.1 Arquitetura para Reidentificação

A arquitetura dos algoritmos de reidentificação pode variar conforme a abordagem proposta, mas na grande maioria dos casos há uma estrutura comum, principalmente quando utilizados modelos de redes neurais.

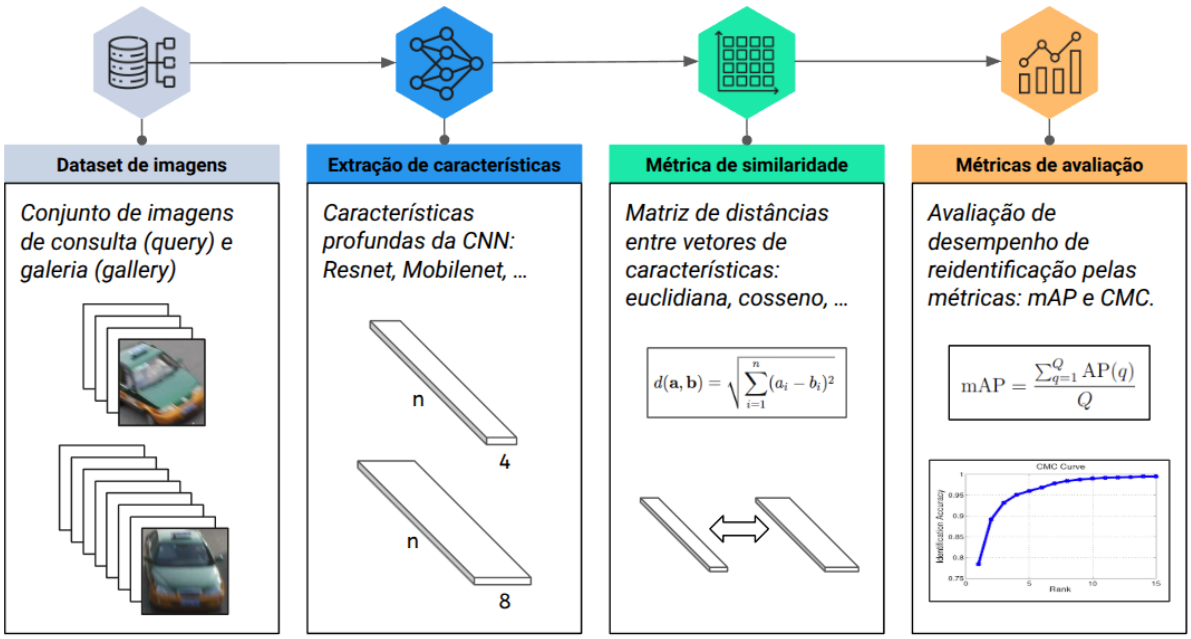
3.1.1 Reidentificação com Modelos de Redes Neurais

As redes neurais profundas trouxeram para o campo da visão computacional grandes melhorias para resolver problemas de precisão na identificação de objetos, assim como abriram caminho para novos desafios à medida que essa identificação se torna mais abrangente, ou seja, é capaz de reconhecer diversos padrões complexos.

A abordagem aqui descrita refere-se a modelos de redes neurais gerados a partir de treinamento supervisionado com base na aplicação alvo. Então, nesse caso, sempre será necessário realizar o processo de treinamento do modelo para que este se adeque ao problema em questão. Após o treinamento, o *pipeline* de inferência do modelo, acoplado à estrutura de reidentificação, costuma seguir a arquitetura exemplificada na Figura 25.

O modelo em si, treinado no *dataset* correspondente à situação desejada, através da rede neural de arquitetura escolhida, deve ser capaz de extrair características discriminativas sobre os objetos em vista. É um processo que ocorre tanto para as imagens da galeria quanto para as imagens de consulta. Com o vetor de características formado, esse conjunto de dados extraídos é comparado por métrica de similaridade, com o objetivo de gerar as distâncias equivalentes para cada amostragem, assim sendo possível associar imagens provavelmente correspondentes para formar um *ranking*, onde o primeiro colocado é o

Figura 25 – *Pipeline* de reidentificação com CNN.



Fonte: Próprio autor.

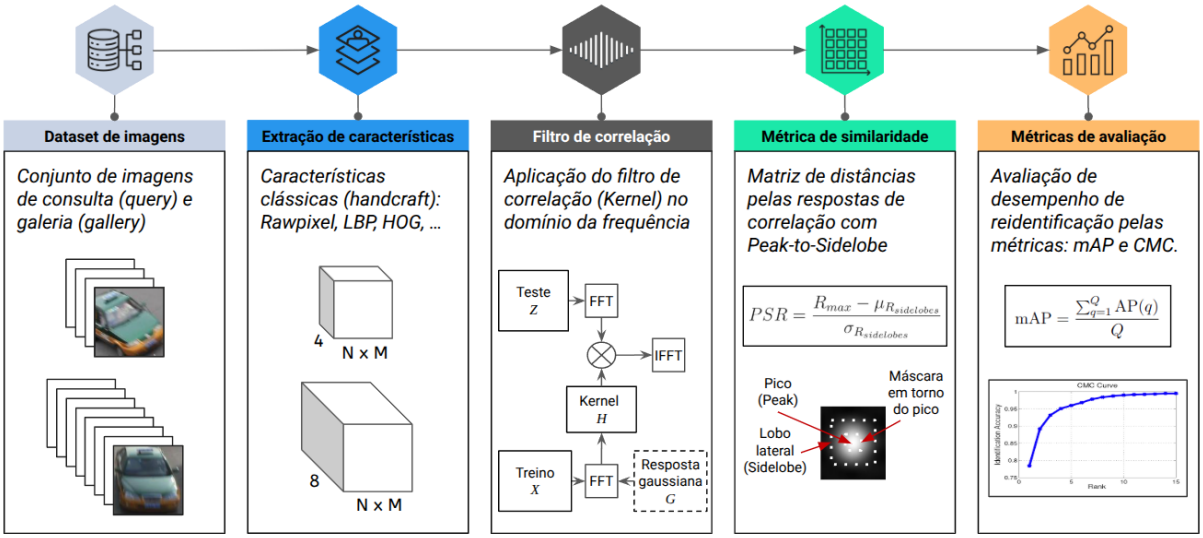
objeto da galeria com identidade mais próxima do objeto de consulta. O desempenho de todo esse processo, quando em ambiente experimental, pode ser medido através de métricas de reidentificação como CMC e mAP.

3.1.2 Reidentificação com Filtro de Correlação Discriminativo

Os algoritmos associados ao tema filtro de correlação não seguem por padrão o *pipeline* descrito sobre redes neurais, justamente por não serem tratados como um problema de reidentificação e sim rastreamento visual de objetos (VOT). Porém, como o objetivo deste trabalho é analisar o funcionamento desta abordagem focada na reidentificação, a arquitetura previamente mencionada para redes neurais será utilizada como base para fins de desenvolvimento e comparação. Dessa forma, o filtro de correlação deve ser ajustado para seguir o mesmo fluxo de processamento.

Portanto, como demonstra a imagem da Figura 26, a arquitetura do formato ajustado para filtro de correlação discriminativo equipara-se às etapas de extração de características, comparação entre as imagens de consulta e da galeria, assim como a medição através de métrica de similaridade. Ao final, é possível avaliar o desempenho através das métricas de reidentificação.

Figura 26 – *Pipeline* de reidentificação com DCF.



Fonte: Próprio autor.

3.1.3 Principais diferenças

Como ilustrado, um dos grandes diferenciais da arquitetura proposta em relação a redes neurais, além do processamento no domínio da frequência, é o treinamento *online*, ou, em tempo de execução, para criar modelos discriminativos de cada objeto, sendo assim mais flexível a qualquer cenário, porém adicionando etapas ao processamento. A Tabela 2 mostra as principais diferenças entre essas duas abordagens em cada etapa de processamento.

Tabela 2 – Comparação entre arquitetura de reidentificação com Rede Neural Convolucional e Filtro de Correlação Discriminativo.

Processos de reidentificação	**Rede Neural Convolucional**	**Filtro de Correlação Discriminativo**
Treinamento	Offline, supervisionado	Online, supervisionado
Extração de características	CNNs (ex.: Mobilenet, Resnet)	Handcraft (ex.: Rawpixel, HOG)
Processo adicional	-	Treina modelo, aplica filtro, calcula correlação (no domínio da frequência)
Métrica de similaridade	Distância (ex.: Euclidean, Cosine)	Peak to Sidelobe Ratio (PSR)

3.2 Desenvolvimento do Filtro de Correlação Discriminativo

Além de adequar a arquitetura de processamento, utilizando filtro de correlação discriminativo, para que sirva de forma equivalente à reidentificação de objetos, é necessário trabalhar as etapas do *pipeline* definido.

3.2.1 Escolha da Abordagem

Durante a última década, muitas propostas foram desenvolvidas para a aplicação de filtro de correlação. Os marcos históricos que foram pioneiros e agregaram notórias melhorias, como MOSSE de Bolme et al. (2010) e KCF de Henriques et al. (2015), fornecem uma boa base de trabalho pela eficácia, sem adicionar muitas camadas complexas. Portanto, o KCF, um passo adiante em relação ao MOSSE, foi a abordagem escolhida para inserção no *pipeline* deste trabalho, pois dispõe das etapas de treino, detecção e correlação de forma bem definida e delimitada.

O objetivo deste trabalho não é buscar o melhor algoritmo proposto para filtro de correlação, mas sim aquele que possa ser executado para fins de reidentificação e não apenas rastreamento, como costuma ser usado, trazendo assim uma nova aplicação para o filtro de correlação. Futuramente, a abordagem proposta pode ser melhorada conforme as abordagens mais recentes ou até mesmo se propor algum conceito novo. O mesmo é válido para a etapa de extração de características, focando em técnicas clássicas já conhecidas e bem estabelecidas em visão computacional.

3.2.2 Extração de Características

Visando testar diferentes extratores de características para análise da resposta final, foram utilizadas as abordagens demonstradas na Figura 27.

Figura 27 – Extração de características clássica.

Fonte: Próprio autor.

Onde *Rawpixel* e *Color Name* mantêm as características visuais, incluindo informação dos canais de cores. Já HOG (*Histogram of Oriented Gradients*), LBP (*Local Binary Patterns*) e saliência ressaltam as características de borda e suas intensidades.

A versão *Color Name* aplicada é uma variante criada pelo autor, onde o *range* de cores não é definido por uma atribuição prévia de nomes de cores e sim pela distribuição dos níveis de cada canal em uma escala menor. Assim, mapeiam-se, para cada canal RGB, os 256 valores de intensidade possíveis, para uma quantidade limitada e menor, no caso, foi escolhida a escala de 5 valores por canal, por manter as principais características de cores sem perder a distinção entre os objetos.

Com essa discretização, o espaço de cor RGB é reduzido para apenas 125 combinações possíveis, em contraste com o método *Rawpixel*, que utiliza o espaço RGB completo, com 16.777.216 variações possíveis. Essa simplificação pode contribuir para a redução de ruídos indesejados e manter somente a informação de cor mais relevante de cada objeto presente na imagem.

O método LBP será aplicado ao conjunto dos 3 canais RGB, com a combinação dos canais processados separadamente, para aproveitar a gama de informações provenientes de cada matriz.

A técnica adotada para extração de características visuais salientes foi a chamada *Fine Grained Saliency*, proposta por Montabone e Soto (2010), por apresentar uma resposta satisfatória para o ressalto de saliências. A técnica está disponível e implementada na biblioteca OpenCV[1].

3.2.3 Filtro de Correlação Kernelizado Modificado

A lógica principal implementada por Henriques et al. (2015), no Algoritmo 1, como explicado na Seção 2.5, foi tomada como base no desenvolvimento do filtro de correlação usado neste trabalho, através das funcionalidades de construção do *kernel*, treinamento e detecção, mantendo-se intacta a formulação original, com algumas ressalvas.

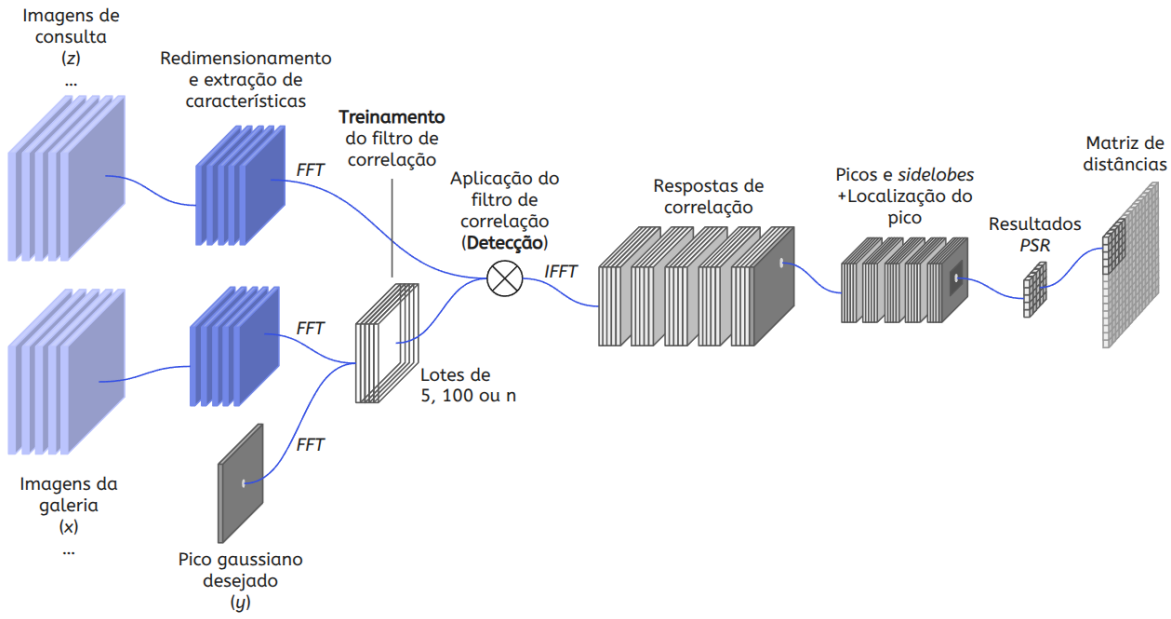
Uma das principais diferenças nesta implementação, em relação ao algoritmo KCF original, é a não utilização do *pipeline* de rastreamento com atualização após o processamento de cada *frame*. Esse procedimento, que normalmente realiza uma interpolação entre as autocorrelações atual e anterior para atualizar a imagem de referência utilizada no rastreamento, foi propositalmente omitido. A justificativa para essa escolha é que o foco da presente abordagem está na reidentificação, sem considerar a continuidade temporal entre os quadros. Dessa forma, elimina-se uma etapa do processamento, tornando a solução mais

[1] Site oficial do OpenCV: https://opencv.org/

eficiente para o objetivo proposto.

Outra funcionalidade que merece destaque e compete com estruturas de processamento com modelos de redes neurais, foi a modificação para realizar operações envolvendo a transformada rápida de Fourier em lotes (*batches*), para obter maior eficiência. Para tal, é necessário que todas as imagens de entrada possuam a mesma escala de resolução. Essa e outras partes do fluxo de processamento podem ser melhor compreendidas analisando-se a representação do *pipeline* proposto ilustrado na Figura 28.

Figura 28 – Fluxo de reidentificação com Filtro de Correlação Kernelizado modificado para processamento em lotes.



Fonte: Próprio autor.

A construção do *kernel* baseia-se em calcular a correlação cruzada entre dois *patches* de imagem, $\mathbf{x}_1$ e $\mathbf{x}_2$, de dimensões $m \times n \times c$, onde c é o número de canais (e.g., RGB). A correlação no domínio da frequência é obtida por:

$$C = \mathcal{F}^{-1}\left(\sum_{i=1}^{c} \overline{\mathcal{F}(\mathbf{x}_1^i)} \odot \mathcal{F}(\mathbf{x}_2^i)\right) \tag{3.1}$$

Em seguida, calcula-se a matriz de distâncias quadráticas, ou seja, o custo da diferença entre as imagens comparadas:

$$D = \sum_{i=1}^{c} \|\mathbf{x}_1^i\|^2 + \sum_{i=1}^{c} \|\mathbf{x}_2^i\|^2 - 2 \cdot \text{fftshift}(C) \tag{3.2}$$

Nesse contexto, $\|\mathbf{x}^i\|^2$ corresponde à soma dos quadrados dos valores no canal i do *patch* $\mathbf{x}$, e fftshift(C) reposiciona o pico de correlação para o centro da matriz C.

Com isso, o *kernel* gaussiano é definido como:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{1}{\sigma^2} \cdot \frac{|D|}{N}\right) \tag{3.3}$$

onde σ é um parâmetro de controle do *kernel* e $N = m \cdot n$ é o número total de elementos em D. O parâmetro σ controla a largura do *kernel* gaussiano, determinando a sensibilidade da função de similaridade às diferenças entre os vetores de entrada. Valores menores de σ tornam o *kernel* mais seletivo, enquanto valores maiores suavizam essa seletividade.

Durante o treinamento, define-se o vetor de resposta desejada y, através de um pico gaussiano centrado, e calcula-se sua transformada $\widehat{y} = \mathcal{F}(y)$. O *kernel* de autocorrelação $k(\mathbf{x}, \mathbf{x})$, da imagem da galeria, é então transformado e usado para obter os coeficientes do filtro no domínio da frequência:

$$\widehat{\alpha}_f = \frac{\widehat{y}}{\mathcal{F}(k(\mathbf{x}, \mathbf{x}))} \tag{3.4}$$

Na fase de detecção, dado um novo *patch* $\mathbf{z}$, referente à imagem de consulta, calcula-se o *kernel* $k(\mathbf{z}, \mathbf{x})$ e sua transformada $\widehat{k}_{zx}$, para então obter a resposta de correlação via:

$$f(\mathbf{z}) = \mathcal{F}^{-1}\left(\widehat{\alpha}_f \odot \widehat{k}_{zx}\right) \tag{3.5}$$

A partir da resposta de correlação $f(\mathbf{z})$, é possível avaliar tanto a localização do pico, que indica a posição estimada do objeto no novo *patch*, quanto a presença de ruído no entorno, refletindo a qualidade da detecção, que, no caso, se refere à similaridade entre as imagens comparadas.

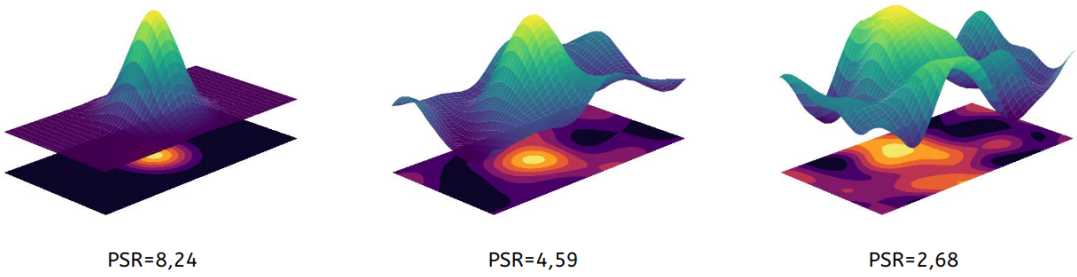
3.2.4 Métrica de Similaridade PSR

A medição da resposta gerada pelo filtro de correlação aplicado à imagem amostrada deve ser inferida através de métrica compatível com a abordagem específica. Para isso, *Peak to Sidelobe Ratio* está mais presente na literatura, principalmente quando se espera um pico gaussiano (ADEBAYO, 2013).

Como pode ser observado na Figura 29, um PSR elevado indica a proximidade das frequências correlacionadas, ou seja, as imagens que foram usadas na correlação são muito

parecidas. À medida que essa resposta se torna mais ruidosa, a similaridade se torna menor, com PSR baixo.

Figura 29 – Respostas de correlação em visualização 3D com os respectivos valores de PSR.



Fonte: Próprio autor.

Da mesma forma que a etapa de aplicação do filtro de correlação foi modificada para realizar o cálculo em lotes, o cálculo da métrica de similaridade, efetuado sobre as respostas de correlação, também foi alterado para computar os resultados em lotes, como é possível observar na Figura 28, anteriormente apresentada.

Com isso, a construção da matriz de distâncias de similaridade é realizada a partir dos valores de PSR, aplicados na Equação 3.6. O cálculo consiste na relação entre o valor produzido pela resposta de correlação entre as amostras de consulta e galeria, e o valor obtido para o PSR ideal, com um pico gaussiano sem ruídos.

$$\text{dist} = 1 - \frac{PSR_{resposta}}{PSR_{gaussiano}} \tag{3.6}$$

3.3 Experimentos

Nesta seção será descrito o processo de experimentação para comparação entre as abordagens de reidentificação com modelos de redes neurais e o filtro de correlação proposto, com diferentes *datasets* e configurações.

3.3.1 Recursos Utilizados

Em ambas as arquiteturas propostas, há pelo menos alguma otimização realizada em GPU, portanto, foi necessário hardware compatível com GPU dedicada. Para os testes variados que foram realizados, o hardware utilizado foi um *notebook* com CPU Intel Core i7-10750H e GPU NVIDIA RTX 2060.

Para o processamento em lotes, as bibliotecas da linguagem de programação Python utilizadas foram Numpy, CuPy e PyTorch, sendo Numpy com otimizações em CPU para operações matriciais em lote, enquanto CuPy e PyTorch para processamento em GPU. Ambas utilizam no *backend* de processamento o módulo CuFFT da NVIDIA, desenvolvido especificamente para as demandas que aplicam a Transformada Rápida de Fourier (FFT) com aceleração em GPU, cada uma com sua implementação. Ainda assim, as duas são muito similares em termos de chamadas no código, para fins de padronização, mas as otimizações internas intrínsecas de cada uma podem influenciar no resultado final.

Através do trabalho desenvolvido por Zhou e Xiang (2019), uma biblioteca foi disponibilizada para rodar de forma fácil o processo de treinamento e inferência de modelos de reidentificação de pessoas em alguns *datasets* conhecidos na literatura. Dessa forma, a biblioteca Torchreid foi utilizada juntamente com funcionalidades próprias, desenvolvidas na linguagem de programação Python, para este processo de reidentificação, justamente pela facilidade de preparação e execução durante o experimento.

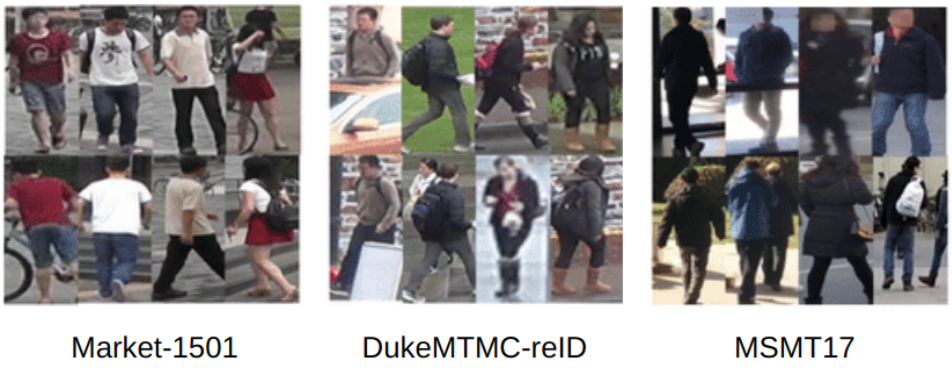
Em relação aos conjuntos de dados, foram selecionados alguns *datasets* com foco em reidentificação de pessoas e veículos. As amostras de imagens fornecidas pelos conjuntos Market1501 (ZHENG et al., 2015) e CityFlowV2-ReID (LUO et al., 2021a) foram utilizadas para teste. Alguns modelos de reidentificação de pessoas já são disponibilizados pela biblioteca Torchreid. Esses modelos foram treinados nos conjuntos de dados Market1501, DukeMTMC-reID (RISTANI et al., 2016b) e MSMT17 (WEI et al., 2018). Já para reidentificação de veículos, como a biblioteca não fornece um modelo pré-treinado, foi utilizado o *dataset* VRIC (KANACI; ZHU; GONG, 2018) para treinar e encontrar os parâmetros para identificação dos padrões referentes a veículos.

3.3.2 Preparação dos Conjuntos de Dados

O *dataset* Market1501 foi formado por diversas imagens de pessoas capturadas por 6 câmeras postas em ambiente externo, em frente a um supermercado na Universidade Tsinghua, em Pequim, na China (ZHENG et al., 2015). As imagens correspondem a 1.501 indivíduos, constituindo o *dataset* com 3.368 imagens de consulta e 19.732 na galeria para teste.

Os *datasets* DukeMTMC-reID (RISTANI et al., 2016b) e MSMT17 (WEI et al., 2018) foram utilizados no treinamento dos modelos disponíveis na plataforma da biblioteca Torchreid, mas não foram utilizados para teste, pois com o conjunto Market1501 já foi possível testar o modelo pré-treinado na mesma base de dados e em relação a diferentes *datasets*. Esse tipo de teste foi denominado por Zheng et al. (2015) de teste no mesmo domínio (*same-domain*) e domínio cruzado (*cross-domain*) respectivamente. A Figura 30 demonstra algumas amostras de cada conjunto mencionado para reidentificação de pessoas.

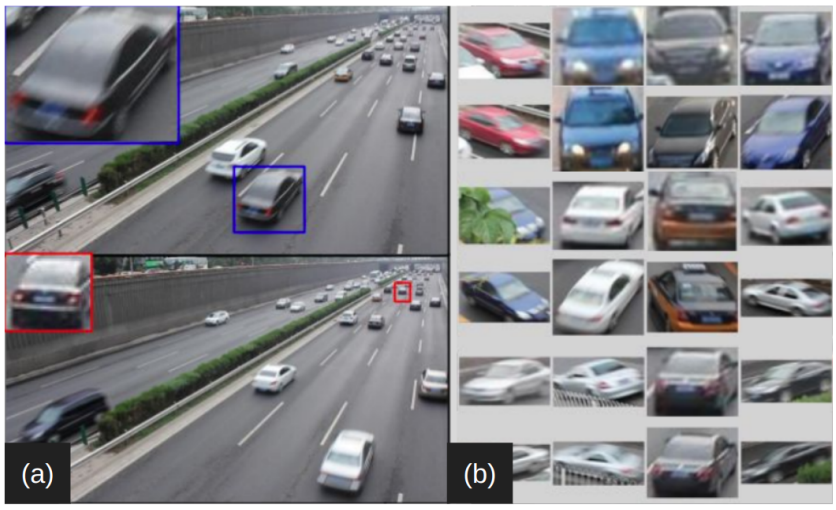
Figura 30 – *Datasets* para reidentificação de pessoas.



Fonte: Ding, Duan e Li (2022). Editado pelo autor.

Para realizar o treinamento do modelo de reidentificação para veículos, foi utilizado o *dataset* VRIC, o qual é uma derivação do conjunto de dados UA-DETRAC (WEN et al., 2020), voltado para os desafios de detecção e rastreamento de objetos, sendo, portanto, convertido para o desafio de reidentificação, como exibido na Figura 31, pelas amostras de imagens extraídas. Como mencionado por Kanaci, Zhu e Gong (2018), o contexto desse *dataset* é trazer um conjunto mais realista se comparado a outros conjuntos anteriormente propostos, contendo variações presentes em ambiente real como resoluções diferentes, imagens borradas, variações na iluminação, oclusão e ponto de vista. Os dados são constituídos de 60.430 imagens de 5.622 veículos, capturados por 60 diferentes câmeras de monitoramento de tráfego, tanto em horários de dia quanto à noite. A quantia usada para treinamento foi de 54.808 e para teste 5.622, que formam, conforme configurada no experimento, a distribuição para os conjuntos de consulta e galeria.

Figura 31 – *Dataset* VRIC. (a) Captura das amostras de imagens dos veículos. (b) Pares de imagens para uso em reidentificação.

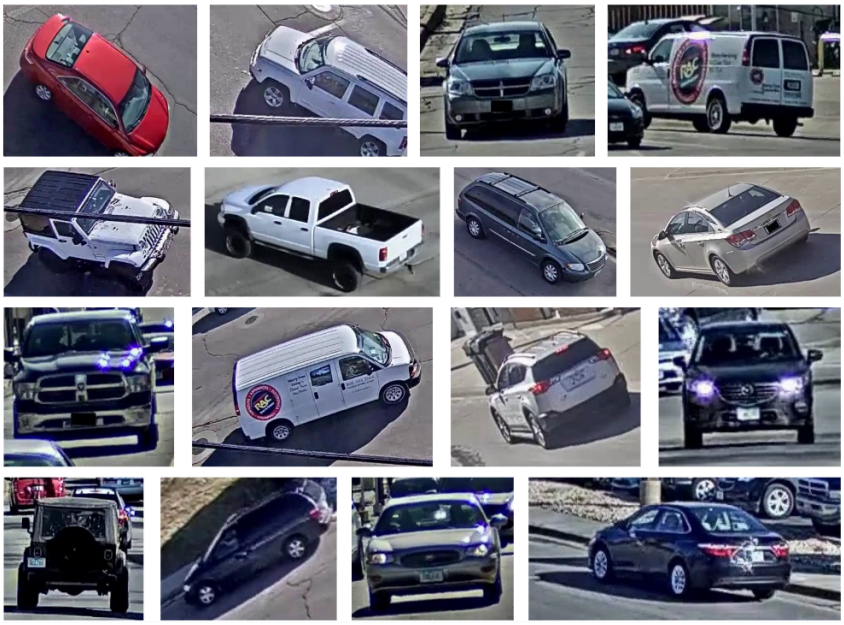


Fonte: Kanaci, Zhu e Gong (2018). Editado pelo autor.

A título informativo, o *dataset* UA-DETRAC foi disponibilizado através do ambiente *AI City Challenge*, patrocinado pela NVIDIA. No site constam diversas competições anuais sobre os desafios em cidades inteligentes dentro dos campos da inteligência artificial, com destaque principalmente em problemas de visão computacional (NAPHADE et al., 2017).

Para teste do modelo treinado para veículos, o *dataset* CityFlowV2-ReID foi utilizado, um conjunto proposto por Tang et al. (2019) para problemas de rastreamento e reidentificação de veículos em múltiplas câmeras, como sugerem as imagens da Figura 32 que compõem o *dataset*, disponibilizado também através dos desafios *AI City Challenge* da NVIDIA. Conforme comentado por Luo et al. (2021a), o trabalho contribui para soluções em sistemas de transporte inteligente (ITS). O *dataset* possui 85.058 imagens extraídas da captura de 46 câmeras de monitoramento de tráfego, dessas, 1.103 servem para consulta e 31.238 para teste.

Figura 32 – *Dataset* CityFlowV2-ReID.



Fonte: Próprio autor.

3.3.3 Escolha dos Modelos Pré-Treinados

Baseando-se na Tabela 3, dos resultados obtidos para alguns modelos pré-treinados e disponibilizados pela biblioteca Torchreid, com foco na reidentificação de pessoas, pode-se ver o desempenho entre demais arquiteturas e o modelo proposto por Zhou et al. (2019) em diferentes configurações.

É possível observar que os modelos OSNet apresentam um bom equilíbrio entre seu desempenho nos *datasets* mencionados e o custo computacional mais baixo. Portanto, para o experimento em questão, foi selecionado o modelo OSNet x0_25, que representa a versão

Tabela 3 – Comparação dos modelos pré-treinados na biblioteca Torchreid, no mesmo domínio dos respectivos *datasets*. Distância euclidiana utilizada como métrica de distância para todos.

Modelo	**# Parâmetros (10^6)**	**GFLOPs**	**Resolução de entrada**	**CMC Rank-1 (mAP)**		
				Market1501	**DukeMTMC-reID**	**MSMT17**
ResNet50	23.5	2.7	(256, 128)	87.9 (70.4)	78.3 (58.9)	63.2 (33.9)
ResNet50_fc512	24.6	4.1	(256, 128)	90.8 (75.3)	81.0 (64.0)	69.6 (38.4)
MLFN	32.5	2.8	(256, 128)	90.1 (74.3)	81.1 (63.2)	66.4 (37.2)
HACNN*	4.5	0.5	(160, 64)	90.9 (75.6)	80.1 (63.2)	64.7 (37.2)
MobileNetV2_x1_0	2.2	0.2	(256, 128)	85.6 (67.3)	74.2 (54.7)	57.4 (29.3)
MobileNetV2_x1_4	4.3	0.4	(256, 128)	87.0 (68.5)	76.2 (55.8)	60.1 (31.5)
OSNet_x1_0	2.2	0.98	(256, 128)	94.2 (82.6)	87.0 (70.2)	74.9 (43.8)
OSNet_x0_75	1.3	0.57	(256, 128)	93.7 (81.2)	85.8 (69.8)	72.8 (41.4)
OSNet_x0_5	0.6	0.27	(256, 128)	92.5 (79.8)	85.1 (67.4)	69.7 (37.5)
OSNet_x0_25	0.2	0.08	(256, 128)	91.2 (75.0)	82.0 (61.4)	61.4 (29.5)

compacta do modelo OSNet x1_0 que possui 2,2 milhões de parâmetros, gerada com o coeficiente multiplicador β=0,25, utilizado para reduzir a largura dos canais da rede e, consequentemente, a quantidade total de parâmetros. Na comparação entre os modelos pré-treinados, é a opção mais leve em termos de quantidade de parâmetros e operações matemáticas em ponto flutuante (FLOPs), ainda assim apresentando bons resultados.

3.3.4 Treinamento de Modelo para Reidentificação de Veículos

Para realizar as comparações de reidentificação em um *dataset* voltado ao tráfego de veículos, é necessário dispor de um modelo treinado especificamente para esse objetivo. No entanto, a biblioteca Torchreid não fornece modelos voltados para veículos, uma vez que seu foco principal é a reidentificação de pessoas. Apesar disso, as arquiteturas disponibilizadas pela biblioteca podem ser empregadas nessa tarefa, desde que treinadas adequadamente com dados compatíveis.

Para garantir a consistência igualmente demonstrada pelos modelos pré-treinados nos *datasets* de pessoas, apresentados na Tabela 3, o modelo voltado para veículos deve seguir o mesmo padrão de configuração no treinamento. A resolução de cada imagem de veículo pode variar bastante, mas para fins de padronização seguiu-se um formato de proporções iguais, com base na resolução utilizada pelos modelos pré-treinados na OSNet, definido então para 128x128 após o redimensionamento, tanto para treinamento quanto teste em *datasets* de veículos. Optou-se por manter a mesma resolução usada no treinamento para

carregar o modelo durante os testes, para evitar distorções ao padrão aprendido pelo modelo.

A arquitetura da rede utilizada foi a OSNet, na configuração x0_25, devido ao seu equilíbrio entre precisão e desempenho de processamento. Os hiperparâmetros relevantes para o treinamento incluem o otimizador AMSGrad, com uma taxa de aprendizado de 0.003, e a função de perda Softmax, seguindo a mesma configuração de treinamento dos modelos pré-treinados já mencionados.

3.3.5 Roteiro de Testes

O experimento para realização da reidentificação utilizando as arquiteturas propostas foi feito através do *framework* Torchreid, utilizando como entrada os *datasets* Market1501 e CityFlowV2-ReID.

Foram calculadas as métricas *Mean Average Precision* (mAP) e *Cumulative Matching Characteristics* (CMC) para reidentificação, além do tempo decorrido em segundos durante a avaliação em cada teste.

Foi colocada em prova a reidentificação em única câmera e em múltiplas câmeras, já que os *datasets* mencionados disponibilizam as imagens em múltiplos pontos de vista. Dentre as várias configurações possíveis, os resultados exibidos nas tabelas da Seção 3.4 correspondem aos 10 melhores resultados ordenados pelo índice mAP.

A seguir serão descritas as configurações definidas para os experimentos de reidentificação usando as duas arquiteturas mencionadas, DCF e OSNet, explicitando-se os hiperparâmetros de teste envolvendo extração de características, os *datasets* em que os modelos foram treinados e as definições para cálculo de similaridade.

3.3.5.1 Configuração DCF

- Arquitetura do modelo utilizado: Filtro de Correlação Discriminativo para Reidentificação, DCF-ReID.
- Metodologias para extração de características: *Rawpixel*, *Color Name*, HOG, LBP e *Saliency*.
- Combinação em dupla de alguns extratores de características.
- Tamanho do *patch* base em pixels: 68x132, 36x68, 20x36.
- Tamanho do lote de processamento: 100 imagens por iteração.
- Métrica de similaridade: *Peak-to-Sidelobe Ratio* (PSR).

- Máscara para cálculo do PSR: 1.0x0.2 (100% e 20%) da resolução do mapa de correlação.

3.3.5.2 Configuração OSNet

- Arquitetura do modelo utilizado: *Omni-Scale Network*, *OSNet-x0_25*.
- Modelos pré-treinados em *datasets* para reidentificação de pessoas: Market1501, DukeMTMC-reID, MSMT17 e Imagenet.
- Modelo pré-treinado em *dataset* para reidentificação de veículos: VRIC.
- Tamanho do *patch* base em pixels: 128x256 (pessoas), 128x128 (veículos).
- Tamanho do lote de processamento: 100 imagens por iteração.
- Métrica de similaridade: Distância euclidiana (*Euclidean*).

3.4 Resultados

3.4.1 Resultados para Reidentificação de Pessoas

Primeiramente, os testes realizados no *dataset* Market1501 foram feitos através do conjunto de imagens de consulta em comparação às imagens referentes a apenas uma câmera, produzindo os resultados da Tabela 4.

Tabela 4 – Resultados de reidentificação no *dataset* Market1501 em 1 câmera, para 66 consultas em uma galeria de 2.738 imagens. Contém a descrição do *dataset* utilizado no treinamento, *features* e tamanho do *patch*.

Abordagem modelo	**Descrição**	**Métrica**	**mAP↑**	**Rank-1↑**	**Rank-5↑**	**Rank-10↑**	**Tempo decorrido (seg.)↓**
OSNet-x0_25	Market1501, 128x256	Euclidean	**0,90**	**0,97**	**0,98**	**0,98**	**4,24**
OSNet-x0_25	MSMT17, 128x256	Euclidean	0,80	**0,97**	**0,98**	**0,98**	4,29
OSNet-x0_25	DukeMTMC-reID, 128x256	Euclidean	0,79	**0,97**	**0,98**	**0,98**	4,34
DCF-ReID	Colorname+Saliency, 36x68	PSR	0,53	0,88	0,97	0,97	13,19
DCF-ReID	Rawpixel+Saliency, 36x68	PSR	0,53	0,88	0,95	0,95	12,70
DCF-ReID	Colorname+LBP, 36x68	PSR	0,52	0,91	0,95	0,97	13,94
DCF-ReID	Rawpixel+LBP, 36x68	PSR	0,51	0,91	0,95	0,95	14,06
DCF-ReID	Rawpixel, 68x132	PSR	0,50	0,85	0,94	0,97	30,51
DCF-ReID	Rawpixel, 20x36	PSR	0,50	0,85	0,94	0,97	5,60
OSNet-x0_25	Imagenet, 128x256	Euclidean	0,50	0,85	0,92	0,92	4,28

Visivelmente, ao se analisar as respostas produzidas e comparadas na Tabela 4, os modelos pré-treinados apresentam métricas com valor superior à abordagem proposta com DCF. Porém, também é percebido que há um viés quando a avaliação acontece no mesmo domínio, ou seja, o modelo testado no *dataset* Market1501 com melhor resultado é o modelo treinado no mesmo *dataset*, ainda que, durante o treinamento haja a separação entre as imagens de treino e teste.

Também é percebido que o modelo pré-treinado no *dataset* Imagenet apresenta resultados de mAP e CMC similares aos produzidos pela abordagem DCF-ReID. Porém, o tempo de processamento segue sendo melhor nos modelos OSNet.

Para aumentar a complexidade do desafio e avaliar a resposta dos algoritmos a múltiplos pontos de vista, causando maior variação no padrão entre as imagens originais de consulta e as imagens alvo da galeria, o *dataset* Market1501, agora com suporte a múltiplas câmeras, foi utilizado. Os resultados podem ser vistos na Tabela 5.

Tabela 5 – Resultados de reidentificação no *dataset* Market1501 em 6 câmeras, para 336 consultas em uma galeria de 1.591 imagens. Contém a descrição do *dataset* utilizado no treinamento, *features* e tamanho do *patch*.

Abordagem modelo	**Descrição**	**Métrica**	**mAP↑**	**Rank-1↑**	**Rank-5↑**	**Rank-10↑**	**Tempo decorrido (seg.)↓**
OSNet-x0_25	Market1501, 128x256	Euclidean	**0,75**	**0,76**	**0,87**	**0,89**	**3,45**
OSNet-x0_25	MSMT17, 128x256	Euclidean	0,28	0,26	0,45	0,52	3,47
OSNet-x0_25	DukeMTMC-reID, 128x256	Euclidean	0,22	0,22	0,41	0,46	3,52
DCF-ReID	Colorname+Saliency, 36x68	PSR	0,05	0,05	0,10	0,15	26,79
DCF-ReID	Rawpixel+Saliency, 36x68	PSR	0,05	0,03	0,09	0,13	27,18
DCF-ReID	Colorname+LBP, 36x68	PSR	0,04	0,03	0,08	0,12	29,57
DCF-ReID	Colorname, 36x68	PSR	0,04	0,04	0,08	0,13	24,30
DCF-ReID	Rawpixel+LBP, 36x68	PSR	0,04	0,02	0,09	0,12	28,42
OSNet-x0_25	Imagenet, 128x256	Euclidean	0,03	0,02	0,05	0,11	3,70
DCF-ReID	HOG, 36x68	PSR	0,02	0,01	0,05	0,07	99,88

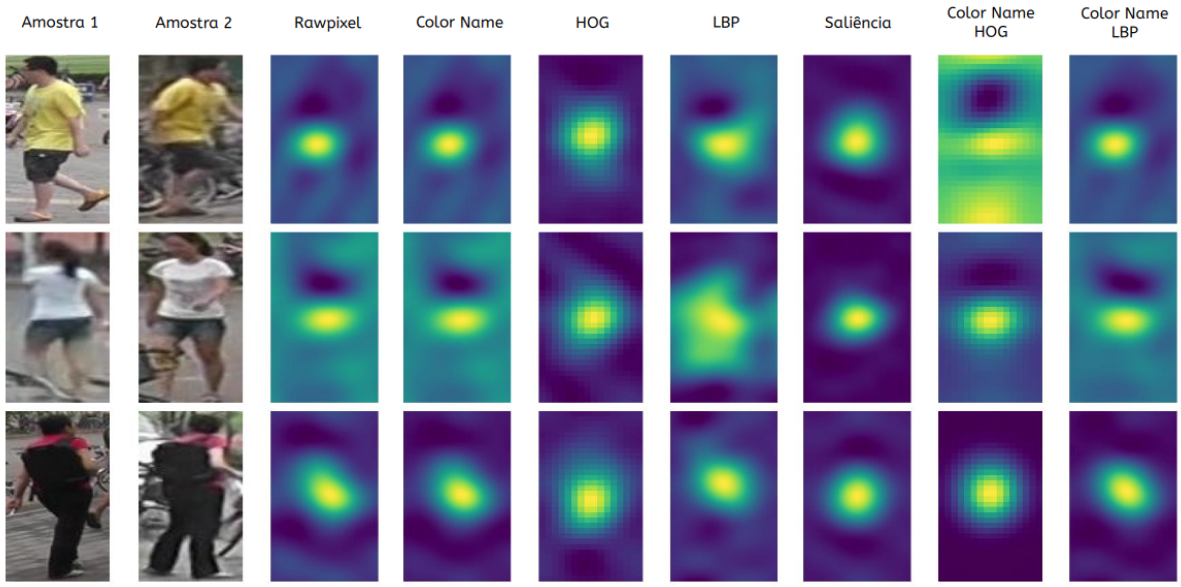
Nestes testes, com foco em reidentificação entre múltiplas câmeras, a metodologia proposta DCF-ReID gera resultados abaixo do esperado se comparados aos modelos de redes neurais, exceto o treinado no *dataset* Imagenet, que apresenta desempenho similar ou até abaixo, algo já percebido no processamento para única câmera.

Vale ainda ressaltar que, ao utilizar a abordagem DCF-ReID, com o extrator de características HOG, o tempo de processamento completo para a avaliação é muito acima dos demais, o que não necessariamente descarta a metodologia, mas pode indicar ineficiência

do algoritmo nesse caso específico, na forma como foi implementada no código através da biblioteca Scipy em Python.

Na Figura 33, é possível observar o mapa de correlação produzido pelo algoritmo DCF, aplicando o *kernel* de correlação entre os pares de imagens amostradas. A primeira figura corresponde às imagens relacionadas, onde a pessoa da consulta é a mesma da amostra da galeria. O pico gaussiano é perceptível na maioria dos casos, exceto na combinação entre *Color Name* e HOG.

Figura 33 – Respostas do DCF por *feature* para imagens relacionadas.



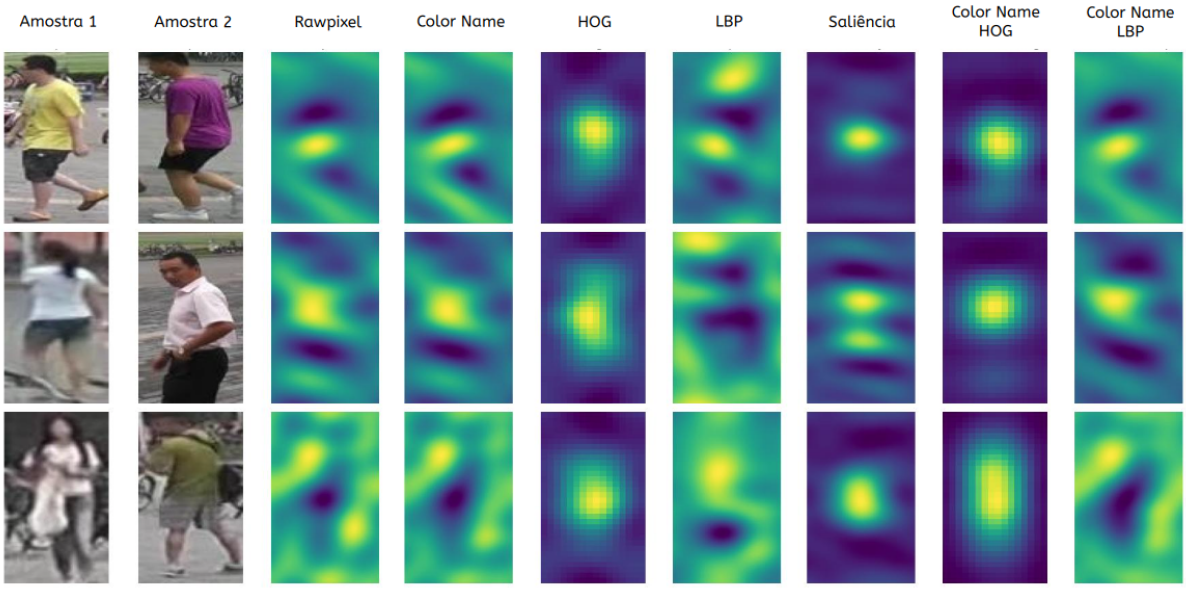
Fonte: Próprio autor.

Já na Figura 34, está a correlação entre imagens não relacionadas, ou seja, a pessoa da consulta não é a mesma da amostra da galeria. É possível notar que, mesmo assim, para os extratores de características HOG e por saliência, o pico gaussiano continua acentuado, o que nesse caso está incorreto. Já nas demais abordagens de extração de características, o mapa produzido possui uma resposta variada e confusa, contendo muito ruído no entorno do pico gaussiano, o que indica a dissimilaridade correta entre as imagens.

3.4.2 Resultados para Reidentificação de Veículos

O mesmo experimento aplicado anteriormente foi feito para o *dataset* CityFlowV2-ReID, primeiramente em única câmera. Os modelos pré-treinados nos *datasets* de reidentificação de pessoas foram mantidos nesse teste, juntamente com o modelo treinado no *dataset* VRIC para veículos, para fins de comparação e análise de desempenho produzido entre modelos treinados e testados em mesma classe e classes distintas, no caso, entre o modelo treinado para pessoas em teste de reidentificação em veículos.

Figura 34 – Respostas do DCF por *feature* para imagens não relacionadas.



Fonte: Próprio autor.

Observa-se que o modelo treinado no *dataset* VRIC apresentou desempenho inferior, conforme mostrado na Tabela 6. Curiosamente, modelos treinados em *datasets* de pessoas obtiveram resultados melhores, levantando dúvidas sobre quais características o modelo realmente aprende e extrai na inferência.

Tabela 6 – Resultados de reidentificação no *dataset* CityFlowV2-ReID em 1 câmera, para 129 consultas em uma galeria de 464 imagens. Contém a descrição do dataset utilizado no treinamento, *features*, tamanho do *patch*.

Abordagem modelo	**Descrição**	**Métrica**	**mAP↑**	**Rank-1↑**	**Rank-5↑**	**Rank-10↑**	**Tempo decorrido (seg.)↓**
OSNet-x0_25	DukeMTMC-reID, 128x256	Euclidean	**0,78**	**0,98**	0,98	0,98	1,44
OSNet-x0_25	Market1501, 128x256	Euclidean	0,73	0,95	**1,00**	**1,00**	**1,40**
OSNet-x0_25	MSMT17, 128x256	Euclidean	0,66	0,93	0,98	0,98	1,49
DCF-ReID	Colorname+LBP, 68x68	PSR	0,57	0,83	0,97	0,99	7,49
DCF-ReID	Rawpixel+LBP, 68x68	PSR	0,56	0,81	0,96	0,98	7,48
OSNet_x0_25	Imagenet, 128x256	Euclidean	0,54	0,81	0,95	0,97	1,50
DCF-ReID	Colorname+Saliency, 68x68	PSR	0,51	0,76	0,91	0,96	6,87
DCF-ReID	HOG, 68x68	PSR	0,51	0,80	0,94	0,96	56,44
DCF-ReID	Rawpixel+Saliency, 68x68	PSR	0,51	0,76	0,91	0,95	6,81
DCF-ReID	LBP, 68x68	PSR	0,49	0,80	0,91	0,94	7,01
OSNet-x0_25	VRIC, 128x128	Euclidean	0,47	0,77	0,91	0,95	1,52

A Tabela 7 apresenta os resultados para o cenário com múltiplas câmeras, que se mostra mais desafiador devido ao maior número de pontos de vista envolvidos na reidentificação.

Tabela 7 – Resultados de reidentificação no *dataset* CityFlowV2-ReID em 40 câmeras, para 527 consultas em uma galeria de 1.845 imagens. Contém a descrição do *dataset* utilizado no treinamento, *features* e tamanho do *patch*.

Abordagem modelo	**Descrição**	**Métrica**	**mAP↑**	**Rank-1↑**	**Rank-5↑**	**Rank-10↑**	**Tempo decorrido (seg.)↓**
OSNet_x0_25	DukeMTMC-reID, 128x256	Euclidean	**0,05**	**0,08**	0,17	**0,23**	7,11
OSNet_x0_25	Market1501, 128x256	Euclidean	0,04	0,07	0,14	0,20	7,06
OSNet_x0_25	MSMT17, 128x256	Euclidean	0,04	0,06	**0,18**	0,22	7,03
OSNet_x0_25	VRIC, 128x128	Euclidean	0,03	0,05	0,11	0,16	**6,93**
OSNet_x0_25	Imagenet, 128x256	Euclidean	0,03	0,06	0,13	0,17	7,56
DCF-ReID	Colorname+LBP, 68x68	PSR	0,02	0,03	0,08	0,13	92,84
DCF-ReID	Colorname+Saliency, 68x68	PSR	0,02	0,02	0,07	0,12	89,40
DCF-ReID	Colorname, 68x68	PSR	0,02	0,02	0,07	0,11	83,85
DCF-ReID	Rawpixel+LBP, 68x68	PSR	0,02	0,03	0,07	0,10	92,41
DCF-ReID	Rawpixel+Saliency, 68x68	PSR	0,02	0,03	0,06	0,11	90,18

Nesse caso, ambas as abordagens não conseguem atender à complexidade encontrada para reidentificação em múltiplas câmeras. Porém, para uma câmera, o DCF-ReID consegue se aproximar de certa forma aos modelos OSNet, sem ter feito qualquer treinamento supervisionado para detecção de pessoas ou veículos.

3.4.3 Análise e Considerações Gerais sobre os Resultados

Nessa análise baseada em testes comparativos entre o modelo OSNet treinado em diferentes conjuntos de dados e a abordagem proposta DCF-ReID, de forma isolada, com foco em reidentificação de objetos em geral, seguindo o conceito de amostras para consulta em uma galeria, é possível obter alguns direcionamentos sobre desempenho.

No geral, os modelos de redes neurais podem oferecer maior precisão dependendo do quão bem os modelos foram treinados para determinada situação, conseguindo também ser mais otimizados para o processamento de grande volume de buscas para as imagens na galeria. Já a reidentificação proposta com DCF não é tão otimizada para grandes volumes, pois necessita aplicar o filtro de correlação para cada combinação, diferente do *pipeline* de redes neurais que apenas calcula a similaridade baseada na distância entre as características extraídas. Conforme o objeto se altera muito da imagem base utilizada, justamente por ser visto de um ponto de vista diferente, outra câmera no caso, o DCF-ReID não consegue encontrar uma boa correlação na maioria dos casos.

O grande diferencial do DCF acaba sendo para buscas na mesma câmera, com o mesmo ponto de vista, sem ter a necessidade de treinar previamente algum modelo, apenas definir uma boa estratégia de extração de características. O algoritmo DCF-ReID também pode ser altamente modificado e otimizado, seja alterando parâmetros internos ou usando diferentes técnicas para extração de características, arquitetura do filtro de correlação e cálculo de métricas, o que abre margem para futuras melhorias.

Os modelos de redes neurais parecem ter uma boa aplicabilidade na tarefa específica de reidentificação quando em múltiplas câmeras ou quando o banco da galeria possui grande volume, o que exige treinamento prévio exaustivo e, mesmo assim, a depender da complexidade e volume de dados, pode ser impreciso. Já a abordagem com DCF pode oferecer um bom custo-benefício, quando se trata de reidentificação local, na mesma câmera, funcionando através de treinamento supervisionado *online*. Entretanto, é importante notar que essa metodologia pode sacrificar um pouco do desempenho computacional se não otimizada.

A análise apresentada no próximo capítulo trará alguns resultados complementares. O objetivo é levantar uma discussão se, na prática, um módulo de reidentificação com o DCF pode servir bem ao objetivo de rastreamento de múltiplos objetos.

4 RASTREAMENTO DE MÚLTIPLOS OBJETOS COM MÓDULO DE REIDENTIFICAÇÃO DCF

Neste capítulo, será descrita a proposta de integração de um módulo de reidentificação baseado no DCF, a partir dos testes apresentados no capítulo anterior. Serão explicadas as escolhas para algoritmos de rastreio existentes e o desenvolvimento do módulo de reidentificação para acoplar ao *pipeline* de rastreamento de múltiplos objetos. Posteriormente, também serão realizados testes comparativos nos *datasets* amplamente utilizados nesse desafio da visão computacional, complementando com *dataset* específico para rastreio de múltiplos veículos. Ao final, deverá ser possível fazer análises e considerações pertinentes sobre a viabilidade da proposta deste trabalho quando o foco é rastreamento de objetos em câmeras de videomonitoramento.

4.1 Algoritmos para Rastreamento de Múltiplos Objetos

Ao longo dos anos, diversas abordagens de rastreio de múltiplos objetos foram propostas, seguindo caminhos diferentes para solucionar problemas específicos. Algumas metodologias seguem uma análise de movimento no espaço 2D da imagem, encontrando a interseção entre objetos para associar sua identificação, assim como utilizam previsões espaciais de posicionamento como recurso auxiliar. Outras técnicas se baseiam em características visuais para realizar essa associação de objetos, os modelos de redes neurais, por exemplo, se encaixam nesse caso, onde a reidentificação é implementada para manter a consistência do rastreio em condições complexas. Essa teoria e outros exemplos podem ser consultados no referencial teórico deste trabalho.

Considerando o contexto em que esse trabalho pretende atuar, ou seja, câmeras de videomonitoramento, é essencial que haja um equilíbrio nas metodologias propostas, a fim de atender a um bom desempenho de acurácia e velocidade de processamento. Isso porque a intenção é que este trabalho seja implementado em aplicações de processamento em tempo real ou quase tempo real.

O algoritmo SORT proposto por Bewley et al. (2016), foi pioneiro em oferecer uma abordagem simples, leve e eficaz para esse tipo de aplicação, porém o trabalho proposto para o rastreio com ByteTrack de Zhang et al. (2022), que funciona de forma similar, aprimora a precisão dos resultados, ao aproveitar todas as detecções fornecidas pelo detector de objetos, sem ignorar mesmo aquelas de confiança baixa. Em comparativos, essa abordagem tem se destacado por manter um padrão de bons resultados se tratando de acurácia a um custo computacional baixo, superando até modelos complexos de redes

neurais específicos para rastreamento (ZHANG et al., 2022). Portanto, a abordagem ByteTrack foi escolhida como base para a integração com a proposta de reidentificação deste trabalho.

4.2 Desenvolvimento do Módulo de Reidentificação DCF

Para adequar o código do DCF-ReID e encapsulá-lo em um módulo acoplável ao algoritmo de rastreamento ByteTrack, foram necessárias algumas modificações nas etapas de inicialização, extração de características e cálculo de similaridade entre os objetos rastreados.

Como o DCF-ReID já havia sido previamente adaptado para operar no mesmo *pipeline* do OSNet durante os testes de reidentificação, o processo inverso também foi realizado, visando criar um módulo de reidentificação baseado no OSNet para integração com o ByteTrack. Dessa forma, para possibilitar uma comparação justa entre os dois modelos integrados ao rastreamento, o OSNet foi ajustado para incorporar as mesmas funcionalidades desenvolvidas para o módulo do DCF-ReID, diferenciando-se apenas pela abordagem base utilizada.

4.3 Integração entre ByteTrack e Módulo de Reidentificação DCF

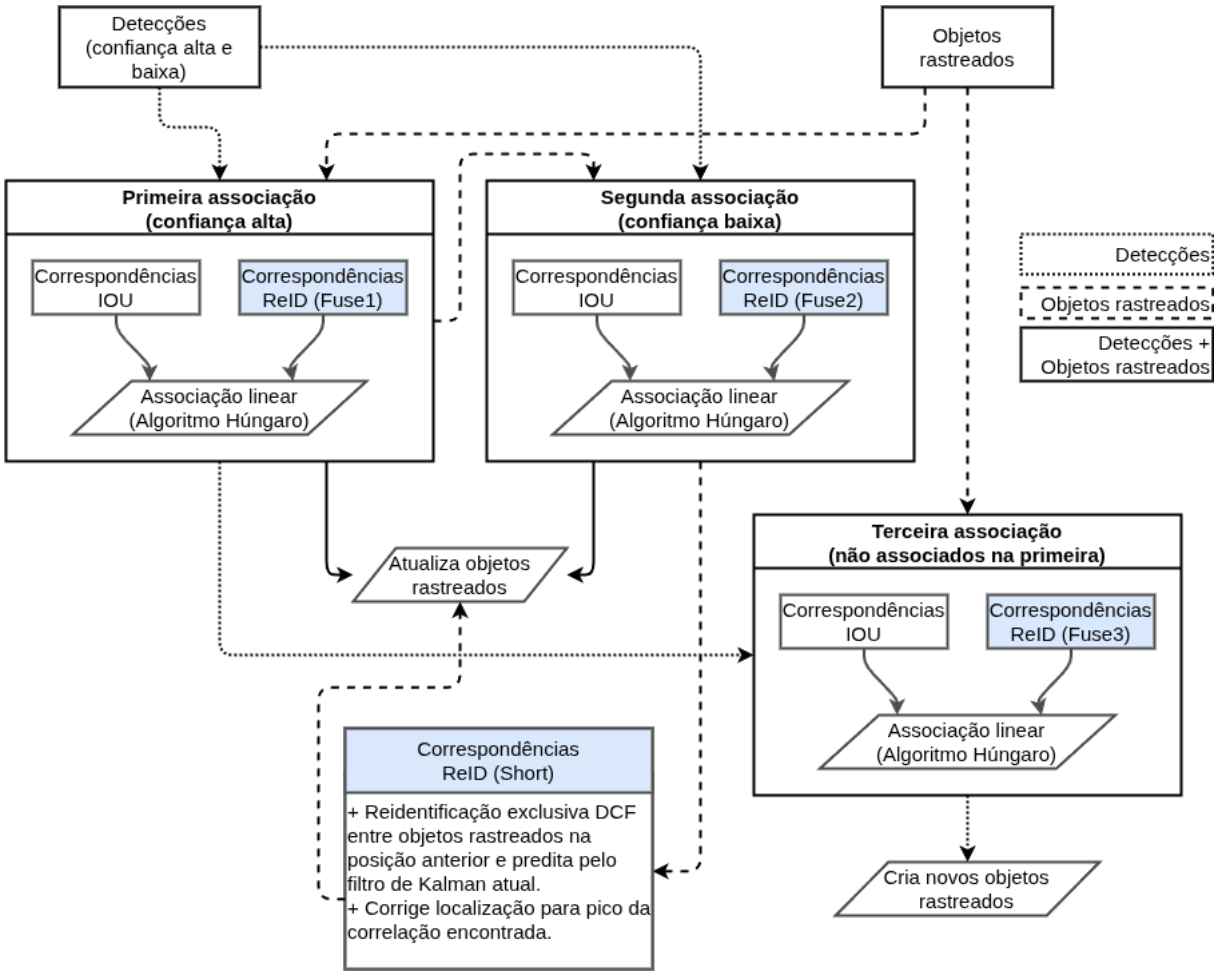
Ao analisar a arquitetura do ByteTrack, com base no diagrama mapeado e apresentado no Apêndice A, foi possível identificar pontos específicos no processo de associação entre detecções e objetos já rastreados que permitem a integração com outros métodos. Destacam-se situações em que há espaço para combinar as associações geradas pela predição do filtro de Kalman e pelo cálculo da Interseção sobre a União (IoU) com métricas adicionais, como a similaridade obtida por reidentificação ou ainda a manutenção de rastreio para objetos já rastreados e sem detecção associada no *frame* processado.

Ao analisar a arquitetura do ByteTrack, com base no diagrama mapeado e apresentado no Apêndice A, foi possível identificar pontos específicos no processo de associação entre detecções e objetos já rastreados que permitem a integração com métodos complementares. Destacam-se, em especial, as situações em que é possível combinar as associações geradas pela predição do filtro de Kalman e pelo cálculo da Interseção sobre a União (IoU) com métricas adicionais, como a similaridade visual obtida por reidentificação. Além disso, há oportunidade para aprimorar a manutenção de rastreio nos casos em que objetos já rastreados não possuem detecção correspondente no *frame* atual.

Uma versão resumida da integração do módulo de reidentificação DCF na estrutura do ByteTrack pode ser vista na Figura 35. Nela, é possível compreender, de forma geral, os

princípios de funcionamento do ByteTrack, onde as detecções dos objetos identificados nas imagens de cada *frame* são passadas por três estágios de associação, para atualizar objetos já rastreados com as novas localizações ou criar novos objetos de rastreio. Os blocos da figura que estão em azul representam as etapas de processamento adicionadas pela reidentificação desenvolvida e serão explicados a seguir.

Figura 35 – Integração do módulo de reidentificação DCF na estrutura do ByteTrack. Versão resumida. Versão completa disponível no Apêndice A.



Fonte: Próprio autor.

Após ter uma visão mais clara da arquitetura de rastreio do ByteTrack, foi possível criar algumas definições, como a separação da reidentificação em níveis de alcance, como descrito abaixo pelo autor deste trabalho.

- *Short ReID*: Reidentificação de curto alcance. Usada para objetos rastreados que perderam referência por breve período ou pequena distância, não se modificam tanto em comparação ao seu último estado visto, e que geralmente não foram marcados como perdidos no rastreio ainda.

- *Medium ReID*: Reidentificação de médio alcance. Usada para objetos rastreados que perderam referência há mais tempo ou em distância maior, e que geralmente já foram marcados como perdidos mas não foram removidos do rastreio local.

- *Long ReID*: Reidentificação de longo alcance. Usada para objetos rastreados que foram marcados como perdidos e já removidos do rastreio local, mas que ficam salvos na galeria para reidentificação mais complexa, geralmente em múltiplas câmeras, com probabilidade alta de modificação de estado em relação à última vez visto, com distanciamento temporal e espacial maior. Essa estratégia não será abordada neste trabalho, pois já foi visto nos resultados do Capítulo 3.4, que a reidentificação de longo alcance pode necessitar uma solução mais robusta a mudanças. Além disso, o ByteTrack por si só, funciona apenas para rastreio em única câmera também.

Outra definição determinada durante o desenvolvimento, mas agora especificamente vinculada ao algoritmo do ByteTrack, é a separação das associações feitas através de reidentificação. Como mencionado anteriormente, a metodologia original encontra a relação entre os objetos detectados e já rastreados através de métrica espacial pelo cálculo de IoU apenas. Na tentativa de melhorar a acurácia dessa correspondência, foi implementada a fusão, ou seja, a multiplicação entre os coeficientes de IoU e similaridade encontrada pelo modelo de reidentificação, como o exemplo da Figura 36, que utiliza o DCF para reidentificação entre a imagem de consulta e a galeria para obter as distâncias de similaridade. As etapas de fusão abaixo descrevem o funcionamento e propósito de cada estágio do ByteTrack, como ilustrado anteriormente na Figura 35.

- *Fuse1 ReID*: Fusão entre métrica de reidentificação e IoU, para reidentificação entre objetos detectados de confiança alta e objetos já rastreados, inclusive objetos marcados como perdidos.

- *Fuse2 ReID*: Fusão entre métrica de reidentificação e IoU, para reidentificação entre objetos detectados de confiança baixa e objetos já rastreados, exclusivamente não associados no primeiro estágio (*Fuse1 ReID*).

- *Fuse3 ReID*: Fusão entre métrica de reidentificação e IoU, para reidentificação entre objetos detectados de confiança alta, não associados no primeiro estágio, e objetos rastreados que estão em fase de ativação, necessitando reincidência de detecção nos primeiros frames em que o objeto é detectado, para maior confiabilidade ao iniciar o rastreio ativo desse objeto.

As associações feitas por fusão podem se enquadrar em reidentificações de curto (*Short*) e médio (*Medium*) alcance e, portanto, esses termos de alcance não serão utilizados nesses

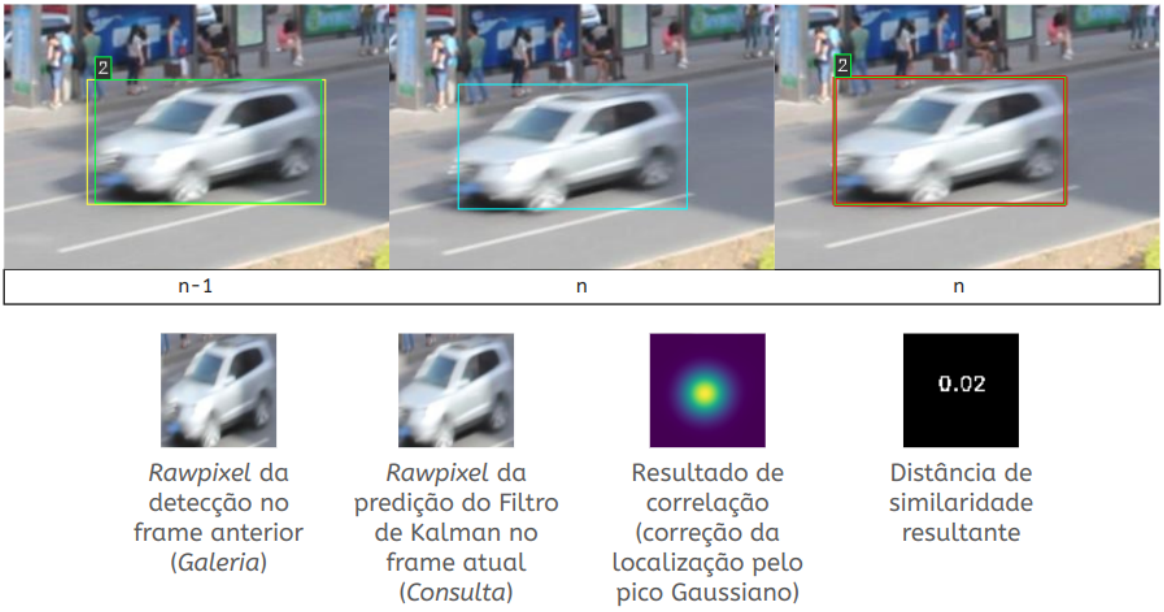
Figura 36 – Reidentificação com DCF no ByteTrack por fusão (*Fuse ReID*).

Fonte: Próprio autor.

casos. Porém, há uma exceção para reidentificação de curto alcance, no segundo estágio do ByteTrack, quando os objetos já rastreados não são associados a nenhuma detecção, possivelmente por confiança baixa ou oclusão do objeto, mas que pela predição do filtro de Kalman e reidentificação do DCF, como demonstrado na Figura 37, é possível realizar a manutenção de identidade e continuação do rastreio do objeto.

Dessa forma, integrando o módulo de reidentificação em estágios separados do ByteTrack, será possível avaliar o impacto que cada um tem no desempenho total. O resultado produzido será analisado na sequência durante os experimentos.

Figura 37 – Reidentificação com DCF no ByteTrack de curto alcance (*Short ReID*). Não há detecção no *frame* atual, portanto o rastreio seria perdido, o DCF mantém rastreando.



Fonte: Próprio autor.

4.4 Experimentos

Nesta seção, será apresentada a experimentação do rastreio de múltiplos objetos utilizando o módulo de reidentificação, baseado nos princípios mencionados anteriormente e nos modelos testados nos experimentos de reidentificação, realizados na Seção 3.3. Foram utilizados *datasets* compatíveis com esse tipo de tarefa dentro da visão computacional para que fosse possível analisar o desempenho por meio de métricas destinadas ao rastreio de objetos.

4.4.1 Recursos Utilizados

O algoritmo do ByteTrack roda um processamento mais leve e foi implementado apenas em CPU na sua versão original. Porém, a reidentificação feita pelo DeepSORT de Wojke, Bewley e Paulus (2017) possui implementação em GPU dedicada. Portanto, o mesmo *hardware* utilizado nos experimentos apresentados no Capítulo 3, descrito na Seção 3.3.1, será mantido, um *notebook* com CPU Intel Core i7-10750H e GPU NVIDIA RTX 2060.

Para fins de comparação, além do rastreamento de múltiplos objetos pelo ByteTrack com e sem reidentificação, com os modelos OSNet e DCF-ReID propostos, outros algoritmos de rastreio serão utilizados, como o clássico SORT, sem reidentificação, e também o DeepSORT, com reidentificação usando como modelo base o OSNet. Existem diversas

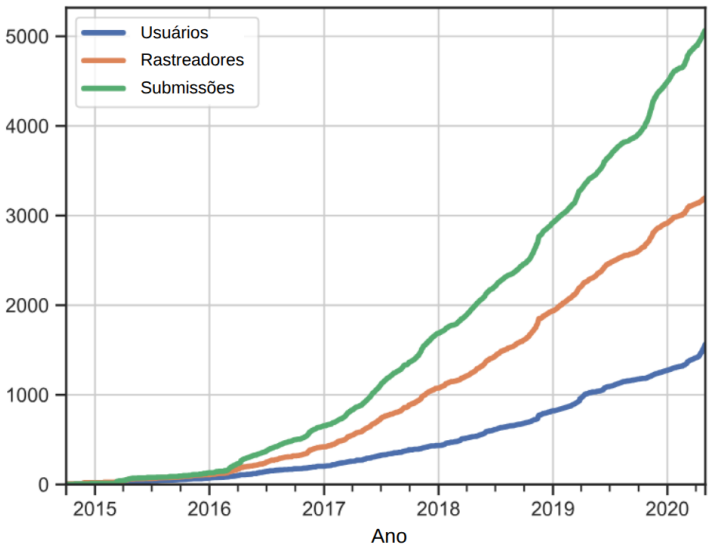
abordagens propostas para rastreadores de objetos ao longo dos anos, porém essas são referências comuns usadas para comparação.

Para compor o *benchmark* sobre rastreamento de múltiplos objetos, os *datasets* fornecidos pelo *MOT Challenge* são amplamente utilizados. Assim, nesse experimento, os *datasets* MOT17 e MOT20, com foco no rastreio de múltiplas pessoas em ambientes diversos, foram utilizados. Já para a avaliação de rastreamento com foco em veículos, foi empregado o *dataset* UA-DETRAC, fornecido pela Universidade Estadual de Nova York, em Albany, também presente no *NVIDIA AI City Challenge*.

4.4.2 Preparação dos Conjuntos de Dados

Desde seu lançamento em 2014, o *benchmark* para rastreamento de múltiplas pessoas em *single-camera*, conhecido como *MOT Challenge*, tem sido amplamente utilizado no meio acadêmico para submissão e avaliação de abordagens nesse campo da visão computacional, como pode ser visto no gráfico da Figura 38, tornando-se um padrão para avaliação de performance (DENDORFER et al., 2020a). Alguns dos principais *datasets* disponibilizados no *MOT Challenge* serão utilizados neste trabalho, sendo esses os conjuntos MOT17 e MOT20, visualizados na Figura 39, que fornecem cenas variadas para rastreamento de múltiplas pessoas.

Figura 38 – Estatísticas de usuário, rastreadores e trabalhos submetidos ao *MOT Challenge*.



Fonte: Imagem adaptada do trabalho sobre o framework *MOT Challenge* (DENDORFER et al., 2020a)

O MOT17 é composto por um misto de cenários, com visão próxima e distante das pessoas, assim como câmera fixa e em movimento, ideal para avaliar algoritmos de rastreio generalistas. O *dataset* é uma extensão do MOT16, porém com melhorias nas anotações,

como precisão das caixas delimitadoras e adição de pedestres não anotados anteriormente. Possui 14 sequências de diferentes cenários, divididas em metade para treinamento e metade para teste (DENDORFER et al., 2020a; MILAN et al., 2016).

O MOT20 possui uma complexidade maior por reunir uma densidade maior de pessoas no mesmo ambiente, podendo chegar até 246 pessoas em um único *frame*. Possui condições de iluminação baixa em alguns casos, com câmera fixa na cena e oscilações no movimento da mesma, assim como ambientes internos e externos. É composto nessa versão por 8 sequências de diferentes cenários divididos pela metade, destinados para treinamento e teste (DENDORFER et al., 2020a; DENDORFER et al., 2020b).

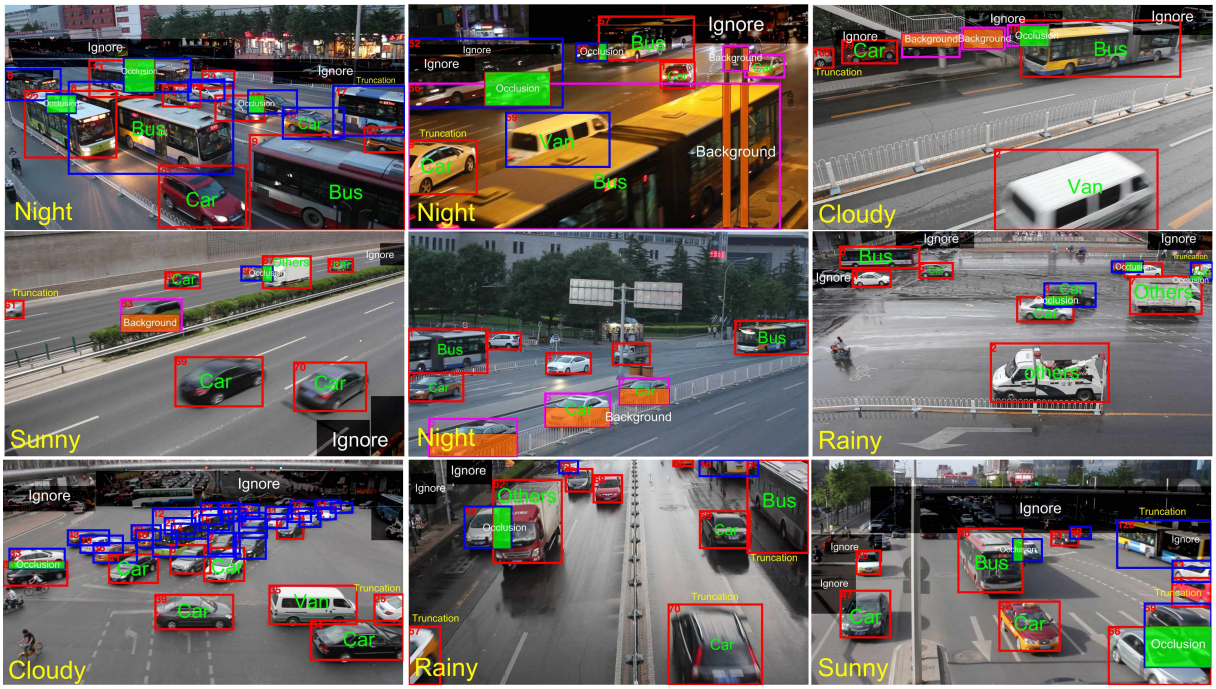
Figura 39 – *Datasets* MOT17 e MOT20.

Fonte: Próprio autor.

Já para avaliação de rastreamento múltiplo de veículos, o *dataset* UA-DETRAC, fornece cenários muito próximos dos encontrados em câmeras de monitoramento, como pode ser visualizado através da Figura 40. É composto por vídeos de 24 diferentes localidades, extraídos nas regiões de Beijing e Tianjin, na China. São 100 vídeos, com separação de 60 para treinamento e 40 para teste, que possuem resolução de 960x540 pixels, a 25 FPS, com variações de iluminação, ângulo e condições climáticas (WEN et al., 2020; NAPHADE et al., 2017).

Conhecendo-se o resultado positivo na reidentificação com DCF entre objetos na mesma câmera (*single camera*), cada fonte de vídeo será tratada individualmente, sem associação de múltiplas câmeras, assim como já seguem por padrão os *datasets* MOT17 e MOT20.

Figura 40 – *Dataset* UA-DETRAC.



Fonte: Wen et al. (2020).

4.4.3 Roteiro de Testes

O experimento para o desafio de rastreamento de múltiplos objetos foi conduzido com os rastreadores mencionados, utilizando o *framework* TrackEval, conforme recomendado no site do *MOT Challenge*. Foram empregados os conjuntos de treino dos *datasets* MOT17 e MOT20, voltados para o rastreamento de pessoas. Embora esses conjuntos de dados forneçam anotações públicas originalmente destinadas ao treinamento, elas foram utilizadas neste trabalho para teste, uma vez que os modelos de reidentificação de pessoas empregados já haviam sido pré-treinados em outros *datasets*. Juntamente, o *dataset* UA-DETRAC foi utilizado para rastreamento de veículos através do conjunto de teste propriamente destinado para realizar avaliações.

Para o MOT17 e MOT20 foram utilizadas como detecções de entrada para os algoritmos de rastreio, as detecções públicas já fornecidas pelos *datasets*, obtidas através das arquiteturas de redes neurais *Deformable Parts Model* (DPM), *Faster R-CNN* (FRCNN) e *Scale-Dependent Pooling* (SDP). Já para o conjunto UA-DETRAC foram utilizadas detecções privadas, geradas pelo próprio autor com a arquitetura para detecção de objetos YOLOv8, já que as detecções de base não estão disponíveis publicamente para este conjunto de dados.

Para a avaliação de desempenho individual de cada algoritmo, o *framework* utilizado disponibiliza diversas métricas de rastreamento. No entanto, apenas um subconjunto dessas

métricas foi adotado na apresentação dos resultados deste trabalho, sendo suficiente para fornecer uma visão representativa do comportamento e da eficácia de cada método avaliado. As métricas consideradas mais relevantes para a análise da precisão na identificação e localização dos objetos rastreados foram: HOTA, MOTA, IDF1, IDSW e Frag. Para a análise do desempenho computacional, utilizou-se a métrica de FPS (quadros por segundo), que indica a velocidade com que o algoritmo processa os frames de um vídeo, uma informação crucial em aplicações que exigem processamento em tempo real.

Foi avaliado o desempenho de rastreamento de múltiplos objetos com e sem reidentificação de objetos, usando o módulo de reidentificação proposto DCF-ReID e o modelo OSNet como base. O modelo OSNet treinado no *dataset* DukeMTMC-reID foi utilizado em ambos os testes, pois anteriormente no experimento de reidentificação, apresentado no Capítulo 3, demonstrou melhores resultados quando aplicado para reidentificação de veículos, se comparado ao modelo treinado no *dataset* VRIC, e para reidentificação de pessoas se manteve nos melhores resultados também.

O processamento foi feito para cada fonte de vídeo ou câmera presente nos *datasets* mencionados, ou seja, única câmera ou vídeo por vez. Os resultados serão exibidos nas tabelas das próximas subseções e correspondem às avaliações realizadas para cada rastreador, ordenados pelo melhor resultado da métrica HOTA.

Também foi realizada uma avaliação do desempenho da abordagem proposta em diferentes taxas de quadros por segundo (FPS) dos vídeos, com o objetivo de analisar seu comportamento em cenários de baixo FPS e em situações de perda de frames, comuns em sistemas de monitoramento por câmeras distribuídas, especialmente quando há instabilidades na rede de transmissão.

A seguir estão descritas as configurações definidas para os módulos de reidentificação integrados aos algoritmos de rastreamento de múltiplos objetos.

4.4.3.1 Configuração do DCF

- Arquitetura do modelo utilizado: Filtro de Correlação Discriminativo para Reidentificação, DCF-ReID.
- Metodologias para extração de características: *Rawpixel*.
- Tamanho do *patch* base em pixels: 36x68.
- Métrica de similaridade: *Peak-to-Sidelobe Ratio* (PSR).
- Máscara para cálculo do PSR: 1.0x0.2 (100% e 20%) da resolução do mapa de correlação.
- Integração: ByteTrack (*Fuse1*, *Fuse2*, *Fuse3* e *Short*).

4.4.3.2 Configuração da rede OSNet

- Arquitetura do modelo utilizado: *Omni-Scale Network*, OSNet-x0_25.
- Modelo pré-treinado em *datasets* para reidentificação: DukeMTMC-reID.
- Tamanho do *patch* base em pixels: 128x256.
- Métrica de similaridade: Distância Euclidiana (*Euclidean*).
- Integração: DeepSORT (modelo base), ByteTrack (*Fuse1*, *Fuse2* e *Fuse3*).

4.5 Resultados

4.5.1 Resultados para Rastreamento de Pessoas

Na Tabela 8 são apresentados os primeiros resultados extraídos através do experimento para rastreamento de múltiplas pessoas realizado no *dataset* MOT17, onde são comparados os rastreadores SORT, DeepSORT e ByteTrack, sendo o ByteTrack com versões diferentes conforme as integrações para reforço por reidentificação, no caso, com os módulos OSNet e DCF-ReID.

Tabela 8 – Resultados dos rastreadores no *dataset* MOT17. As detecções públicas utilizadas foram extraídas com SDP (*Scale-Dependent Pooling*).

Abordagem modelo	**Descrição**	**HOTA↑**	**MOTA↑**	**IDF1↑**	**IDSW↓**	**Frag↓**	**FPS↑**
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse1+Short	**56,00**	**67,00**	**66,32**	**492**	**946**	22
ByteTrack + DCF-ReID	Rawpixel, 36x68, Short	55,22	66,69	65,38	575	1.012	200
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse1	54,98	64,45	66,10	551	1.950	23
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse1	54,57	64,36	65,17	624	2.002	57
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse3	54,55	64,41	65,14	615	1.997	276
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse2	54,55	64,34	65,14	630	2.019	267
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse2	54,51	64,34	65,06	637	2.007	59
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse3	54,51	64,34	65,06	637	2.007	59
ByteTrack	-	54,50	64,33	65,05	637	2.006	408
DeepSORT + OSNet-x0_25	DukeMTMC-reID, 128x256	51,16	63,85	59,39	952	2.240	20
SORT	-	49,34	60,93	55,77	746	1.159	**607**

No *dataset* MOT17, o módulo de reidentificação próprio DCF-ReID, assim como o módulo com OSNet, agregam uma pequena melhoria na acurácia de rastreio e identificação, porém adicionam maior atraso no processamento, principalmente com OSNet. O módulo DCF-ReID nos estágios *Short, Fuse2 e Fuse3* não tem tanto impacto no processamento, pois são acionados com menor frequência que o estágio *Fuse1*, sendo esse, praticamente acionado a cada iteração do ByteTrack.

Já na Tabela 9 estão os resultados referentes ao experimento com o conjunto MOT20, que possui maior densidade de pessoas no mesmo espaço e condições desafiadoras de iluminação, portanto, um processamento mais pesado e complexo.

Tabela 9 – Resultados dos rastreadores no *dataset* MOT20. As detecções públicas utilizadas foram extraídas com FRCNN (*Faster R-CNN*).

Abordagem modelo	**Descrição**	**HOTA↑**	**MOTA↑**	**IDF1↑**	**IDSW↓**	**Frag↓**	**FPS↑**
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse1+Short	**41,41**	**59,58**	**49,20**	**5.944**	**15.904**	1
ByteTrack + DCF-ReID	Rawpixel, 36x68, Short	40,28	59,10	47,75	6.583	16.728	19
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse1	38,49	50,76	46,44	6.193	41.683	1
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse1	37,73	50,69	45,33	6.978	41.801	17
ByteTrack	-	37,66	50,69	45,17	7.007	41.832	35
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse2	37,66	50,69	45,17	7.007	41.832	18
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse2	37,66	50,69	45,17	7.007	41.832	30
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse3	37,66	50,69	45,17	7.007	41.832	18
ByteTrack + DCF-ReID	Rawpixel, 36x68, Fuse3	37,66	50,70	45,16	6.998	41.827	29
DeepSORT + OSNet-x0_25	DukeMTMC-reID, 128x256	31,38	50,23	35,75	11.617	41.569	5
SORT	-	28,38	44,49	30,38	14.531	22.718	**140**

Como o *dataset* MOT20 possui uma densidade maior de objetos sendo rastreados simultaneamente, a velocidade de processamento cai bastante em todos os rastreadores, mas novamente o ByteTrack com o módulo de reidentificação DCF-ReID fica na melhor colocação em praticamente todas as métricas, sendo essas: HOTA, MOTA, IDF1, IDSW e Frag.

4.5.2 Resultados para Rastreamento de Veículos

O resultado referente ao *dataset* UA-DETRAC, para análise do rastreio de múltiplos veículos, está presente na Tabela 10, contendo nesse conjunto de teste condições muito próximas das encontradas em câmeras de monitoramento de tráfego.

Tabela 10 – Resultados dos rastreadores no *dataset* UA-DETRAC. As detecções privadas utilizadas foram extraídas com YOLOv8 pelo próprio autor.

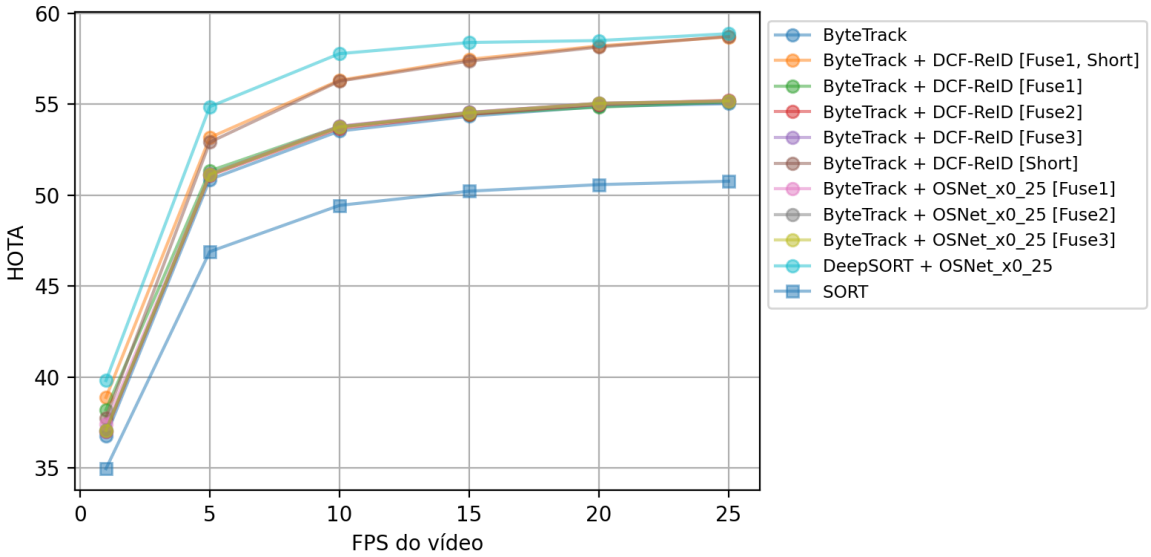
Abordagem modelo	**Descrição**	**HOTA↑**	**MOTA↑**	**IDF1↑**	**IDSW↓**	**Frag↓**	**FPS↑**
DeepSORT + OSNet-x0_25	DukeMTMC-reID, 128x256	**58,89**	**59,11**	69,80	2.294	3.345	21
ByteTrack + DCF-ReID	Rawpixel, 68x68, Short	58,74	57,29	70,52	**233**	871	470
ByteTrack + DCF-ReID	Rawpixel, 68x68, Fuse1+Short	58,72	57,41	**70,60**	235	**847**	79
ByteTrack + DCF-ReID	Rawpixel, 68x68, Fuse2	55,22	52,91	66,52	256	2.966	408
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse1	55,20	52,93	66,49	254	2.901	74
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse3	55,19	52,93	66,45	258	2.906	76
ByteTrack + DCF-ReID	Rawpixel, 68x68, Fuse3	55,19	52,94	66,45	258	2.907	578
ByteTrack + OSNet-x0_25	DukeMTMC-reID, 128x256, Fuse2	55,17	52,94	66,42	260	2.912	75
ByteTrack + DCF-ReID	Rawpixel, 68x68, Fuse1	55,14	52,92	66,44	256	2.889	87
ByteTrack	-	55,04	52,56	66,31	247	2.771	1.059
SORT	-	50,77	49,40	57,18	2.100	3.579	**1.194**

Através do conjunto UA-DETRAC é possível verificar uma diferença maior nos resultados encontrados, onde o DeepSORT, bem como o ByteTrack com módulo DCF-ReID, apresentam superioridade em relação ao ByteTrack original e demais versões, quando comparadas às métricas de acurácia de rastreio e de identificação. Entretanto, nesse caso, o DeepSORT é mais lento. Assim como no SORT, o DeepSORT apresenta uma troca alta de IDs dos objetos durante o processo, visto através da métrica IDSW (*ID Switch*).

Sabendo que em condições reais, com monitoramento através de câmeras distribuídas pela cidade, instabilidades na transmissão de vídeo podem ocorrer, um teste forçado com o *dataset* UA-DETRAC foi realizado. O FPS original dos vídeos foi reduzido para simular possíveis falhas no fornecimento dos frames, o que pode gerar situações que exijam mais do suporte por reidentificação no rastreamento, pois os objetos rastreados podem ter uma mudança visual e de localização maior entre frames. Os resultados obtidos podem ser

observados na Figura 41.

Figura 41 – Desempenho HOTA em função do FPS do vídeo de entrada para o *dataset* UA-DETRAC.



Fonte: Próprio autor.

Para alguns dos rastreadores listados para o UA-DETRAC em relação ao FPS do vídeo amostrado, pode-se perceber a diferença de respostas para a métrica de avaliação HOTA. Como mencionado, o DeepSORT apresenta melhor desempenho geral, mas quando a falha entre frames começa a aumentar, ou seja, o FPS do vídeo fica menor, como um *frame* a cada um segundo, o resultado se aproxima do mesmo apresentado pelo ByteTrack.

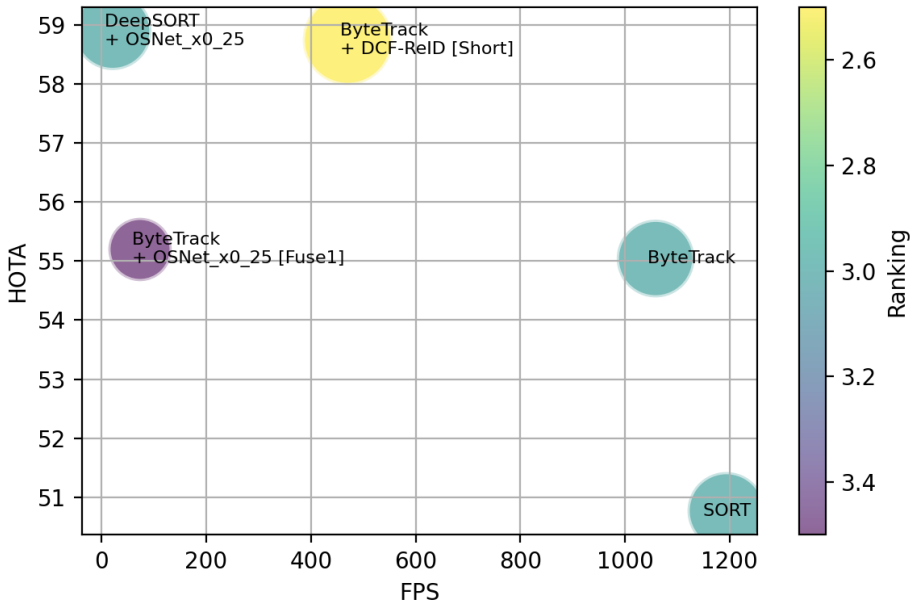
Também é possível notar que há uma variação mínima, tanto para melhor quanto para pior, quando o módulo de reidentificação DCF está inserido no ByteTrack nos estágios *Fuse1, Fuse2 e Fuse3*. Nesses casos, o benefício em relação ao ByteTrack original parece ser muito pequeno ou nulo, quando combinado com IoU.

Porém, a integração do DCF-ReID ao ByteTrack para reidentificação de curto alcance (*Short ReID*) fornece um suporte de rastreio contínuo que eleva os resultados gerados pelo ByteTrack original, produzindo a acurácia equiparada ao DeepSORT, porém a um custo computacional muito menor de processamento, analisando-se pelo FPS mais alto em que o algoritmo é capaz de rodar.

Isso pode ser evidenciado através da Figura 42, que resume bem o comportamento dos rastreadores apresentados, relacionando a acurácia de rastreio de múltiplos objetos, através da métrica HOTA, com o FPS de processamento, ou seja, o balanço ideal entre acurácia de rastreio e velocidade de processamento. O *ranking* dos rastreadores apresentados foi calculado como sendo a média das colocações individuais obtidas para as duas métricas

comparadas, considerando que valores mais próximos de 1, ou seja, a primeira colocação, representam melhor desempenho na comparação entre HOTA e FPS, portanto, um melhor equilíbrio entre precisão de rastreio e velocidade de processamento.

Figura 42 – Desempenho dos rastreadores para a métrica HOTA em função do FPS de processamento para o *dataset* UA-DETRAC. Quanto maior a métrica HOTA e o FPS melhor.



Fonte: Próprio autor.

4.5.3 Análise e Considerações Gerais sobre os Resultados

O trabalho aqui desenvolvido, baseado no filtro de correlação discriminativo, especificamente citando o trabalho do KCF, teve como foco o estudo desse tipo de abordagem já conhecida no campo da visão computacional, porém não vista estritamente como solução para reidentificação de objetos e, geralmente, vista como metodologia de rastreamento visual.

Em relação aos testes feitos nos três *datasets* mencionados, o ByteTrack se sobressai no MOT17 e MOT20, porém no UA-DETRAC o DeepSORT demonstra maior acurácia. Já o SORT fica para trás em basicamente todas as métricas, exceto para o FPS, justamente por ser uma abordagem simples e extremamente leve.

Os resultados e análise desenvolvidos demonstram que o módulo de reidentificação DCF-ReID pode melhorar o rastreio equilibrado que o ByteTrack já oferece. O algoritmo base desenvolvido para correlação é simples e pode ser melhorado, a partir de estudos mais avançados que sigam essa estratégia, para possivelmente aumentar os níveis de assertividade a um custo computacional ainda menor.

5 CONCLUSÃO

Após as análises feitas nos casos de reidentificação e rastreamento de objetos, foi possível observar vários aspectos das abordagens clássicas e modernas, envolvendo algoritmos que trabalham no domínio do espaço, da frequência e com redes neurais, onde algumas decisões de projeto variam conforme a necessidade da aplicação. No caso deste trabalho, o direcionamento principal é para o processamento de vídeo fornecido por câmeras de monitoramento em cidades inteligentes.

Com isso, pode-se mencionar que há uma necessidade no desenvolvimento de soluções que ofereçam um equilíbrio entre acurácia e velocidade de processamento em tempo real. Tanto a reidentificação quanto o rastreamento de objetos, baseados em redes neurais, resultam em uma melhor acurácia de um modo geral, pois os modelos conseguem aprender características e padrões mais profundos e complexos, mas com um custo maior como contrapartida. A carga computacional necessária prejudica o processamento em tempo real na maioria dos casos, o que normalmente inviabiliza o seu uso em câmeras de monitoramento.

Nesse contexto, considerando o estudo realizado sobre o filtro de correlação, aplicado apenas para reidentificação de objetos, sem envolver rastreamento, nota-se que tal filtro pode oferecer resultados satisfatórios, comprovados pelos testes feitos nos *datasets* de reidentificação. A abordagem proposta pode ser adequada para reidentificação em única câmera, com uma galeria formada por quantidades de imagens pequenas. O principal destaque fica pela flexibilidade de modificar o algoritmo para obter melhores respostas e também por não ser necessário realizar treinamento supervisionado *offline*. Uma vez definida a extração de características, não é necessário dedicar tempo rotulando ou treinando o modelo para a tarefa desejada.

Considerando-se agora a adequação feita no filtro de correlação para criar um módulo de reidentificação integrado ao algoritmo de rastreio do ByteTrack, foi possível concluir que melhorias são proporcionadas por esse módulo, diferente das integrações com redes neurais que tornam inviável o processamento em tempo real. O DCF-ReID aprimora o ByteTrack a um custo computacional menor, tornando-o assim uma alternativa mais eficiente se comparada ao DeepSORT.

A reidentificação de curto alcance do DCF pode trabalhar bem como suporte de reidentificação, combinada a algoritmos clássicos e leves, sendo ideal para processamento de vídeo em tempo real em única câmera. Além disso, é uma arquitetura flexível, possibilitando melhorias e customizações em diferentes partes do processo, como na extração

de características, na implementação do filtro de correlação em si, cálculo de métricas e processamento em lotes na GPU.

5.1 Contribuições da Dissertação

A dificuldade em encontrar referencial teórico relativo ao uso de filtro de correlação para a tarefa de reidentificação em visão computacional foi um dos principais motivadores deste trabalho. O estudo realizado, visando essa técnica como recurso principal tanto para reidentificação quanto à integração no rastreio de múltiplos objetos, mostrou a viabilidade da abordagem para processamento de vídeo através de câmeras de monitoramento, ressaltando os pontos positivos e negativos encontrados.

Foi possível demonstrar que metodologias clássicas ainda podem ser utilizadas para atingir resultados satisfatórios, em alguns casos se aproximar ou até superar resultados obtidos com redes neurais, a um custo operacional e computacional menor.

Com as análises feitas, o uso de modelos treinados em redes neurais pode se encaixar nos problemas de reidentificação com grandes bancos de imagens e também, não testado nesse trabalho mas sugerido pelo indicativo das métricas obtidas, para reidentificação em múltiplas câmeras. Já a utilização de filtro de correlação é mais eficiente na reidentificação dentro do rastreamento de objetos localmente, em uma única câmera.

5.2 Trabalhos Futuros

Como mencionado durante o trabalho desenvolvido, o foco se concentrou na comprovação do uso de filtro de correlação discriminativo como um método para reidentificação e rastreio de múltiplos objetos.

Assim, apenas um algoritmo baseado no KCF, que oferece bom desempenho geral e simplicidade, foi utilizado para fins de teste. O KCF é uma metodologia que trouxe ótimos resultados quando proposta, porém muitos trabalhos foram elaborados posteriormente melhorando seu desempenho e utilização. Da mesma forma, a arquitetura do DCF-ReID aqui desenvolvida também pode ser customizada e aprimorada.

Portanto, para superar os resultados obtidos, estudos mais aprofundados podem ser realizados, tratando isoladamente cada parte que constitui a solução proposta. Por exemplo, melhorias ou otimizações podem ser desenvolvidas para os *kernels* de processamento no domínio da frequência, para os cálculos de correlação entre as imagens, assim como a utilização de redes neurais combinadas com o filtro de correlação, para realizar a extração de características mais complexas e de forma mais eficiente.

REFERÊNCIAS

ADEBAYO, A.-A. Mace correlation filter algorithm for face verification in surveillance scenario. *JOURNAL OF COMPUTER SCIENCE AND ENGINEERING (JCSE)*, Volume 18, p. Page 5–18, 04 2013. Citado 2 vezes nas páginas 52 e 68.

ALMASAWA, M. O.; ELREFAEI, L. A.; MORIA, K. M. A survey on deep learning-based person re-identification systems. *IEEE Access*, v. 7, p. 175228–175247, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:209336262>. Citado 2 vezes nas páginas 37 e 39.

AMIRI, A.; KAYA, A.; KECELI, A. S. *A Comprehensive Survey on Deep-Learning-based Vehicle Re-Identification: Models, Data Sets and Challenges*. 2024. Disponível em: <https://arxiv.org/abs/2401.10643>. Citado na página 58.

BABU, E. K. et al. Facial feature extraction using a symmetric inline matrix-lbp variant for emotion recognition. *Sensors*, v. 22, n. 22, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/22/8635>. Citado 2 vezes nas páginas 54 e 55.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, v. 2008, 01 2008. Citado na página 32.

BEWLEY, A. et al. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. Disponível em: <http://dx.doi.org/10.1109/ICIP.2016.7533003>. Citado 3 vezes nas páginas 30, 31 e 81.

BOLLIER, D. *How Smart Growth Can Stop Sprawl: A Fledgling Citizen Movement Expands*. Washington, DC: Essential Books, 1998. Citado na página 18.

BOLME, D. S. et al. Visual object tracking using adaptive correlation filters. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2010. p. 2544–2550. Citado 5 vezes nas páginas 19, 47, 48, 49 e 65.

BORJI, A. et al. Salient object detection: A survey. *Computational Visual Media*, Springer Science and Business Media LLC, v. 5, n. 2, p. 117–150, jun. 2019. ISSN 2096-0662. Disponível em: <http://dx.doi.org/10.1007/s41095-019-0149-9>. Citado na página 55.

BRADSKI, G. Real time face and object tracking as a component of a perceptual user interface. In: *Proceedings Fourth IEEE Workshop on Applications of Computer Vision. WACV'98 (Cat. No.98EX201)*. [S.l.: s.n.], 1998. p. 214–219. Citado na página 29.

BROMLEY, J. et al. Signature verification using a "siamese"time delay neural network. In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. (NIPS'93), p. 737–744. Citado na página 37.

CHEN, N.; CHEN, Y. Smart city surveillance at the network edge in the era of iot: Opportunities and challenges. In: _____. [S.l.: s.n.], 2018. p. 153–176. ISBN 978-3-319-76668-3. Citado na página 19.

COMANICIU, D.; MEER, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 5, p. 603–619, 2002. Citado na página 29.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, set. 1995. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00994018>. Citado na página 26.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.: s.n.], 2005. v. 1, p. 886–893 vol. 1. Citado 2 vezes nas páginas 26 e 54.

DECANN, B.; ROSS, A. Relating roc and cmc curves via the biometric menagerie. *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, p. 1–8, 2013. Disponível em: <https://api.semanticscholar.org/CorpusID:7489867>. Citado na página 39.

DENDORFER, P. et al. *MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking*. 2020. Citado 2 vezes nas páginas 87 e 88.

DENDORFER, P. et al. *MOT20: A benchmark for multi object tracking in crowded scenes*. 2020. Citado 3 vezes nas páginas 33, 34 e 88.

DING, Y.; DUAN, Z.; LI, S. Source-free unsupervised multi-source domain adaptation via proxy task for person re-identification. *The Visual Computer*, v. 38, 06 2022. Citado na página 71.

FARENZENA, M. et al. Person re-identification by symmetry-driven accumulation of local features. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2010. p. 2360–2367. Citado na página 57.

FERNANDEZ-SANJURJO, M.; MUCIENTES, M.; BREA, V. M. Real-time multiple object visual tracking for embedded gpu systems. *IEEE Internet of Things Journal*, v. 8, n. 11, p. 9177–9188, 2021. Citado 2 vezes nas páginas 60 e 61.

FOURIER, J.-B. J. *Théorie Analytique de la Chaleur*. Paris: Chez Firmin Didot, père et fils, 1822. Citado na página 41.

GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo, Brasil: Pearson Prentice Hall, 2010. ISBN 978-85-7605-401-6. Citado 4 vezes nas páginas 41, 42, 44 e 45.

GORDON, N.; SALMOND, D.; SMITH, A. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, v. 140, p. 107–113, 1993. Disponível em: <https://digital-library.theiet.org/doi/abs/10.1049/ip-f-2.1993.0015>. Citado na página 29.

GRAY, R. M. *Toeplitz and Circulant Matrices: A Review*. [S.l.: s.n.], 2006. Citado na página 50.

HENRIQUES, J. F. et al. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), v. 37, n. 3, p. 583–596, mar. 2015. ISSN 2160-9292.

Disponível em: <http://dx.doi.org/10.1109/TPAMI.2014.2345390>. Citado 8 vezes nas páginas 20, 47, 49, 50, 51, 53, 65 e 66.

HESTER, C. F.; CASASENT, D. Multivariant technique for multiclass pattern recognition. *Applied Optics*, United States: National Library of Medicine (NLM), v. 19, n. 11, p. 1758–1761, June 1 1980. ISSN 1559-128X. A technique for multiclass optical pattern recognition of different perspective views of an object is described. Each multiclass representation of an object is described as an orthonormal basis function expansion, and a single averaged matched spatial filter is then produced from a weighted linear combination of these functions. The technique is demonstrated for a terminal missile guidance application using IR tank imagery. Citado na página 47.

HORN, B. K.; SCHUNCK, B. G. Determining optical flow. *Artificial Intelligence*, v. 17, n. 1, p. 185–203, 1981. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/0004370281900242>. Citado na página 29.

HYONA, J.; LI, J.; OKSAMA, L. Eye behavior during multiple object tracking and multiple identity tracking. *Vision*, v. 3, n. 3, 2019. ISSN 2411-5150. Disponível em: <https://www.mdpi.com/2411-5150/3/3/37>. Citado na página 28.

JAVED, S. et al. *Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook*. 2021. Disponível em: <https://arxiv.org/abs/2112.02838>. Citado 3 vezes nas páginas 47, 48 e 51.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, v. 82, n. 1, p. 35–45, 03 1960. ISSN 0021-9223. Disponível em: <https://doi.org/10.1115/1.3662552>. Citado na página 29.

KANACI, A.; ZHU, X.; GONG, S. Vehicle re-identification in context. In: *Pattern Recognition - 40th German Conference, GCPR 2018, Stuttgart, Germany, September 10-12, 2018, Proceedings*. [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 70 e 71.

KARANAM, S. et al. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 41, n. 3, p. 523–536, 2019. Citado na página 39.

KOSTIC, Z. et al. *Smart City Intersections: Intelligence Nodes for Future Metropolises*. 2022. Disponível em: <https://arxiv.org/abs/2205.01686>. Citado na página 18.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>. Citado na página 26.

KUHN, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, v. 2, n. 1-2, p. 83–97, 1955. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109>. Citado na página 31.

LI, W. et al. Deepreid: Deep filter pairing neural network for person re-identification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 152–159. Citado na página 58.

LI, W.-H. et al. Correlation based identity filter: An efficient framework for person search. In: ZHAO, Y.; KONG, X.; TAUBMAN, D. (Ed.). *Image and Graphics*. Cham: Springer International Publishing, 2017. p. 250–261. ISBN 978-3-319-71607-7. Citado 2 vezes nas páginas 57 e 61.

LI, Y.; CHEN, C. L. P.; ZHANG, T. A survey on siamese network: Methodologies, applications, and opportunities. *IEEE Transactions on Artificial Intelligence*, v. 3, n. 6, p. 994–1014, 2022. Citado na página 37.

LIAO, R.; CHEN, L. *An Evolutionary Note on Smart City Development in China*. 2022. Disponível em: <https://arxiv.org/abs/2203.13169>. Citado na página 18.

LIAO, S. et al. Person re-identification by local maximal occurrence representation and metric learning. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 2197–2206. Citado na página 57.

LIU, W. et al. Ssd: Single shot multibox detector. In: _____. *Lecture Notes in Computer Science*. Springer International Publishing, 2016. p. 21–37. ISBN 9783319464480. Disponível em: <http://dx.doi.org/10.1007/978-3-319-46448-0_2>. Citado 2 vezes nas páginas 26 e 27.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981. (IJCAI'81), p. 674–679. Citado na página 29.

LUITEN, J. et al. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, Springer Science and Business Media LLC, v. 129, n. 2, p. 548–578, out. 2020. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1007/s11263-020-01375-2>. Citado 2 vezes nas páginas 35 e 36.

LUKEZIC, A. et al. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, Springer Science and Business Media LLC, v. 126, n. 7, p. 671–688, jan. 2018. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1007/s11263-017-1061-3>. Citado 2 vezes nas páginas 47 e 59.

LUO, H. et al. *An Empirical Study of Vehicle Re-Identification on the AI City Challenge*. 2021. Citado 2 vezes nas páginas 70 e 72.

LUO, W. et al. Multiple object tracking: A literature review. *Artificial Intelligence*, v. 293, p. 103448, 2021. ISSN 0004-3702. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0004370220301958>. Citado 3 vezes nas páginas 19, 28 e 30.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008. Citado na página 40.

MARRON, M. et al. Comparing a kalman filter and a particle filter in a multiple objects tracking application. In: *2007 IEEE International Symposium on Intelligent Signal Processing*. [S.l.: s.n.], 2007. p. 1–6. Citado na página 29.

MILAN, A. et al. *MOT16: A Benchmark for Multi-Object Tracking*. 2016. Citado 3 vezes nas páginas 33, 34 e 88.

MONTABONE, S.; SOTO, A. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing*, v. 28, n. 3, p. 391–402, 2010. ISSN 0262-8856. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0262885609001371>. Citado na página 66.

MONTOYA-TORRES, J. R. et al. Big data analytics and intelligent transportation systems. *IFAC-PapersOnLine*, v. 54, n. 2, p. 216–220, 2021. ISSN 2405-8963. 16th IFAC Symposium on Control in Transportation Systems CTS 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896321004614>. Citado na página 18.

NAPHADE, M. et al. The nvidia ai city challenge. In: . [S.l.: s.n.], 2017. p. 1–6. Citado 2 vezes nas páginas 72 e 88.

OJALA, T.; PIETIKäINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, v. 29, n. 1, p. 51–59, 1996. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/0031320395000674>. Citado na página 54.

OLADIMEJI, D. et al. Smart transportation: An overview of technologies and applications. *Sensors*, v. 23, n. 8, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/8/3880>. Citado na página 19.

OLIVEIRA, H. S.; MACHADO, J. J. M.; TAVARES, J. M. R. S. Re-identification in urban scenarios: A review of tools and methods. *Applied Sciences*, v. 11, n. 22, 2021. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/11/22/10809>. Citado 3 vezes nas páginas 37, 39 e 40.

PARK, Y. et al. Multiple object tracking in deep learning approaches: A survey. *Electronics*, v. 10, n. 19, 2021. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/10/19/2406>. Citado 2 vezes nas páginas 29 e 30.

Pooja G. et al. Recent trends and challenges in smart cities. *EAI Endorsed Transactions on Smart Cities*, v. 6, n. 3, p. e4, set. 2022. Citado na página 18.

PYLYSHYN, Z. W.; STORM, R. W. Tracking multiple independent targets: Evidence for a parallel tracking mechanism*. *Spatial Vision*, Brill, Leiden, The Netherlands, v. 3, n. 3, p. 179 – 197, 1988. Disponível em: <https://brill.com/view/journals/sv/3/3/article-p179_3.xml>. Citado 2 vezes nas páginas 8 e 28.

REDMON, J. et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. Citado 2 vezes nas páginas 26 e 27.

REHMAN, A.; HUSSEIN, S.; SULTANI, W. *Joint Stream: Malignant Region Learning for Breast Cancer Diagnosis*. 2024. Disponível em: <https://arxiv.org/abs/2406.18212>. Citado na página 46.

REN, S. et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. Citado 2 vezes nas páginas 26 e 27.

RISTANI, E. et al. *Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking*. 2016. Disponível em: <https://arxiv.org/abs/1609.01775>. Citado 2 vezes nas páginas 34 e 35.

RISTANI, E. et al. *Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking*. 2016. Citado na página 70.

SAWIDES, M.; KUMAR, B.; KHOSLA, P. Corefaces - robust shift invariant pca based correlation filter for illumination tolerant face recognition. In: . [S.l.: s.n.], 2004. v. 2, p. II–834. ISBN 0-7695-2158-4. Citado na página 52.

SOLOMON, C.; BRECKON, T. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in MATLAB*. 1. ed. Hoboken, NJ, USA: Wiley-Blackwell, 2011. ISBN 978-0-470-84472-4. Citado 6 vezes nas páginas 22, 41, 42, 43, 44 e 45.

STUCHI, J. et al. A frequency-domain approach with learnable filters for image classification. *SSRN Electronic Journal*, 01 2023. Citado na página 46.

SUN, Z. et al. A comprehensive review of pedestrian re-identification based on deep learning. *Complex & Intelligent Systems*, v. 10, n. 2, p. 1733–1768, April 2024. ISSN 2198-6053. Disponível em: <https://doi.org/10.1007/s40747-023-01229-7>. Citado 3 vezes nas páginas 37, 38 e 40.

SZELISKI, R. *Computer Vision: Algorithms and Applications*. 2nd. ed. Cham, Switzerland: Springer, 2022. ISBN 978-3-030-34371-2. Citado 4 vezes nas páginas 40, 41, 43 e 45.

TANG, Z. et al. *CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification*. 2019. Disponível em: <https://arxiv.org/abs/1903.09254>. Citado na página 72.

TERVEN, J.; CORDOVA-ESPARZA, D.-M.; ROMERO-GONZALEZ, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, MDPI AG, v. 5, n. 4, p. 1680–1716, nov. 2023. ISSN 2504-4990. Disponível em: <http://dx.doi.org/10.3390/make5040083>. Citado na página 27.

THOMPSON, N. C. et al. *The Computational Limits of Deep Learning*. 2022. Disponível em: <https://arxiv.org/abs/2007.05558>. Citado na página 19.

TIAN, Y.; YE, Q.; DOERMANN, D. *YOLOv12: Attention-Centric Real-Time Object Detectors*. 2025. Disponível em: <https://arxiv.org/abs/2502.12524>. Citado na página 27.

ULLAH, I. et al. A brief survey of visual saliency detection. *Multimedia Tools and Applications*, v. 79, 12 2020. Citado 2 vezes nas páginas 55 e 56.

VALMADRE, J. et al. *End-to-end representation learning for Correlation Filter based tracking*. 2017. Disponível em: <https://arxiv.org/abs/1704.06036>. Citado na página 60.

VIOLA, P.; JONES, M. Robust real-time face detection. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001.* [S.l.: s.n.], 2001. v. 2, p. 747–747. Citado na página 26.

WANG, G.; SONG, M.; HWANG, J.-N. *Recent Advances in Embedding Methods for Multi-Object Tracking: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2205.10766>. Citado na página 30.

WEI, L. et al. *Person Transfer GAN to Bridge Domain Gap for Person Re-Identification*. 2018. Citado na página 70.

WEIJER, J. van de et al. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, v. 18, n. 7, p. 1512–1523, 2009. Citado 2 vezes nas páginas 53 e 54.

WEN, L. et al. *UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking*. 2020. Citado 3 vezes nas páginas 71, 88 e 89.

WOJKE, N.; BEWLEY, A.; PAULUS, D. *Simple Online and Realtime Tracking with a Deep Association Metric*. 2017. Disponível em: <https://arxiv.org/abs/1703.07402>. Citado 3 vezes nas páginas 30, 31 e 86.

YADAV, S.; PAYANDEH, S. Datar: Depth augmented target redetection using kernelized correlation filter. *Multimedia Systems*, v. 29, 10 2022. Citado na página 59.

YE, M. et al. *Transformer for Object Re-Identification: A Survey*. 2024. Disponível em: <https://arxiv.org/abs/2401.06960>. Citado na página 38.

ZAIDI, S. S. A. et al. *A Survey of Modern Deep Learning based Object Detection Models*. 2021. Citado na página 26.

ZAKRIA et al. *Trends in Vehicle Re-identification Past, Present, and Future: A Comprehensive Review*. 2021. Disponível em: <https://arxiv.org/abs/2102.09744>. Citado na página 19.

ZHANG, Y. et al. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. 2022. Disponível em: <https://arxiv.org/abs/2110.06864>. Citado 4 vezes nas páginas 30, 31, 81 e 82.

ZHAO, D. et al. Multi-object tracking with correlation filter for autonomous vehicle. *Sensors*, v. 18, n. 7, 2018. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/18/7/2004>. Citado na página 60.

ZHAO, Q.; KOCH, C. Learning visual saliency. In: *2011 45th Annual Conference on Information Sciences and Systems*. [S.l.: s.n.], 2011. p. 1–6. Citado na página 55.

ZHAO, R.; OUYANG, W.; WANG, X. Unsupervised salience learning for person re-identification. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2013. p. 3586–3593. Citado na página 57.

ZHENG, L. et al. Scalable person re-identification: A benchmark. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1116–1124. Citado na página 70.

ZHENG, L.; YANG, Y.; HAUPTMANN, A. G. *Person Re-identification: Past, Present and Future*. 2016. Disponível em: <https://arxiv.org/abs/1610.02984>. Citado na página 58.

ZHENG, M. et al. *Re-Identification with Consistent Attentive Siamese Networks*. 2019. Disponível em: <https://arxiv.org/abs/1811.07487>. Citado na página 37.

ZHOU, K.; XIANG, T. *Torchreid: A Library for Deep Learning Person Re-Identification in Pytorch*. 2019. Citado na página 70.

ZHOU, K. et al. *Omni-Scale Feature Learning for Person Re-Identification*. 2019. Citado 3 vezes nas páginas 21, 38 e 72.

ZHU, C. et al. Efficient and practical correlation filter tracking. *Sensors*, v. 21, n. 3, 2021. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/21/3/790>. Citado na página 59.

ZOU, Z. et al. *Object Detection in 20 Years: A Survey*. 2023. Citado na página 26.

Apêndices

APÊNDICE A – INTEGRAÇÃO ENTRE BYTETRACK E MÓDULO DE REIDENTIFICAÇÃO DCF