



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
PIAUÍ CAMPUS CORRENTE
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

LEANDRO NATÁLIO FERREIRA BATISTA

**APLICAÇÃO E ANÁLISE DE ALGORITMOS DE RASTREAMENTO VISUAL
PARA DETECÇÃO DE BOLA EM PARTIDAS DE VOLEIBOL**

CORRENTE, PIAUÍ

2025

LEANDRO NATÁLIO FERREIRA BATISTA

APLICAÇÃO E ANÁLISE DE ALGORITMOS DE RASTREAMENTO VISUAL
PARA DETECÇÃO DE BOLA EM PARTIDAS DE VOLEIBOL

Trabalho de Conclusão de Curso (monografia)
apresentado como exigência parcial para obten-
ção do diploma do Curso de Análise e Desen-
volvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia do Piauí, Cam-
pus Corrente.

Orientador: Prof. Me. Carlos Estevão Bastos
Sousa

CORRENTE, PIAUÍ

2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Batista, Leandro Natálio Ferreira Batista

B333a Aplicação e análise de algoritmos de rastreamento visual para detecção de bola em partidas de voleibol / Leandro Natálio Ferreira Batista Batista. - 2025.
76 p.: il. color.

Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Corrente, 2025.

Orientador : Prof Me. Carlos Estevão Bastos Sousa.

1. Análise e Desenvolvimento de Sistemas - estudo e ensino. 2. Visão computacional. 3. Rastreamento visual de objetos . 4. Detecção de objetos. I.Título.

CDD - 004

Elaborado por Tefischer Huanderson Soares e Sousa CRB 3/1251

À minha família, que sempre esteve ao meu lado, e aos professores e amigos que me ajudaram a superar os desafios e a alcançar este objetivo.
Este trabalho é dedicado a vocês.

AGRADECIMENTOS

Agradeço, primeiramente, à minha família pelo apoio constante, pela confiança e por sempre estarem presentes em todos os momentos. À minha mãe, Benilce Ferreira Araújo Batista, e ao meu pai, Adalberto Rodrigues Batista, que sempre acreditaram em mim e me incentivaram a seguir em frente nos estudos, mesmo diante das dificuldades, meu eterno reconhecimento.

Agradeço ao meu orientador e amigo, Carlos Estevão Bastos Sousa, pela paciência, dedicação e orientação ao longo de todo o desenvolvimento deste trabalho. Seu apoio constante, aliado à sua experiência, foi essencial para que eu conseguisse superar os desafios dessa jornada. Sem sua orientação, tanto acadêmica quanto pessoal, este trabalho não teria sido possível.

Aos professores que contribuíram para minha formação acadêmica, expresso minha profunda gratidão. Agradeço especialmente aos professores Felipe Gonçalves dos Santos e Túlio Vidal Rolim, cujas orientações e incentivos não apenas ampliaram meus conhecimentos, mas também despertaram em mim o interesse pela pesquisa que me guiou ao longo desta jornada. Cada ensinamento foi uma peça importante na construção do meu aprendizado e na realização deste trabalho. À amizade e ao conhecimento construído graças a todos vocês, meu muito obrigado.

Aos meus amigos e colegas, em especial Rayline Pereira Mendes, José Wellington Santos Nascimento e Gabriel Rocha Paes da Costa, que estiveram ao meu lado durante toda a jornada acadêmica, meu sincero agradecimento. Compartilhar momentos de aprendizado, troca de ideias e, principalmente, de apoio com vocês tornou essa trajetória mais leve, mesmo nos piores momentos. A convivência, as conversas e o companheirismo foram fundamentais para enfrentar os desafios e celebrar as conquistas ao longo do caminho.

Por fim, agradeço a todos que, de alguma forma, contribuíram para a realização deste trabalho, seja por meio de palavras de incentivo, gestos de apoio ou contribuições diretas. Cada um de vocês teve um papel importante nesta conquista, e meu sincero reconhecimento é dedicado a todos.

RESUMO

O avanço das tecnologias, como a Visão Computacional, tem transformado significativamente a forma como assistimos e analisamos eventos esportivos, otimizando processos internos, como a tomada de decisões por árbitros e a elaboração de estratégias por treinadores. No contexto do voleibol, esporte de alta dinâmica e complexidade tática, a aplicação de técnicas de rastreamento visual da bola enfrenta desafios particulares, decorrentes do tamanho reduzido do objeto, sua elevada velocidade e as frequentes oclusões causadas pelos jogadores. Este trabalho tem como objetivo principal avaliar e comparar o desempenho de diferentes algoritmos de rastreamento visual, como CSRT, KCF, MOSSE, MIL, TLD, MedianFlow e Boosting, em vídeos de jogos de vôlei, com o intuito de identificar as abordagens mais eficazes para a detecção e acompanhamento da trajetória da bola em tempo real. Em adição, também é realizado o treinamento de um modelo do YOLO para investigar como o uso de Redes Neurais Profundas pode contribuir para a detecção precisa e eficiente da bola, considerando as particularidades do voleibol. A avaliação dos algoritmos é realizada com base em métricas, como a Interseção sobre União para medir a precisão do rastreamento, a taxa de *Frames* por Segundo para avaliar o desempenho em tempo real e o Erro Médio de Rastreamento para quantificar os desvios na estimativa da posição da bola. Embora os algoritmos avaliados apresentem limitações em relação ao *dataset* testado, essas restrições estão relacionadas à complexidade dos vídeos de partidas profissionais, obtidos na Internet, no canal do YouTube Volleybal Playoffs, que são utilizados na construção do conjunto de imagens e incluem variações de iluminação, baixa resolução e movimentação rápida da bola. A fim de complementar a análise e avaliar o desempenho dos algoritmos em condições mais controladas, são realizados testes com vídeos gravados em ambiente controlado, utilizando bolas coloridas nas cores verde, laranja e azul, para facilitar a detecção e o acompanhamento da trajetória. A comparação entre os resultados obtidos com o *dataset* elaborado a partir de vídeos de partidas profissionais e os testes realizados no conjunto elaborado com vídeos em ambiente controlado demonstra que as condições controladas contribuem para um melhor desempenho dos algoritmos. Esses dados fornecem uma base preliminar que pode contribuir para futuras pesquisas voltadas ao desenvolvimento de sistemas integrados de apoio à arbitragem e à análise de desempenho esportivo.

Palavras-chaves: Visão Computacional; Rastreamento Visual de Objetos; Vôlei; Detecção de Objetos; Arbitragem Esportiva.

ABSTRACT

The advancement of technologies, such as Computer Vision, has significantly transformed the way we watch and analyze sports events, optimizing internal processes like decision-making by referees and strategy development by coaches. In the context of volleyball, a sport characterized by high dynamics and tactical complexity, the application of ball tracking techniques faces particular challenges due to the small size of the object, its high speed, and frequent occlusions caused by players. This study aims to evaluate and compare the performance of different visual tracking algorithms, such as CSRT, KCF, MOSSE, MIL, TLD, MedianFlow, and Boosting, in volleyball game videos, with the goal of identifying the most effective approaches for real-time ball detection and trajectory tracking. Additionally, a YOLO model is trained to investigate how the use of Deep Neural Networks can contribute to the precise and efficient detection of the ball, considering the specific challenges of volleyball. The evaluation of the algorithms is based on metrics such as Intersection over Union to measure tracking accuracy, Frames Per Second to assess real-time performance, and Mean Tracking Error to quantify deviations in the ball's position estimation. Although the evaluated algorithms exhibit limitations regarding the tested dataset, these restrictions are related to the complexity of the professional match videos obtained from the YouTube channel Volleyball Playoffs, which are used to construct the image set and include variations in lighting, low resolution, and rapid ball movement. To complement the analysis and assess the performance of the algorithms under more controlled conditions, tests are conducted with videos recorded in a controlled environment, using colored balls in green, orange, and blue to facilitate the detection and tracking of the trajectory. The comparison between the results obtained with the dataset created from professional match videos and the tests conducted with the controlled environment video set demonstrates that controlled conditions contribute to better algorithm performance. These findings provide a preliminary foundation that can support future research aimed at developing integrated systems to assist refereeing and sports performance analysis.

Keywords: Computer Vision; Visual Object Tracking; Volleyball; Object Detection; Sports Officiating.

LISTA DE ILUSTRAÇÕES

Figura 1 – Aula de vôlei na ACM.....	20
Figura 2 – Campeonato Mundial de Voleibol de 1952.....	21
Figura 3 – Oclusão de objetos.....	26
Figura 4 – Variação de iluminação.....	27
Figura 5 – Mudança leve de iluminação.....	27
Figura 6 – Mudança de aparência em razão da perspectiva.....	28
Figura 7 – Mudança de aparência em razão de velocidade.....	28
Figura 8 – Exemplo de cenário densamente povoado.....	29
Figura 9 – Funcionamento da operação dos filtros de correlação.....	30
Figura 10 – <i>Pipeline</i> de rastreamento de um filtro de correlação discriminativo.....	31
Figura 11 – <i>Fluxo do processo de treinamento e previsão em Machine Learning</i>	32
Figura 12 – Aprendizado por múltiplas instâncias.....	33
Figura 13 – Erro Forward-Backward no rastreamento de objetos.....	34
Figura 14 – Percurso metodológico.....	36
Figura 15 – <i>Imagens do dataset</i>	37
Figura 16 – <i>Estrutura de pastas do dataset</i>	38
Figura 17 – Imagens dos vídeos de ambiente controlado.....	39
Figura 18 – IoU dos rastreadores ao longo dos <i>frames</i> dos vídeos de partidas profissionais. 44	
Figura 19 – IoU dos rastreadores ao longo dos <i>frames</i> dos vídeos de ambiente controlado. 45	
Figura 20 – Erro Médio de Rastreamento em vídeos de partidas profissionais.....	47
Figura 21 – Erro Médio de Rastreamento em vídeos de partidas com ambiente controlado. 48	
Figura 22 – Análise visual do vídeo de partida profissional.....	49
Figura 23 – Análise visual do vídeo de cenário controlado.....	50
Figura 24 – Curvas de Precisão, Revocação e F1.....	51
Figura 25 – <i>Troca de ID ao longo dos frames</i>	52
Figura 26 – Representação de imagem digital.....	62
Figura 27 – Representação de FPS.....	63
Figura 28 – Etapas convencionais da detecção de objetos.....	64
Figura 29 – Classificação de modelos genéricos de detecção de objetos.....	65
Figura 30 – Diagrama de sequência do <i>pipeline</i> de rastreamento de objetos.....	68
Figura 31 – Representação do processo de IoU.....	78
Figura 32 – Especificações da Quadra de Vôlei.....	82

LISTA DE TABELAS

Tabela 1 – Rastreadores aplicados.....	39
Tabela 2 – Parâmetros principais dos rastreadores KCF e CSRT.....	40
Tabela 3 – Descrição dos vídeos de partidas profissionais.....	43
Tabela 4 – Descrição dos vídeos de partidas amadoras.....	44
Tabela 5 – Médias de FPS por rastreador e resolução.....	46
Tabela 6 – Síntese dos resultados em cenário profissional.....	52
Tabela 7 – Síntese dos resultados em ambiente controlado.....	53

LISTA DE ABREVIATURAS E SIGLAS

ACM	Associação Cristã de Moços
bbox	<i>Bounding Box</i>
CBV	Confederação Brasileira de Voleibol
CF	Filtro de Correlação (<i>Correlation Filter</i>)
CNN	Redes Neurais Convolucionais (<i>Convolutional Neural Network</i>)
CSR-DCF <i>Reliability</i>	<i>Discriminative Correlation Filter with Channel and Spatial</i>
DCF <i>Filter</i>)	Filtro de Correlação Discriminativo (<i>Discriminative Correlation</i>
DL	Aprendizado Profundo (<i>Deep Learning</i>)
FBT	<i>Forward-Backward Tracking</i>
FIVB	Federação Internacional de Voleibol (<i>Fédération Internationale de Volley- ball</i>)
FPS	Frames por Segundo (<i>Frames Per Second</i>)
IoU	Interseção sobre União (<i>Intersection over Union</i>)
KCF <i>Filters</i>) LSTM	Filtros de Correlação Kernelizados (<i>Kernelized Correlation</i> <i>Long Short-Term Memory</i>)
ML	Aprendizado de Máquina (<i>Machine Learning</i>)
MIL <i>Learning</i>)	Aprendizado por Múltiplas Instâncias (<i>Multiple Instance</i>
MOSSE	<i>Minimum Output Sum of Squared Error</i>
MOT <i>Tracking</i>)	Rastreamento de Múltiplos Objetos (<i>Multiple Object</i>
MTE	Erro Médio de Rastreamento (<i>Mean Tracking Error</i>)
OD	Deteção de Objetos (<i>Object Detection</i>)
OpenCV	Biblioteca de Visão Computacional de Código Aberto (<i>Open Source Com- puter Vision Library</i>)
OT <i>Tracking</i>)	Rastreamento de um Único Objeto (<i>Object</i>
px	<i>Pixel</i>
RNNs	Redes Neurais Recorrentes (<i>Recurrent Neural Networks</i>)
ROI	Região de Interesse (<i>Region of Interest</i>)
TLD	Rastreamento, Aprendizado e Deteção (<i>Tracking, Learning, and Detection</i>)
VOT	Rastreamento Visual de Objetos (<i>Visual Object Tracking</i>)

YOLO

You Only Look Once

LISTA DE SÍMBOLOS

%	Porcentagem
©	Copyright
®	Marca registrada
\$	Dólar
∈	Pertence
arg	Argumento do mínimo
min	Mínimo
max	Máximo
	Soma
λ	Letra grega lambda
α	Letra grega alfa
β	Letra grega beta
μ	Letra grega mi
ℓ	Letra grega minúscula l (lambda minúsculo estilizado)
⊙	Multiplicação ponto a ponto
*	Operação de convolução/correlação
sign	Função sinal
F^{-1}	Transformada de Fourier inversa
L	Função de perda
U	União de conjuntos
\	Diferença de conjuntos
$\ \cdot \ $	Norma Euclidiana

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Objetivo Geral.....	18
1.2	Objetivos Específicos.....	18
1.3	Produção Científica.....	18
1.4	Estrutura deste trabalho.....	18
2	CONTEXTUALIZAÇÃO.....	20
2.1	História do Vôlei.....	20
2.2	Estrutura e Regras.....	21
2.3	Funções e Ações do Jogo.....	22
3	FUNDAMENTAÇÃO TEÓRICA.....	23
3.1	Trabalhos Relacionados.....	23
3.2	Rastreamento <i>Online</i> e Rastreamento <i>Offline</i>	25
3.3	Desafios no Rastreamento de Objetos.....	25
3.3.1	Oclusão.....	26
3.3.2	Variação de iluminação.....	26
3.3.3	Mudança de aparência.....	28
3.3.4	Sobreposição e Confusão de Objetos.....	29
3.4	Princípios dos Algoritmos de Rastreamento de Objetos.....	30
3.4.1	Filtro de Correlação (Correlation Filter - CF).....	30
3.4.2	Filtro de Correlação Discriminativo (Discriminative Correlation Filter - DCF)	31
3.4.3	Aprendizado de Máquina (Machine Learning - ML).....	32
3.4.4	Aprendizado por Múltiplas Instâncias (Multiple Instance Learning - MIL)	33
3.4.5	Forward-Backward Tracking (FBT).....	34
4	METODOLOGIA.....	36
4.1	Revisão da Literatura.....	36
4.2	Coleta e Preparação dos Dados.....	37
4.3	Aplicação dos Algoritmos.....	39
4.4	Avaliação dos Algoritmos.....	41
4.5	Execução de Testes.....	41
5	RESULTADOS.....	43
5.1	Equipamentos Utilizados.....	43
5.2	Resultados Quantitativos.....	43
5.2.1	Resultados da Interseção sobre União.....	43

5.2.2	Resultados de Quadros por Segundo (<i>Frames Per Second</i> - FPS)	46
5.2.3	Resultados do Erro Médio de Rastreamento	47
5.3	Análise Visual	48
5.4	Resultados do YOLOv8	51
5.5	Síntese dos Resultados	52
6	CONSIDERAÇÕES FINAIS	55
6.1	Trabalhos Futuros	56
	REFERÊNCIAS	57
	APÊNDICE A – IMAGENS E VÍDEOS DIGITAIS	62
A.1	Imagens e Vídeos Digitais	62
A.1.1	Imagens Digitais	62
A.1.2	Vídeos Digitais	63
	APÊNDICE B – DETECÇÃO DE OBJETOS	64
B.1	Deteção de Objetos (<i>Object Detection</i> - OD)	64
B.1.1	Métodos Convencionais de Deteção de Objetos	64
B.1.2	Métodos Contemporâneos de Deteção de Objetos	65
	APÊNDICE C – RASTREAMENTO DE OBJETOS	67
C.1	Rastreamento de Objetos (<i>Object Tracking</i> - OT)	67
C.1.1	Abordagens Clássicas de Rastreamento de Objetos	67
C.1.2	Abordagens Modernas de Rastreamento de Objetos	69
	APÊNDICE D – ALGORITMOS DE RASTREAMENTO DE OBJETOS	70
D.1	Algoritmos de Rastreamento de Objetos	70
D.1.1	Boosting	70
D.1.2	<i>Discriminative Correlation Filter with Channel and Spatial Reliability</i> (CSR-DCF ou CSRT)	71
D.1.3	<i>Kernelized Correlation Filters</i> (KCF)	72
D.1.4	MedianFlow	73
D.1.5	<i>Multiple Instance Learning</i> (MIL)	74
D.1.6	<i>Minimum Output Sum of Squared Error</i> (MOSSE)	75
D.1.7	<i>Tracking, Learning and Detection</i> (TLD)	76
	APÊNDICE E – MÉTRICAS UTILIZADAS	78
E.1	Métricas Utilizadas	78
E.1.1	Interseção sobre União (<i>Intersection over Union</i> - IoU)	78
E.1.2	Erro Médio de Rastreamento (<i>Mean Tracking Error</i> - MTE)	79

E.1.3	Quadros por Segundo (<i>Frames Per Second</i> - FPS).....	80
E.1.4	Robustez.....	80
ANEXO A – ESPECIFICAÇÕES DA QUADRA DE VÔLEI.....		82

1 INTRODUÇÃO

Os esportes se beneficiam significativamente dos avanços da tecnologia, que possibilitam a automação de diversas tarefas complexas. O uso de técnicas de Visão Computacional (*Computer Vision* - CV), área que busca desenvolver métodos para interpretar e processar informações visuais de forma automatizada, em esportes engloba uma vasta gama de aplicações, como a análise detalhada de jogadas e a classificação de eventos com base no desempenho de jogadores e equipes (NAIK; HASHMI; BOKDE, 2022). Essas tecnologias não apenas transformam como os esportes são assistidos e analisados, mas também podem otimizar processos internos, como a tomada de decisões por árbitros e treinadores assistentes.

Entre os diversos esportes que podem se beneficiar dessas inovações, o voleibol se destaca pela alta dinâmica e pela complexidade das interações entre jogadores e bola. Devido às movimentações rápidas e às constantes mudanças na trajetória do objeto, o rastreamento automatizado da bola representa um grande desafio. Além disso, fatores como a velocidade do jogo, o tamanho reduzido da bola e as frequentes oclusões causadas pelos jogadores dificultam a aplicação de técnicas tradicionais de detecção e acompanhamento (HAN; WANG; WANG, 2024). Assim, torna-se essencial investigar métodos mais eficazes para lidar com esses desafios.

De acordo com Garcia, Meireles & Pereira (2021), o voleibol tem-se consolidado como um dos esportes mais praticados mundialmente, abrangendo diversas faixas etárias. Sua popularidade deve-se, em grande parte, à clareza de suas regras, fator que contribui para sua ampla aceitação e disseminação global. Ademais, devido à dinâmica acelerada e à complexidade estratégica da modalidade, a aplicação de tecnologias avançadas apresenta um grande potencial para aprimorar a precisão e a eficiência na tomada de decisões durante as partidas.

Diante dos desafios inerentes ao rastreamento da bola no voleibol, o Rastreamento Visual de Objetos (*Visual Object Tracking* – VOT) emerge como uma abordagem essencial dentro da CV, dedicada a estimar a trajetória e o estado de um alvo ao longo de uma sequência de imagens (JAVED *et al.*, 2022). O VOT pode ser aplicado tanto ao rastreamento de um único objeto (*Object Tracking* – OT) quanto ao de múltiplos objetos (*Multiple Object Tracking* – MOT), desempenhando um papel central na interpretação automatizada de cenários dinâmicos, conforme discutido por Nunes (2023).

Com aplicações em diversas áreas, como análise esportiva, videovigilância, estudo de movimentos, navegação de veículos autônomos, monitoramento de tráfego e controle de robôs móveis (CHEN *et al.*, 2022), o VOT possui um grande potencial para transformar a forma como interagimos com o ambiente ao nosso redor através de vídeos e imagens digitais, conceitos apresentados no Apêndice A. No contexto esportivo, sua implementação pode aprimorar significativamente a análise e avaliação de partidas, permitindo um acompanhamento mais preciso e automatizado dos eventos em tempo real. Adicionalmente, a evolução das técnicas de rastreamento tem desempenhado um papel crucial no desenvolvimento de sistemas de auxílio ao esporte, tornando-os cada vez mais precisos e robustos.

Nesse cenário, as abordagens convencionais de VOT concentram-se em algoritmos que utilizam modelos baseados em aparências, aprendizado online ou uma combinação de ambos (ZOU *et al.*, 2023). Com os avanços na CV e no Aprendizado de Máquina (*Machine Learning* - ML), surgiram métodos mais sofisticados que integram detecção e rastreamento em um único processo (JAVED *et al.*, 2022). Informações adicionais sobre o funcionamento da detecção e as diferenças entre esses modelos são apresentadas nos Apêndices B e C, respectivamente.

Outras tecnologias também desempenham um papel relevante na análise de jogos, destacando-se, por exemplo, o uso de chips que, conforme descrito por Ghosh *et al.* (2023), permitem monitorar dados específicos de objetos, como velocidade e aceleração, sendo uma alternativa para o rastreamento de elementos pontuais. No entanto, a aplicação de chips é restrita a dados isolados, sem capturar o contexto global da partida, como o posicionamento e a interação entre jogadores.

Em contrapartida, ainda segundo *Ghosh et al.* (2023), a CV possibilita uma análise mais completa e contextualizada, registrando movimentos e estratégias em um escopo amplo e dinâmico. Sistemas baseados em CV utilizam algoritmos avançados capazes de interpretar as dinâmicas do jogo em diferentes camadas de interação. Embora demandem um investimento inicial elevado e uma infraestrutura computacional robusta, sua flexibilidade e precisão fazem dela uma ferramenta valiosa para a análise aprofundada de esportes.

Apesar dos avanços da CV, o rastreamento visual no voleibol permanece desafiador devido à alta velocidade e às constantes mudanças de trajetória da bola. Segundo Kaur & Singh (2023), embora o VOT tenha evoluído com o avanço dos algoritmos de Aprendizado Profundo (*Deep Learning* - DL), do poder computacional e de novas tecnologias, ainda enfrenta desafios clássicos, como variações na aparência do alvo, mudanças de iluminação, oclusões, instabilidade da câmera e limitações na capacidade de processamento dos rastreadores. Superar essas dificuldades é fundamental para que as análises automatizadas possam fornecer dados precisos e confiáveis, permitindo uma avaliação mais eficaz e uma tomada de decisão mais assertiva durante as partidas.

Diante desse cenário, a necessidade de um sistema de rastreamento preciso e adaptável torna-se ainda mais crítica em esportes como o vôlei, onde rápidas mudanças de cenário e complexas interações entre jogadores e bola podem dificultar uma avaliação eficiente. Em uma partida de vôlei, a bola pode mudar de direção em alta velocidade e ser ocluída por jogadores em movimento, tornando essencial que o sistema de rastreamento lide com essas rápidas mudanças e com a variabilidade nas condições de iluminação e ângulos de câmera.

Ademais, um sistema de rastreamento preciso oferece benefícios substanciais para a arbitragem e a análise de desempenho, garantindo decisões mais justas, enquanto fornece dados valiosos para aprimorar o treinamento e as estratégias dos jogadores, contribuindo para uma evolução contínua no esporte.

Dessarte, torna-se necessário avaliar como diferentes abordagens de OT podem lidar com as complexidades específicas dos cenários de uma partida de vôlei. Isso envolve a análise de como algoritmos clássicos, apresentados no Apêndice D, e técnicas de DL podem ser adaptados para aprimorar a detecção e o acompanhamento da bola e dos jogadores em tempo real, além de investigar como essas abordagens podem superar desafios como oclusões e variações de iluminação.

1.1 OBJETIVO GERAL

O objetivo deste trabalho é avaliar o desempenho de diferentes algoritmos de rastreamento visual para identificar aqueles mais adequados à análise de partidas de vôlei. A pesquisa busca determinar as abordagens que oferecem maior precisão e adaptabilidade, possibilitando aprimoramentos na análise de desempenho e na obtenção de informações detalhadas e em tempo real.

1.2 OBJETIVOS ESPECÍFICOS

1. Revisar a literatura existente sobre técnicas de OT;
2. Adquirir vídeos de partidas profissionais de vôlei;
3. Construir um *dataset* com vídeos de partidas profissionais de vôlei;
4. Adquirir um *dataset* com vídeos de partidas amadoras em ambiente controlado;
5. Implementar e testar algoritmos de OT;
6. Treinar algoritmos de DL;
7. Analisar as limitações e capacidades dos algoritmos clássicos de rastreamento visual no vôlei;
8. Investigar o desempenho de técnicas baseadas em DL na detecção e OT, com ênfase na bola;
9. Avaliar a influência de fatores como oclusão, variações de iluminação e alta velocidade

dos alvos no desempenho dos algoritmos de rastreamento.

1.3 PRODUÇÃO CIENTÍFICA

A produção científica desenvolvida ao longo deste trabalho inclui a seguinte publicação:

- BATISTA, L. N. F.; MENDES, R. P.; SOUSA, C. E. B. Estudo comparativo de algoritmos clássicos de rastreamento visual em vôlei. In: VIII Congresso Internacional de Gestão e Tecnologias, 2024, Teresina-PI. Anais VIII Cointer PDVGT 2024, 2024.

1.4 ESTRUTURA DESTE TRABALHO

Este trabalho está estruturado em seis capítulos distintos. Neste capítulo, denominado Introdução, são abordados os objetivos da pesquisa, a relevância do estudo e a estrutura do trabalho. No Capítulo 2, denominado Contextualização, é apresentada uma breve abordagem sobre a história e as regras do vôlei. No Capítulo 3, intitulado Fundamentação Teórica, são discutidos os conceitos e as teorias que sustentam a pesquisa. No Capítulo 4, denominado Metodologia, são descritos os procedimentos e as técnicas utilizadas para a realização da pesquisa. No Capítulo 5, intitulado Resultados, são apresentados e analisados os resultados obtidos ao longo do estudo. Por fim, no Capítulo 6, denominado Considerações Finais, são elencadas as conclusões sobre o trabalho, bem como sugestões para futuras pesquisas.

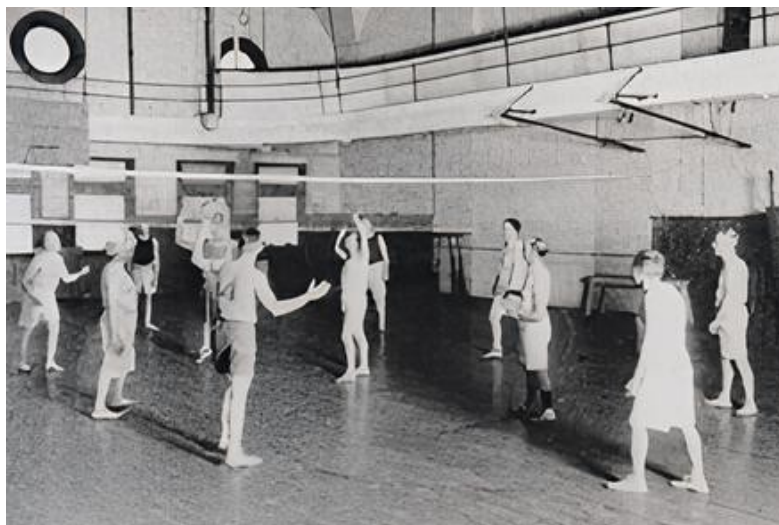
2 CONTEXTUALIZAÇÃO

Neste capítulo, são abordados os principais aspectos do voleibol, com foco em sua evolução histórica, desde sua criação em 1895 até sua consolidação como esporte olímpico. Também são discutidas a estrutura do jogo, incluindo quadra, rede e equipamentos, as regras fundamentais, como sistema de pontuação e funções dos jogadores, e os fundamentos técnicos, como saque, levantamento, ataque e bloqueio. O objetivo é oferecer uma visão clara e organizada dos elementos que compõem essa modalidade esportiva.

2.1 HISTÓRIA DO VÔLEI

O voleibol surgiu no final do século XIX, nos Estados Unidos, criado por William George Morgan em 1895, na Associação Cristã de Moços (ACM) de Holyoke, Massachusetts. Inicialmente chamado de *mintonette*, o esporte foi concebido como uma alternativa de atividade física menos agressiva do que o basquete, voltada para homens de meia-idade. A popularidade da prática cresceu rapidamente, sendo renomeada como *volleyball* durante uma conferência da ACM, graças à característica de manter a bola em constante movimento sobre a rede (*volley*). Na Figura 1 é ilustrada uma partida de vôlei realizada no local de seu surgimento.

Figura 1: Aula de vôlei na ACM.



Fonte: Fédération Internationale de Volleyball (2025).

Segundo Mezzaroba & Pires (2011), a expansão global do voleibol foi impulsionada pela ACM, que disseminou o esporte para diversos países no início do século XX, como Canadá, Cuba, Filipinas e Brasil. Em 1947, a criação da Federação Internacional de Voleibol (*Fédération Internationale de Volleyball* - FIVB) contribuiu para sua organização em competições internacionais, culminando na inclusão do voleibol nos Jogos Olímpicos a partir de 1964. Na Figura 2 é ilustrado um momento histórico do Campeonato Mundial de Voleibol de 1952, evidenciando a consolidação do esporte em competições globais.

Figura 2: Campeonato Mundial de Voleibol de 1952.



Fonte: Wikipédia (2025).

No Brasil, o vôlei chegou por volta de 1915, com registros apontando sua introdução pela A

CM em São Paulo (NOGUEIRA, 2024). Inicialmente praticado de forma amadora, o esporte sofreu um impulso significativo a partir da década de 1960, com a organização da Confederação Brasileira de Voleibol (CBV). Desde então, consolidou-se como uma das modalidades mais populares no país, sendo considerado o segundo esporte nacional, superado apenas pelo futebol.

Ao longo das décadas, o vôlei passou por adaptações, incluindo mudanças em suas regras para adequação às transmissões televisivas e maior dinamismo dos jogos. Essas transformações contribuíram para a espetacularização da modalidade, promovendo seu consumo como entretenimento e sua consolidação como fenômeno cultural e esportivo no Brasil.

2.2 ESTRUTURA E REGRAS

O voleibol é um esporte coletivo praticado em uma quadra retangular de 18 metros de comprimento por 9 metros de largura, detalhada na figura do Anexo A, dividida por uma rede central. Cada equipe é composta por seis jogadores em quadra e até seis reservas. O objetivo do jogo é fazer com que a bola toque o solo do lado adversário, marcando pontos, enquanto se evita que o mesmo aconteça no próprio lado.

A quadra é dividida por uma linha central, separando os campos de cada equipe. Sobre esta linha, encontra-se uma rede com altura de 2,43 m para competições masculinas e 2,24 m para femininas (Confederação Brasileira de Voleibol, 2021). A área próxima à rede é delimitada por uma linha de 3 metros, separando a zona de ataque da zona de defesa. Todas as linhas fazem parte da área de jogo, e uma bola que toque uma linha é considerada válida.

No que diz respeito às regras, cada equipe deve organizar seus jogadores em duas fileiras, com três atletas posicionados na zona de ataque, área próxima à rede, e três na zona de defesa, na parte traseira da quadra, conforme estabelecido pela Confederação Brasileira de Voleibol (2021) no regulamento oficial do voleibol. A formação inicial determina a ordem de rotação dos jogadores, que se move no sentido horário toda vez que a equipe recupera o direito de sacar. As posições dos jogadores em quadra são numeradas da seguinte maneira, com três jogadores ao longo da extensão da rede formando a linha de frente, que ocupa as posições 4 (frente-esquerda), 3 (frente-central) e 2 (frente-direita). Os três restantes formam a linha de trás, ocupando as posições 5 (de trás-esquerda), 6 (de trás-central) e 1 (de trás-direita) (Confederação Brasileira de Voleibol, 2021). O sistema de pontuação é baseado em sets de 25 pontos. Para vencer um set, é obrigatória uma vantagem mínima de dois pontos sobre a equipe adversária.

Caso a partida seja decidida no quinto set, chamado de *tie-break*, este será disputado até que uma das equipes alcance 15 pontos, também com a exigência de uma diferença mínima de dois pontos. Durante o jogo, cada equipe pode realizar até três toques na bola antes de enviá-la ao campo adversário, sendo vedado que o mesmo jogador toque a bola consecutivamente. Infrações como tocar na rede, exceder o limite de toques permitidos ou executar o saque de maneira incorreta resultam na perda do ponto.

2.3 FUNÇÕES E AÇÕES DO JOGO

Conforme discutido em *Hartmann et al.* (2022), os jogadores desempenham papéis distintos durante o jogo, de acordo com suas posições e especializações. O líbero é um especialista em recepção e defesa, utiliza uniforme diferenciado e não pode atacar ou sacar, tendo como principal função reforçar o sistema defensivo da equipe. Os atacantes e bloqueadores, posicionados na linha de frente, são responsáveis por realizar ataques potentes para marcar pontos e bloquear investidas adversárias. O levantador organiza as jogadas ofensivas, posicionando a bola estrategicamente para os atacantes.

Durante o jogo, diversas ações técnicas se destacam. O saque é a ação que inicia o *rally* e

pode ser realizado de diferentes formas, como saque flutuante ou viagem ao fundo do mar, com o objetivo de dificultar a recepção adversária. A recepção é o primeiro contato com a bola, geralmente feita com manchete, e o levantamento prepara o ataque, permitindo ao atacante finalizar a jogada. O ataque consiste em enviar a bola ao campo adversário com força e precisão e, entre as variações, destacam-se a cortada, a largada e a exploração do bloqueio adversário. A defesa é fundamental para evitar que a bola toque o solo no próprio campo, com movimentos como peixinho, rolamento e manchete.

A seguir, no Capítulo 3, são apresentados os conceitos teóricos e os trabalhos relacionados que embasam esta pesquisa, destacando os principais desafios e abordagens no campo do rastreamento de objetos.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são explorados os trabalhos relacionados, os principais desafios do VOT em vídeos digitais, tipos de rastreamento, bases nas quais os algoritmos implementados se apoiam, com enfoque específico na aplicação ao voleibol.

3.1 TRABALHOS RELACIONADOS

Os estudos mais recentes em VOT têm-se concentrado amplamente no rastreamento de múltiplos alvos, enquanto as pesquisas focadas no rastreamento de um único objeto frequentemente exploram técnicas baseadas em Aprendizagem Profunda ou realizam comparações entre algoritmos com abordagens similares. Entre essas iniciativas, destacam-se aquelas direcionadas ao avanço da área em aplicações específicas, como no contexto esportivo. Esses trabalhos não apenas examinam métodos existentes, mas também propõem métricas de avaliação e adaptações para superar desafios recorrentes, incluindo oclusões, variações de iluminação e mudanças na aparência dos objetos rastreados.

Brdjanin et al. (2020) realizaram uma avaliação dos rastreadores de objeto único disponíveis na Biblioteca de Visão Computacional de Código Aberto (*Open Source Computer Vision Library* - OpenCV), proposta por Bradski (2000), focando em seu desempenho sob diferentes condições. Utilizando o conjunto de dados OTB-100 (WU; LIM; YANG, 2013), a avaliação foi realizada com dois métodos principais, a Avaliação de Uma Passagem (*One-Pass Evaluation* - OPE) e a Avaliação da Taxa de Sucesso (*Success Rate Evaluation* - SRE). Os resultados foram organizados em três categorias principais, sendo elas a taxa de sucesso, velocidade e precisão. Entre os rastreadores avaliados, Boosting (*FREUND; SCHAPIRE et al.*, 1996) e MIL (BABENKO; YANG; BELONGIE, 2010) apresentaram as maiores taxas de sucesso, enquanto MOSSE (*BOLME et al.*, 2010) e MedianFlow (*KALAL; MIKOLAJCZYK; MATAS*, 2010) destacaram-se por sua velocidade.

O rastreador CSRT (*LUKEZIC et al.*, 2017) demonstrou alta precisão, avaliada com base em dois critérios principais, o Gráfico de Precisão (*Precision Plot* - PP) e Gráfico de Sucesso (*Success Plot* - SP). No Gráfico de Precisão, esta foi medida pela distância euclidiana entre o centro do objeto rastreado e o centro do objeto real (*ground truth*), considerando-se um limiar de 50 *pixels* (px). O CSRT destacou-se por apresentar uma baixa distância média entre os centros, indicando menor desvio (*drift*). Já no Gráfico de Sucesso, a precisão foi avaliada pela sobreposição entre o retângulo delimitador do objeto rastreado e o *ground truth*, utilizando a métrica de Interseção Sobre União (*Intersection Over Union* - IoU). Apesar de não estar diretamente relacionado ao contexto esportivo, este trabalho fornece uma visão relevante sobre a avaliação de algoritmos de VOT em cenários variados, destacando aspectos fundamentais que podem ser aplicados em estudos mais específicos, como o rastreamento no voleibol.

Gudauskas & Matusevičius (2021) realizaram um estudo sobre técnicas de MOT aplicadas à análise de esportes paralímpicos, com foco no *goalball*, esporte paralímpico desenvolvido para pessoas com deficiência visual, no qual os jogadores, utilizando vendas nos olhos, tentam arremessar uma bola com guizos sonoros no gol adversário enquanto se orientam pelo som. O trabalho comparou diferentes métodos de rastreamento, incluindo algoritmos clássicos, como Boosting, CSR-DCF, KCF, MOSSE e TLD, além de abordagens baseadas em Redes Neurais Convolucionais (*Convolutional Neural Network* - CNN). Os experimentos demonstraram que algoritmos clássicos, como MOSSE e KCF, apresentam resultados competitivos. O MOSSE destacou-se como o algoritmo mais rápido, enquanto o KCF obteve maior precisão no rastreamento de jogadores em determinados cenários.

Os autores concluíram que, embora os métodos baseados em Redes Neurais apresentem resultados promissores, os algoritmos clássicos ainda desempenham um papel importante, especialmente em termos de velocidade e simplicidade de implementação. Além disso, os resultados sugerem que a combinação de técnicas clássicas e modernas pode ser uma direção promissora para melhorar o rastreamento de múltiplos objetos em vídeos esportivos.

Huang (2023), propõe em seu trabalho uma variante da estrutura *You Only Look Once* versão 4 (YOLOv4), modelo de detecção de objetos em tempo real amplamente utilizado por sua rapidez e precisão (JOCHER; CHAURASIA; QIU, 2023), especificamente adaptada para prever e analisar trajetórias de voleibol em sequências de vídeo. A estrutura utiliza um modelo YOLOv4 pré-treinado para identificar regiões propostas com altas pontuações de confiança, essencial para a detecção precisa de bolas de vôlei em movimento rápido. Incorporando camadas de convolução gráfica, o modelo agrega características profundas de forma eficaz, permitindo capturar a complexa dinâmica do movimento do vôlei.

Em complemento, os autores introduziram uma nova função de perda que modela a área alvo utilizando uma distribuição gaussiana, melhorando significativamente a detecção e localização de alvos pequenos que frequentemente mudam de tamanho durante o jogo. Os resultados experimentais demonstraram que essa abordagem não apenas melhora a precisão na detecção do vôlei, mas também aborda desafios como oclusão e variações de escala da bola durante as partidas. De forma geral, a estrutura proposta do GE-YOLO introduz avanços promissores na análise de vídeos esportivos, especialmente no aprimoramento da estimativa e detecção de trajetórias no voleibol.

Enquanto nos trabalhos anteriores são abordados algoritmos e técnicas de OT, Silva (2021) apresenta uma perspectiva aplicada, destacando o uso da tecnologia na análise de desempenho no voleibol, com foco na importância de ferramentas tecnológicas para otimizar resultados e aprimorar o desenvolvimento dos atletas.

No estudo de Silva (2021) é discutido o uso da tecnologia na análise do desempenho do voleibol, destacando pontos-chave. O estudo sublinha a importância da análise de desempenho, indicando que ferramentas como sistemas de suporte se tornaram essenciais para otimizar resultados e aprimorar o desenvolvimento individual dos atletas. No trabalho, é abordado o contexto histórico, observando o aumento significativo na demanda por análises de desempenho desde o simpósio da FIVB em 1981, que promoveu o uso dessas análises no esporte.

Por fim, Silva (2021) discute o desenvolvimento tecnológico, focando em três fatores críticos para a adoção de inovações tecnológicas: custo, facilidade de uso e confiabilidade. Na pesquisa, é enfatizado o valor da coleta de dados através de vários meios, como vídeo e questionários, para fornecer uma base mais precisa para a tomada de decisões pela equipe técnica. Finalmente, as descobertas demonstram que o uso de *insights* baseados em dados pode impactar significativamente as decisões táticas durante os jogos, permitindo que os treinadores façam escolhas informadas em vez de depender apenas da experiência empírica. Em última análise, no estudo, é ressaltado o papel transformador da tecnologia na análise do desempenho do voleibol e na melhoria dos processos de tomada de decisão.

3.2 RASTREAMENTO *ONLINE* E RASTREAMENTO *OFFLINE*

O rastreamento de objetos pode ser realizado em dois paradigmas principais (LUO *et al.*, 2021): rastreamento *online* e rastreamento *offline*. Ambos com vantagens e desvantagens, devendo ser escolhidos de acordo com a aplicação e as restrições do sistema.

No rastreamento *online*, também chamado de rastreamento sequencial, as decisões são tomadas em tempo real, com base nas informações disponíveis até o momento atual (LIU; CAESAR, 2023). Esse método é amplamente utilizado em sistemas que exigem respostas imediatas, como veículos autônomos, videomonitoramento e aplicações de segurança (ZHIHUI; SIJING, 2021). Uma das principais vantagens do rastreamento *online* é a baixa latência, permitindo a interação em tempo real. No entanto, essa abordagem pode ser limitada em termos de precisão, pois não utiliza informações futuras para corrigir erros ou melhorar a continuidade do rastreamento.

Por outro lado, o rastreamento *offline* realiza a análise de forma retrospectiva, utilizando todos os dados disponíveis para gerar resultados mais robustos e precisos (FU *et al.*, 2024). Esse método é comum em aplicações como produção cinematográfica, análises esportivas (YARDIMCI; TEKEREK, 2022) e estudos científicos, onde a precisão é priorizada em detrimento da velocidade de processamento. A possibilidade de revisar e corrigir erros com base em dados futuros permite maior confiabilidade nos resultados. Contudo, a necessidade de processar todos os dados previamente limita seu uso em cenários de tempo real.

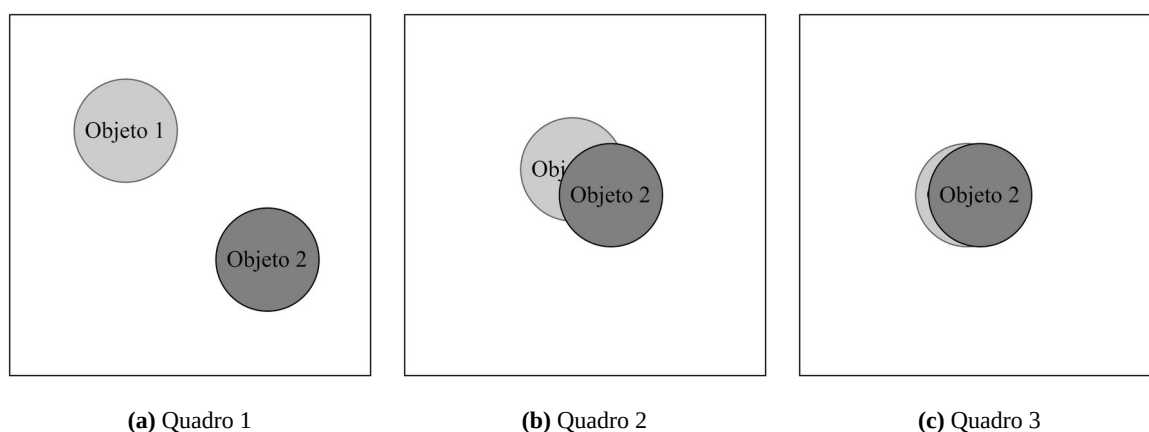
3.3 DESAFIOS NO RASTREAMENTO DE OBJETOS

O OT é uma tarefa fundamental em várias aplicações de CV, como vigilância, monitoramento de tráfego, análise de atividades esportivas e interação homem-máquina. No entanto, essa tarefa enfrenta inúmeros desafios que podem comprometer a precisão e a robustez dos algoritmos de rastreamento. Nesta seção, discutiremos alguns dos principais desafios, incluindo a oclusão, a mudança de aparência, variação de iluminação e sobreposição e confusão de objetos.

3.3.1 Oclusão

A oclusão, que ocorre quando um objeto de interesse está total ou parcialmente coberto por outro objeto no mesmo quadro (BASHAR *et al.*, 2022), é uma causa principal para trocas de ID e fragmentação de trajetórias (LUO *et al.*, 2021). Esse fenômeno dificulta o rastreamento em cenários com muitos objetos, como em partidas esportivas. Para mais, oclusões temporárias, onde o objeto é parcial ou totalmente bloqueado por outros elementos na cena, exemplificadas na Figura 3, podem levar a perdas de rastreamento e à necessidade de algoritmos robustos que recuperem o objeto após a oclusão.

Figura 3: Oclusão de objetos.



Fonte: Elaborado pelo autor (2025).

Na Figura 3, são ilustrados três quadros que exemplificam diferentes níveis de oclusão entre objetos. Na Subfigura 3a, os objetos estão distantes, sem sobreposição. Na Subfigura 3b, os objetos estão próximos, com o objeto 1 parcialmente ocluído pelo objeto 2, mas ainda visível de forma independente. Já na Subfigura 3c, o objeto 1 está quase totalmente sobreposto pelo objeto 2, demonstrando uma oclusão quase total.

O processo de recuperação após a oclusão é crucial para manter a integridade das trajetórias dos objetos. A eficácia desses métodos varia de acordo com a complexidade da cena e a natureza das oclusões, sendo que soluções mais avançadas, segundo *Bashar et al. (2022)*, frequentemente combinam múltiplas abordagens, como modelos baseados em previsão e a reconstrução de trajetórias, para melhorar a precisão e a robustez do rastreamento.

3.3.2 Variação de iluminação

As variações de iluminação, demonstradas na Figura 4, tendem a ocorrer com a mudança de luz no cenário e nos objetos alvo (*CHEN et al., 2022*), afetando significativamente a aparência dos mesmos. Essas mudanças podem resultar de várias fontes, como alterações nas condições de iluminação natural ao longo do dia, mudanças na iluminação artificial, sombras projetadas por outros objetos em movimento ou mudanças na orientação e posição do alvo em relação às fontes de luz.

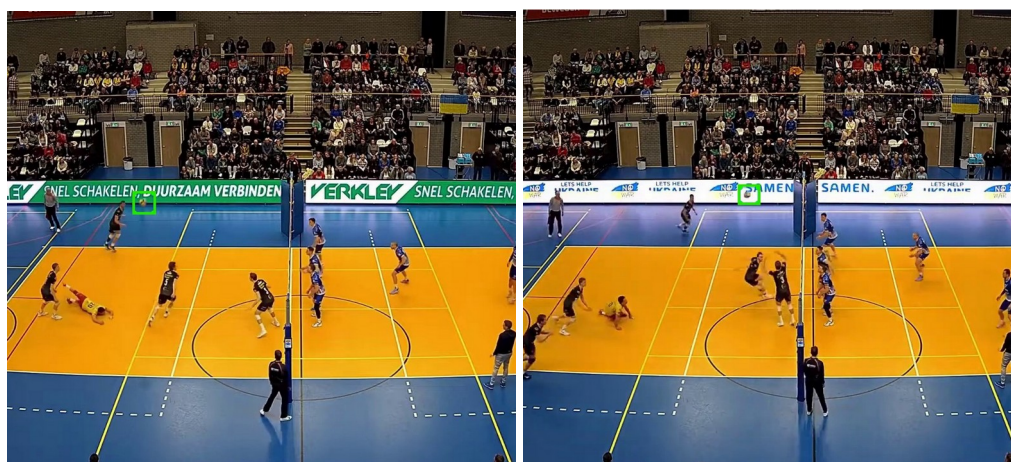
Figura 4: Variação de iluminação.



Fonte: *Liu et al. (2020)*.

É possível observar na Figura 4 como a variação de iluminação afeta os objetos-alvo, podendo dificultar a continuidade do rastreamento. Outro exemplo é apresentado na Figura 5.

Figura 5: Mudança leve de iluminação.



(a) Iluminação normal

(b) Iluminação intensificada

Fonte: Adaptado de Volleybal Playoffs (2024).

Na Figura 5 é ilustrado como diferentes condições de iluminação podem impactar o rastreamento de objetos em vídeos de partidas de vôlei. Na Subfigura 5a, é possível observar a quadra sob condições de iluminação normal, enquanto na Subfigura 5b, há um leve aumento na luminosidade, ocasionado por elementos presentes na cena, como o painel de LED perimetral ao redor da quadra. É importante destacar que o intervalo entre os quadros das Subfiguras é pequeno, o que evidencia a rápida transição de luminosidade.

Essas variações podem introduzir desafios consideráveis na tarefa de reconhecimento visual e, por consequência, no rastreamento de objetos, uma vez que os algoritmos precisam ser capazes de se adaptar a essas mudanças.

3.3.3 Mudança de aparência

A maioria dos alvos muda sua aparência durante o movimento, assim como ilustrado na Figura 6. Isso ocorre devido a questões como a velocidade do alvo, mudanças de iluminação, variações na perspectiva e deformações do objeto. Essas mudanças podem gerar desafios significativos para os algoritmos de rastreamento, pois a variação na aparência pode dificultar a identificação e a continuidade do rastreamento do objeto ao longo do tempo. Portanto, conforme *Ali et al. (2016)*, é crucial que os métodos de rastreamento consigam adaptar-se a essas mudanças dinâmicas na aparência do alvo para garantir um rastreamento preciso e contínuo.

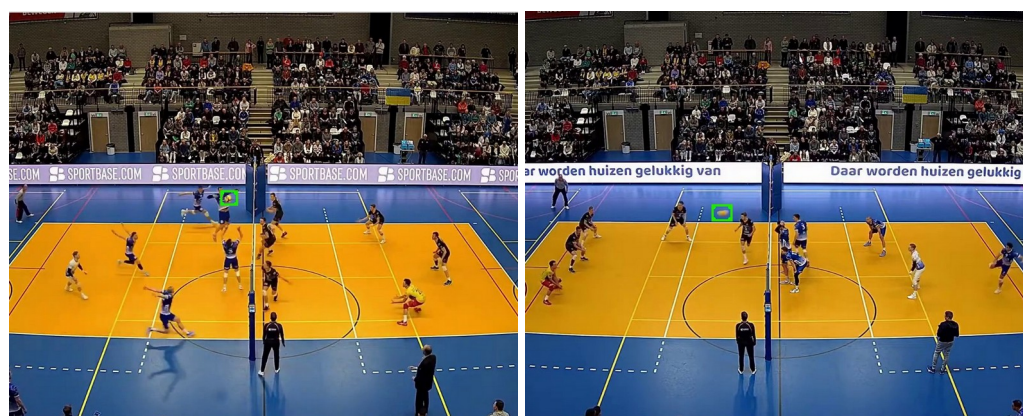
Figura 6: Mudança de aparência em razão da perspectiva.



Fonte: *Liu et al. (2020)*.

Na Figura 6, é retratado como a mudança de perspectiva afeta a aparência de um alvo em movimento. À medida que o objeto se desloca no espaço, a sua orientação em relação à câmera ou ao observador pode variar significativamente, alterando as características visuais que são capturadas. Outro exemplo, mais presente no cenário avaliado neste trabalho, é a mudança de aparência em razão da velocidade do objeto, conforme apresentado na Figura 7.

Figura 7: Mudança de aparência em razão de velocidade.



(a) Aparência normal

(b) Aparência distorcida

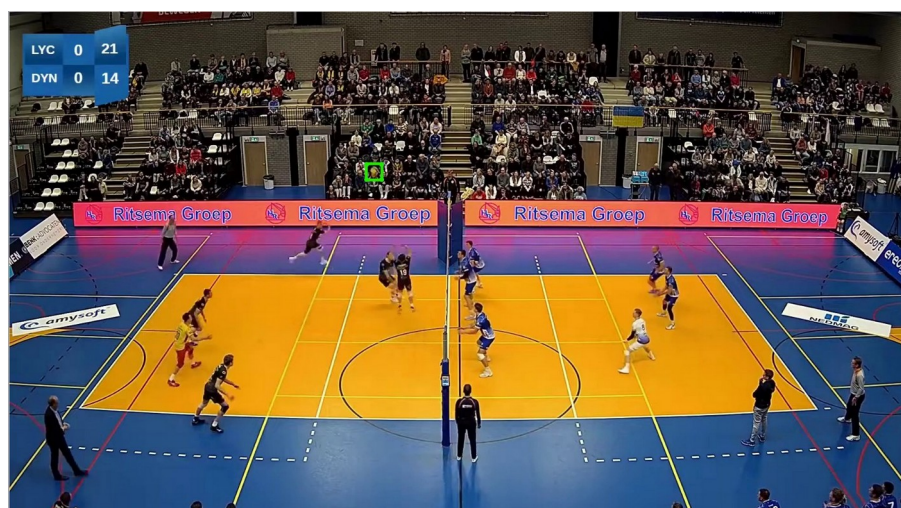
Fonte: Adaptado de Volleybal Playoffs (2024).

Na Figura 7, é possível observar a mudança na aparência da bola em função da velocidade. Na Subfigura 7a, a bola apresenta sua forma normal, enquanto na Subfigura 7b, é possível notar uma distorção moderada, evidenciando os efeitos dinâmicos que influenciam sua aparência e comportamento durante o movimento.

3.3.4 Sobreposição e Confusão de Objetos

Em cenários densamente povoados, a sobreposição de objetos, como ilustrado na Figura 8, pode causar confusão na identificação e rastreamento. Quando objetos estão muito próximos uns dos outros, pode ser difícil distinguir entre eles, especialmente se eles têm características visuais semelhantes. Segundo *Chen et al. (2022)*, o desempenho dos rastreadores em cenas desordenadas tem uma grande lacuna em comparação com o desempenho dos rastreadores em outros desafios. Portanto, algoritmos de rastreamento devem ser capazes de lidar com essa sobreposição e minimizar a confusão para garantir a precisão na detecção e rastreamento de cada objeto individualmente.

Figura 8: Exemplo de cenário densamente povoado.



Fonte: Adaptado de Volleybal Playoffs (2024).

Na Figura 8, é ilustrado um cenário densamente povoado, no qual vários objetos estão próximos uns dos outros. Nesse contexto, a bola, que é o alvo do rastreamento, apresenta características visuais semelhantes às dos objetos ao seu redor e às cores do fundo. Essa semelhança visual dificulta a distinção entre o alvo e os demais elementos, aumentando a probabilidade de falhas no rastreamento, como a perda temporária do objeto ou a identificação incorreta de falsos positivos. Essa complexidade é acentuada pela movimentação dinâmica dos objetos e pelas variações de iluminação, fatores que desafiam a robustez dos algoritmos de rastreamento.

Nesses ambientes, a dificuldade em separar visualmente objetos que se sobrepõem é um dos principais desafios enfrentados por algoritmos de rastreamento, especialmente quando as características dos objetos, como cor, forma ou textura, são semelhantes.

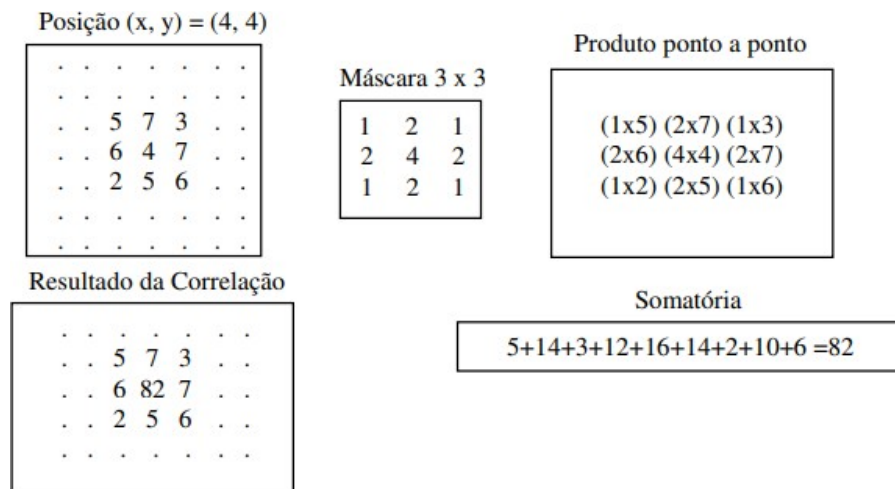
3.4 PRINCÍPIOS DOS ALGORITMOS DE RASTREAMENTO DE OBJETOS

Nesta seção, são apresentadas as principais bases teóricas que fundamentam os algoritmos de OT disponíveis na biblioteca OpenCV. Cada abordagem apresenta características específicas que as tornam mais adequadas para diferentes cenários de aplicação. A seguir, são descritas as bases utilizadas por esses algoritmos.

3.4.1 Filtro de Correlação (Correlation Filter - CF)

Os rastreadores baseados em *Correlation Filters* utilizam técnicas de filtragem no domínio da frequência para realizar o rastreamento dos objetos (BOLME *et al.*, 2010). Esses métodos são rápidos e eficientes, pois aproveitam a Transformada de Fourier para realizar convoluções de forma otimizada. A ideia principal é aprender um modelo discriminativo que maximize a resposta para o objeto de interesse e minimize a resposta para o fundo. Isso é feito ao calcular um filtro que, quando aplicado à imagem, gera um pico de resposta na posição do objeto. Na Figura 9 é ilustrado o funcionamento básico de uma correlação, demonstrando o cálculo ponto a ponto e a somatória resultante.

Figura 9: Funcionamento da operação dos filtros de correlação.



Fonte: Marengoni & Stringhini (2010).

Na Figura 9, é apresentado o processo de correlação, que consiste na aplicação de uma máscara (ou *kernel*) sobre uma imagem. Esse processo é amplamente utilizado em processamento de imagens para realçar características, como bordas, ou para aplicar filtros específicos (BENHAJYOUSSEF; SAIDANI, 2024). A matriz maior, no canto superior esquerdo, representa a imagem original, onde cada elemento corresponde à intensidade de um *pixel*. A máscara 3×3 , localizada no centro superior, é aplicada sobre a imagem, multiplicando cada valor da máscara pelos valores correspondentes dos *pixels* da região processada.

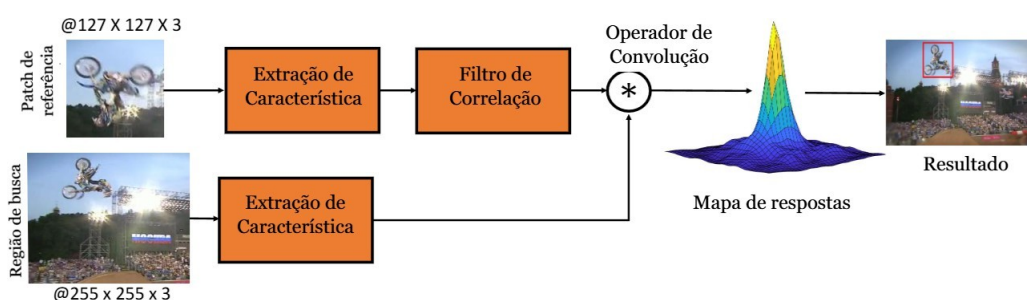
O resultado dessa multiplicação ponto a ponto é mostrado na matriz superior direita, e a soma desses produtos gera o valor 82, que é atribuído ao *pixel* central da região processada. Esse valor é inserido na matriz resultante, como ilustrado no canto inferior esquerdo. O procedimento é repetido para todas as posições possíveis da máscara na imagem, gerando uma nova matriz que reflete a aplicação da correlação. Esse processo é fundamental em diversas operações de processamento de imagens, como detecção de bordas, suavização e realce de características específicas. A escolha da máscara determina o tipo de transformação aplicada à imagem, permitindo uma ampla gama de aplicações.

Métodos baseados em correlação, como KCF e MOSSE, são amplamente utilizados em rastreamento visual devido à sua eficiência computacional. No entanto, apresentam limitações em cenários com mudanças de escala significativas, rotações ou oclusões prolongadas Kumar *et al.* (2024).

O *Discriminative Correlation Filter* (HENRIQUES *et al.*, 2012) é uma extensão dos CF que introduz uma abordagem discriminativa para o VOT, como apresentado na Figura 10. Diferentemente dos métodos tradicionais, o DCF aprende a distinguir o objeto de interesse do fundo de forma mais eficaz, utilizando *features* avançadas, como *Histogram of Oriented Gradients* (HoG) e *kernels* para mapear os dados para um espaço de maior dimensionalidade.

Além disso, ele é computacionalmente eficiente, graças ao uso da Transformada de Fourier, e robusto a desafios como oclusões, mudanças de iluminação e variações na aparência do objeto.

Figura 10: Pipeline de rastreamento de um filtro de correlação discriminativo.



Fonte: Adaptado de Javed *et al.* (2022).

Na Figura 10 é apresentado o processo de rastreamento com DCF que é iniciado com a extração de características de dois elementos fundamentais, o *patch* de referência que contém o objeto de interesse e a região de busca que corresponde a uma área ampliada onde o objeto pode estar situado. Posteriormente, essas características são processadas por um filtro de correlação, que aplica um operador de convolução para estabelecer comparações entre elas. O resultado dessa operação é um mapa de respostas, cuja finalidade é indicar a posição mais provável do objeto dentro da região de busca. Por fim, a localização do objeto no quadro é determinada com base no pico do mapa de respostas, possibilitando a predição precisa de sua posição.

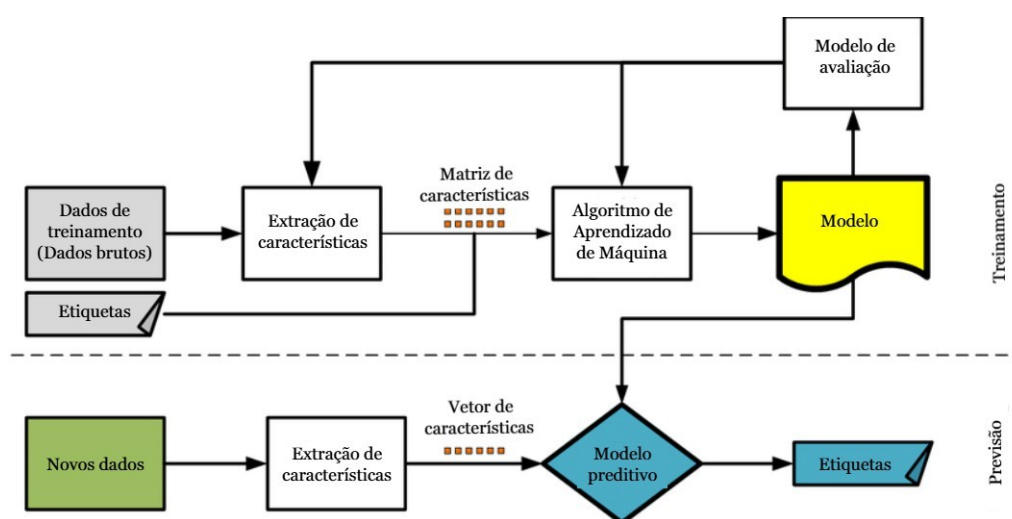
No rastreador CSRT, por exemplo, o DCF é combinado com técnicas de seleção de canais e ponderação espacial (LUKEZIC *et al.*, 2017). Isso significa que o algoritmo avalia a confiabilidade de diferentes partes da imagem e dos canais de características, como cor e textura, atribuindo pesos maiores às regiões mais relevantes para o rastreamento. Para além disso, o CSRT utiliza uma janela de busca adaptativa, que permite lidar melhor com mudanças de escala e movimentos rápidos.

Essa abordagem é particularmente eficaz em cenários desafiadores, como ambientes com alta densidade de objetos, mudanças abruptas de iluminação ou oclusões temporárias. No entanto, o custo computacional é maior em comparação com os filtros de correlação tradicionais, o que pode limitar sua aplicação em dispositivos com recursos limitados.

3.4.3 Aprendizado de Máquina (Machine Learning - ML)

Os rastreadores baseados em *Machine Learning* utilizam algoritmos de ML para construir modelos capazes de distinguir o objeto de interesse do fundo, como ilustrado na Figura 11. Um exemplo dessa abordagem é o rastreador Boosting, que utiliza o algoritmo AdaBoost (FREUND; SCHAPIRE; ABE, 1999) para atualizar o modelo de rastreamento em tempo real.

Figura 11: Fluxo do processo de treinamento e previsão em *Machine Learning*.



Fonte: Adaptado de Nguyen et al. (2017).

O processo apresentado na Figura 11 é dividido em duas etapas principais: treinamento e previsão. Na etapa de treinamento, os dados brutos, acompanhados de suas respectivas etiquetas, que representam os valores ou categorias esperados para cada dado e servem como referência para o aprendizado do modelo, passam por um processo de extração de características, resultando em uma matriz de características. Essa matriz é então utilizada por um algoritmo de ML para construir um modelo, que é avaliado e ajustado iterativamente para melhorar sua precisão.

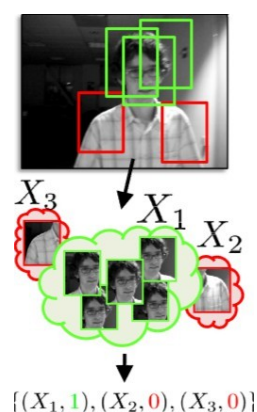
Na etapa de previsão, novos dados são submetidos ao mesmo processo de extração de características, gerando um vetor de características que é alimentado no modelo preditivo. O modelo, por sua vez, realiza a classificação ou previsão, gerando as etiquetas correspondentes. Esse fluxo é amplamente utilizado em aplicações de VOT, onde o modelo precisa ser continuamente atualizado para lidar com mudanças no objeto de interesse, como variações de iluminação, oclusões ou deformações.

Seguindo essa premissa, o AdaBoost funciona combinando vários classificadores fracos, modelos simples, em um classificador forte, que é capaz de distinguir o objeto de interesse com maior precisão. Durante o rastreamento, o modelo é continuamente atualizado com novas amostras do objeto e do fundo, permitindo que ele se adapte a mudanças no ambiente ou no próprio objeto. Embora essa abordagem seja adaptativa e flexível, ela é sensível a ruídos e mudanças abruptas no objeto, como deformações ou rotações. Em adição, o custo computacional pode ser elevado, especialmente em cenários com alta variabilidade ou em aplicações de tempo real.

3.4.4 Aprendizado por Múltiplas Instâncias (Multiple Instance Learning - MIL)

O *Multiple Instance Learning* é uma técnica que considera múltiplas amostras positivas e negativas para treinar um modelo de rastreamento (BABENKO; YANG; BELONGIE, 2010). Diferentemente de abordagens tradicionais, que utilizam apenas uma amostra positiva, o objeto, e várias negativas, o fundo, o MIL trabalha com conjuntos de amostras, como exemplificado na Figura 12, permitindo maior robustez a oclusões parciais e mudanças no objeto.

Figura 12: Aprendizado por múltiplas instâncias.



Fonte: Adaptado de Babenko, Yang & Belongie (2010).

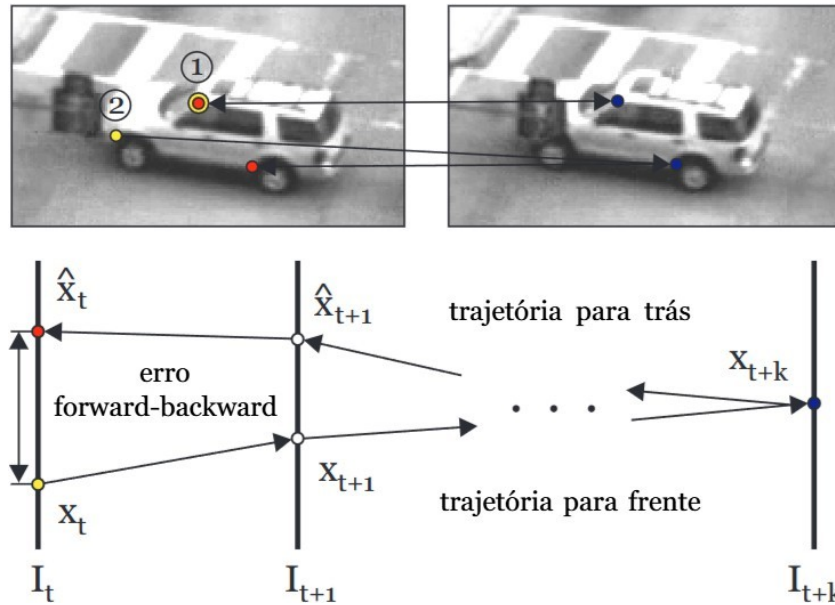
É ilustrado na Figura 12 o conceito de aprendizado por múltiplas instâncias, onde uma imagem é dividida em regiões (caixas verdes e vermelhas) que formam grupos chamados *bags* (X_1 , X_2 , X_3). Cada *bag* contém várias instâncias, e o rótulo da *bag* é determinado pela presença ou ausência de pelo menos uma instância positiva, representada pelas caixas verdes, que contêm o rosto. *Bags* com instâncias positivas são rotuladas como 1, enquanto aquelas sem instâncias positivas são rotuladas como 0.

No rastreador MIL, o modelo é treinado para identificar o objeto de interesse mesmo quando ele está parcialmente ocluído ou quando sua aparência muda devido a fatores como iluminação ou deformações. Isso é feito ao considerar que, dentro de um conjunto de amostras positivas, pelo menos uma delas representa o objeto corretamente. Essa abordagem reduz o impacto de erros de rastreamento temporários, tornando o algoritmo mais robusto em cenários dinâmicos. Apesar de suas vantagens, o MIL pode apresentar dificuldades em cenários com mudanças muito rápidas ou oclusões prolongadas, onde a atualização do modelo pode ser comprometida. Além do mais, o custo computacional é maior em comparação com métodos mais simples, como os filtros de correlação.

3.4.5 Forward-Backward Tracking (FBT)

O FBT é uma técnica que utiliza a consistência entre o rastreamento para frente e para trás no tempo, como exemplificado na Figura 13, para detectar falhas (KALAL; MIKOLAJCZYK; MATAS, 2010). A ideia principal é que, se o rastreamento for realizado corretamente, a posição do objeto ao rastrear para frente e depois para trás deve ser consistente.

Figura 13: Erro Forward-Backward no rastreamento de objetos.



Fonte: Adaptado de Kalal, Mikolajczyk & Matas (2010).

Na Figura 13 é apresentado o conceito de *forward-backward error* no OT, que penaliza trajetórias inconsistentes ao comparar os movimentos para frente (*forward*) e para trás (*backward*). O ponto 1, visível em ambas as imagens, apresenta trajetórias consistentes nos dois sentidos, indicando que o rastreador está funcionando corretamente. Já o ponto 2, que está ocluído na segunda imagem, gera trajetórias inconsistentes entre os movimentos *forward* e *backward*, evidenciando uma falha no rastreamento.

O rastreador MedianFlow é baseado nessa abordagem e utiliza a mediana dos deslocamentos de pontos de interesse para estimar a posição do objeto. Essa técnica é ideal para cenários com movimentos suaves e previsíveis, onde a consistência temporal pode ser mantida. Além disso, o MedianFlow é capaz de detectar falhas no rastreamento, interrompendo o processo quando a incerteza é muito alta.

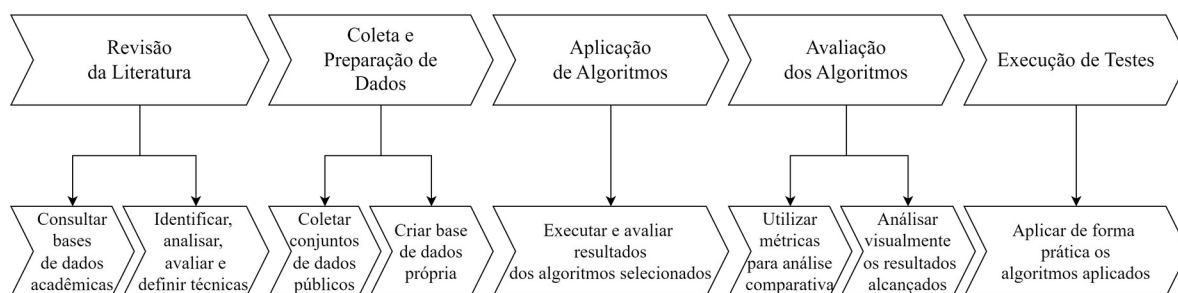
No entanto, o *Forward-Backward Tracking* apresenta limitações em cenários com movimentos rápidos ou abruptos, onde a consistência temporal é difícil de manter. Ademais, a técnica depende da qualidade dos pontos de interesse detectados, o que pode ser um desafio em ambientes com baixa textura ou iluminação variável.

Adiante, no Capítulo 4, são descritos os procedimentos metodológicos adotados nesta pesquisa, incluindo a revisão da literatura, a coleta e preparação dos dados, a implementação e avaliação dos algoritmos, bem como a execução dos testes.

4 METODOLOGIA

A metodologia adotada nesta pesquisa está estruturada de forma sistemática, conforme apresentada na Figura 14, combinando revisão bibliográfica, aplicação de algoritmos e avaliação experimental.

Figura 14: Percurso metodológico.



Fonte: Elaborado pelo autor (2025).

Na Figura 14 é ilustrado o percurso metodológico seguido, que é iniciado com a Revisão da Literatura, prossegue pela Coleta e Preparação de Dados, Aplicação de Algoritmos, Avaliação dos Algoritmos e Execução de Testes. As etapas foram organizadas de forma a garantir uma análise ampla e consistente, sendo detalhadas ao longo das seções subsequentes deste trabalho.

4.1 REVISÃO DA LITERATURA

A primeira etapa desta pesquisa envolve uma revisão extensiva da literatura científica sobre VOT, com ênfase em aplicações esportivas, priorizando trabalhos com foco no vôlei e outros esportes com bola. A busca foi conduzida em bases de dados acadêmicas de renome, como IEEE Xplore, Google Scholar, ScienceDirect, SpringerLink, entre outras, abrangendo publicações em português e inglês. Foram utilizadas palavras-chave relevantes em ambas as línguas para garantir uma cobertura ampla do tema. Essas palavras incluíram “rastreamento de objetos”, “detecção de objetos”, “visão computacional”, “vôlei”, “esportes”, “algoritmos de rastreamento” e seus equivalentes em inglês, como “*object tracking*”, “*object detection*”, “*computer vision*”, “*volleyball*”, “*sports*”, “*tracking algorithms*”, além de suas combinações. Ademais, foi dada preferência à escolha de trabalhos publicados nos últimos 5 anos.

Esta revisão bibliográfica possui como objetivo identificar as principais técnicas de rastreamento utilizadas em aplicações similares, analisar as vantagens e desvantagens de cada técnica, avaliar o estado da arte em relação à precisão, robustez e eficiência computacional dos algoritmos existentes, e definir os algoritmos que seriam aplicados e comparados nesta pesquisa.

4.2 COLETA E PREPARAÇÃO DOS DADOS

Os conjuntos de dados disponíveis não atendiam de forma satisfatória às necessidades deste projeto, especialmente na detecção de bolas de vôlei em cenários variados. Diante disso, foi desenvolvido um conjunto de imagens próprio, alinhado ao ângulo de visão e à perspectiva específicos do projeto. A inclusão de desafios, como fundos densamente povoados, garante maior representatividade e adequação às condições reais do cenário de aplicação. Essa abordagem não apenas atende às particularidades do projeto, mas também contribui para a originalidade e a qualidade do trabalho.

Para a construção desse conjunto, todos os vídeos utilizados foram obtidos a partir de conteúdos disponíveis publicamente no YouTube®, mais especificamente do canal Volleybal Playoffs (2024). A escolha dessa plataforma se justifica pelo amplo acesso a conteúdos diversificados e pelo grande número de conteúdos oficiais disponíveis, possibilitando a coleta de vídeos sob diferentes condições de filmagem, o que contribui para a variabilidade e robustez da base de dados. Além disso, a utilização de vídeos do YouTube® favorece a replicabilidade dos experimentos por outros pesquisadores.

O conjunto de imagens elaborado reúne 5.209 imagens extraídas dos vídeos, que expõem diversas condições, contendo variações diversas e contextos de quadra. Essas características são essenciais para avaliar o desempenho dos algoritmos de modo amplo. As imagens estão organizadas em formato compatível com o YOLO em sua versão 8 (JOCHER; CHAURASIA; QIU, 2023). Alguns exemplos de imagens do conjunto, com suas respectivas caixas delimitadoras, são apresentados na Figura 15.

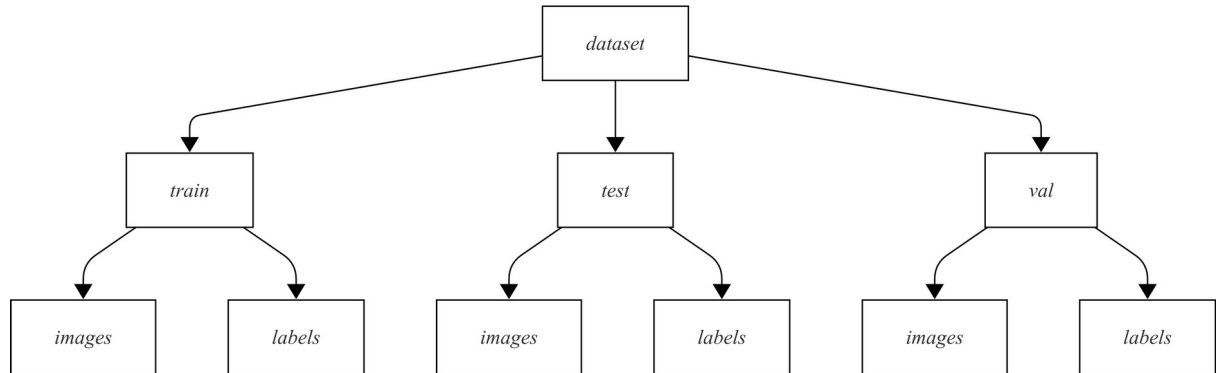
Figura 15: Imagens do *dataset*.



Fonte: Adaptado de Volleybal Playoffs (2024).

Na Figura 15, que exibe algumas das imagens do *dataset*, pode-se observar características importantes dos vídeos utilizados. As imagens apresentam uma câmera fixa, sem variação de ângulo, e são capturadas de uma perspectiva elevada, contemplando toda a quadra de jogo. Adicionalmente, a iluminação uniforme, na maior parte do tempo, e a ausência de obstruções visuais, como objetos em movimento fora do contexto do jogo, favorecem a análise.

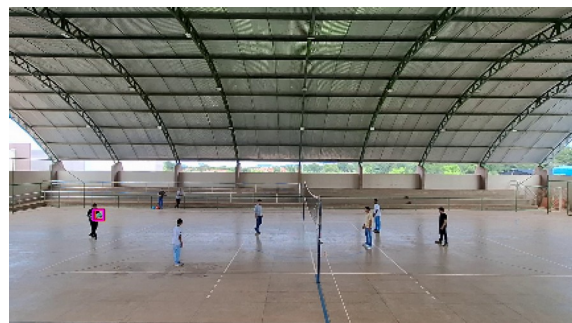
Conforme ilustrado na Figura 16, o *dataset* foi dividido em três partes principais, seguindo a mesma divisão explorada por Pereira (2024) que direciona 80% das imagens para treinamento, 10% para validação e 10% para teste, objetivando garantir uma distribuição equilibrada para as etapas de ajuste e avaliação dos modelos. Cada pasta principal contém duas subpastas: *labels*, com as anotações, e *images*, com as imagens correspondentes. Todas as imagens possuem características uniformes, como resolução de 1920×1080 px e profundidade de cor de 24 bits, assegurando consistência na análise.

Figura 16: Estrutura de pastas do *dataset*.

Fonte: Elaborado pelo autor (2025).

A estrutura apresentada na Figura 16 reflete a organização lógica do conjunto de imagens, facilitando o acesso e o uso eficiente dos recursos durante os experimentos. Ademais, o *dataset* inclui exemplos que desafiam os algoritmos, como deformações da bola causadas pelo movimento e situações em que o objeto se confunde com o fundo. Esses elementos permitem uma avaliação mais detalhada da robustez e precisão dos modelos em condições adversas, contribuindo para uma análise mais confiável e realista.

Por fim, foram realizadas gravações em ambiente controlado para avaliar como fatores específicos, como a cor da bola, podem influenciar o processo de rastreamento. As bolas utilizadas nesses vídeos possuem as cores azul, laranja e verde, permitindo analisar o impacto de diferentes tonalidades no desempenho dos algoritmos. Alguns exemplos dessas gravações são apresentados na Figura 17. Esse conjunto conta com aproximadamente 2.099 imagens extraídas dos vídeos.

Figura 17: Imagens dos vídeos de ambiente controlado.**(a)** Exemplo bola azul**(b)** Exemplo bola verde

Fonte: Elaborado pelo autor (2025).

Os vídeos ilustrados na Figura 17 foram gravados utilizando *smartphones*, com resoluções de 1920×1080 px, Full HD, a 60 FPS e 1280×720 px, HD, a 30 FPS. Esses detalhes técnicos foram utilizados para alinhar-se com os vídeos obtidos para a criação do *dataset*.

4.3 APLICAÇÃO DOS ALGORITMOS

Resultante da revisão da literatura, são selecionados e aplicados os algoritmos de rastreamento, principalmente os clássicos, mais frequentes. Para mais, a disponibilidade de documentação foi um fator crucial para facilitar o processo de aplicação. Os algoritmos selecionados foram aplicados utilizando a linguagem de programação Python e ferramentas de CV, como a OpenCV. Na Tabela 1 são apresentados os rastreadores que foram aplicados neste trabalho.

Tabela 1: Rastreadores aplicados.

Rastreador	Características
CSRT	Alta precisão, adaptativo, utiliza informações de canais de cor e
	confia- bilidade espacial. Ideal para cenários complexos.
KCF	Rápido, não adapta escala, baseado em filtros de correlação. Recomen- dado para aplicações em tempo real.
MOSSE	Muito rápido, baixa precisão, utiliza filtros adaptativos.
Boosting	Adaptativo, sensível a ruídos, baseado em classificador de <i>boosting</i> .
MIL	Robusto a oclusões parciais, utiliza aprendizado de múltiplas instâncias.
TLD	Indicado para cenários com oclusões. Recuperação após oclusão, combina rastreamento, detecção e aprendi- zado. Ideal para objetos que desaparecem e retornam.
MedianFlow	Detecção de falhas, utiliza fluxo óptico. Útil para sistemas que exigem detecção de falhas.
YOLOv8	Alta precisão e velocidade, é principalmente um detector de objetos, mas também combina detecção com rastreamento em tempo real, utilizando os rastreadores BoT-SORT ou ByteTrack.

Fonte: Elaborado pelo autor (2025).

Na Tabela 1, são sintetizadas as principais características dos algoritmos de rastreamento aplicados. O YOLOv8, versão selecionada por ser a mais otimizada, é um modelo baseado em CNNs que utiliza uma abordagem de detecção de objetos em tempo real, dividindo a imagem em regiões e prevendo *bounding boxes* e classes simultaneamente. O modelo foi treinado com uma parte do conjunto de imagens e aplicado em conjunto com o rastreador ByteTrack, rastreador padrão utilizado pelo YOLO, enquanto os demais algoritmos seguiram as implementações disponíveis no OpenCV.

Quanto aos parâmetros utilizados na aplicação dos algoritmos, a Tabela 2 exibe os valores padrão dos rastreadores KCF e CSRT. Esses valores foram retirados diretamente da instância de suas implementações, conforme especificado na documentação oficial. Eles representam as configurações básicas utilizadas para configurar esses algoritmos.

Tabela 2: Parâmetros principais dos rastreadores KCF e CSRT.

Rastreador	Parâmetro	Valor Padrão
CSRT	admm_iterations	4
	cheb_attenuation	45.0
	filter_lr	0.02
	hog_clip	0.20
	num_hog_channels_used	18
	padding	3.0
	scale_lr	0.025
	scale_step	1.02
	template_size	200.0
	use_hog	True
KCF	compress_feature	True
	interp_factor	0.075
	lambda_	0.0001
	output_sigma_factor	0.0625
	pca_learning_rate	0.15
	resize	True
	sigma	0.20
	split_coeff	True

Fonte: Elaborado pelo autor (2025).

Como apresentado na Tabela 2, os rastreadores KCF e CSRT possuem diversos parâmetros ajustáveis que permitem sua personalização para diferentes cenários de rastreamento visual. No caso do rastreador CSRT, parâmetros como *admm_iterations*, que define o número de iterações do método ADMM, um método de otimização iterativa amplamente utilizado para resolver problemas convexos dividindo-os em subproblemas menores, e *cheb_attenuation*, que controla a atenuação de Chebyshev para suavizar a resposta do filtro, são exemplos de ajustes disponíveis. Outros parâmetros incluem *filter_lr*, que determina a taxa de aprendizado para a atualização do filtro, e *padding*, que adiciona um fator de preenchimento ao redor do objeto rastreado para incluir mais informações de contexto.

Já o rastreador KCF apresenta parâmetros como *compress_feature*, que indica se os recursos serão comprimidos para reduzir a dimensionalidade, e *interp_factor*, que controla a taxa de interpolação para atualizar o modelo. Além disso, parâmetros como *lambda_*, que atua como regularizador para evitar *overfitting*, e *resize*, que define se o quadro será redimensionado para maior eficiência computacional, também são ajustáveis. Esses parâmetros tornam os rastreadores KCF e CSRT altamente configuráveis, permitindo sua adaptação a diferentes condições de rastreamento, como mudanças de escala, iluminação ou oclusões.

Entretanto, os demais rastreadores disponíveis no OpenCV, como MedianFlow, Boosting, MOSSE, MIL e TLD, não possuem a classe de parâmetros ajustáveis documentados, o que limita sua personalização. Por essa razão, todas as implementações realizadas utilizaram as configurações padrão fornecidas pela biblioteca, assegurando uniformidade e equidade na análise.

4.4 AVALIAÇÃO DOS ALGORITMOS

As principais métricas consideradas incluem a precisão, medida pela taxa de Interseção sobre União (*Intersection over Union* - IoU), que avalia a capacidade do algoritmo de localizar corretamente os objetos nas imagens; a robustez, avaliada pela análise visual e pelo Erro Médio de Rastreamento (*Mean Tracking Error* - MTE), que mede a habilidade do algoritmo em lidar com variações nas condições de iluminação, oclusões e mudanças nas características dos objetos; e a eficiência computacional, que se refere ao tempo de processamento necessário para rastrear os objetos em um vídeo, mensurado pela taxa de *Frames por Segundo* (*Frames Per Second* - FPS).

Além das métricas quantitativas, melhor apresentadas no Apêndice E, a análise visual dos resultados foi realizada, com o objetivo de observar o comportamento dos algoritmos em situações reais de jogo. Foram analisados aspectos como a continuidade do rastreamento ao longo do vídeo, a resposta a oclusões temporárias e a capacidade de retomar o rastreamento após eventos rápidos, como saques e bloqueios. Esta análise qualitativa buscou identificar possíveis falhas ou limitações dos algoritmos, que não seriam evidentes apenas por meio de métricas quantitativas.

4.5 EXECUÇÃO DE TESTES

A execução de testes envolveu a aplicação prática dos algoritmos de rastreamento em vídeos de partidas de vôlei, com o objetivo de verificar sua eficácia e desempenho em condições reais. Os testes foram conduzidos em várias etapas, começando com a execução dos algoritmos em clipes de vídeo previamente preparados, a fim de avaliar como cada algoritmo lidava com diferentes cenários, como oclusão, variações de iluminação e movimentos complexos.

Para garantir condições iniciais iguais para todos os rastreadores, cada algoritmo foi configurado para iniciar a análise a partir da mesma *bounding box* (ou *bbox*), que é uma região retangular delimitada no vídeo que indica a posição inicial do objeto a ser rastreado. Diversas baterias de testes foram realizadas, e os resultados de cada uma foram comparados com base nas métricas estabelecidas na etapa de avaliação. Essas métricas incluíram critérios como precisão, robustez e eficiência computacional.

Os resultados obtidos foram analisados para verificar se os algoritmos atendiam aos critérios definidos. Quando necessário, ajustes e otimizações foram realizados com base no desempenho observado, e novas iterações foram conduzidas para alcançar os melhores resultados possíveis. Adicionalmente, a execução de testes incluiu a análise da capacidade dos algoritmos de se adaptarem a eventos rápidos e situações dinâmicas, como mudanças abruptas no jogo. Essa etapa foi essencial para garantir que os algoritmos selecionados fossem confiáveis e eficazes em um ambiente esportivo real, fornecendo dados valiosos para a conclusão da pesquisa.

No Capítulo 5, são apresentados os resultados alcançados a partir da aplicação desses procedimentos metodológicos.

5 RESULTADOS

Neste capítulo, são apresentados os resultados obtidos a partir das análises realizadas. Em primeiro plano, são descritos os dados e as configurações dos equipamentos utilizados, detalhando as características técnicas e os parâmetros empregados durante os experimentos. Em seguida, são discutidos os resultados com base nos cenários avaliados, destacando as implicações e limitações identificadas ao longo da realização deste estudo.

5.1 EQUIPAMENTOS UTILIZADOS

Os algoritmos foram executados em um sistema com processador AMD Ryzen™ 5 5500U, com frequência de 2.10 GHz, e 8 GB de memória RAM utilizável. O dispositivo conta com uma placa de vídeo integrada Radeon™ Graphics, adequada para processamento gráfico moderado, e o sistema operacional utilizado foi o Windows® 11, de 64 bits, com arquitetura x64. Para o treinamento do modelo YOLO, foi utilizado o ambiente Google Colab™, empregando a versão nano do YOLOv8 (YOLOv8n). O treinamento foi realizado em uma GPU Tesla® T4, com 14.75 GB de memória total, sendo 4.84 GB reservados, 0.43 GB alocados e 9.48 GB livres para o processamento.

5.2 RESULTADOS QUANTITATIVOS

Os detalhes dos vídeos utilizados para testes e aplicação das métricas estão apresentados nas Tabelas 3 e 4. Essas tabelas fornecem informações sobre a taxa de quadros por segundo (FPS), resolução, tamanho do objeto em *pixels* e a cor da bola.

Tabela 3: Descrição dos vídeos de partidas profissionais.

Vídeo	FPS	Resolução (px)	Tamanho do Objeto (px)
1	30	1280 × 720	16 × 16
2	60	1920 × 1080	22 × 22

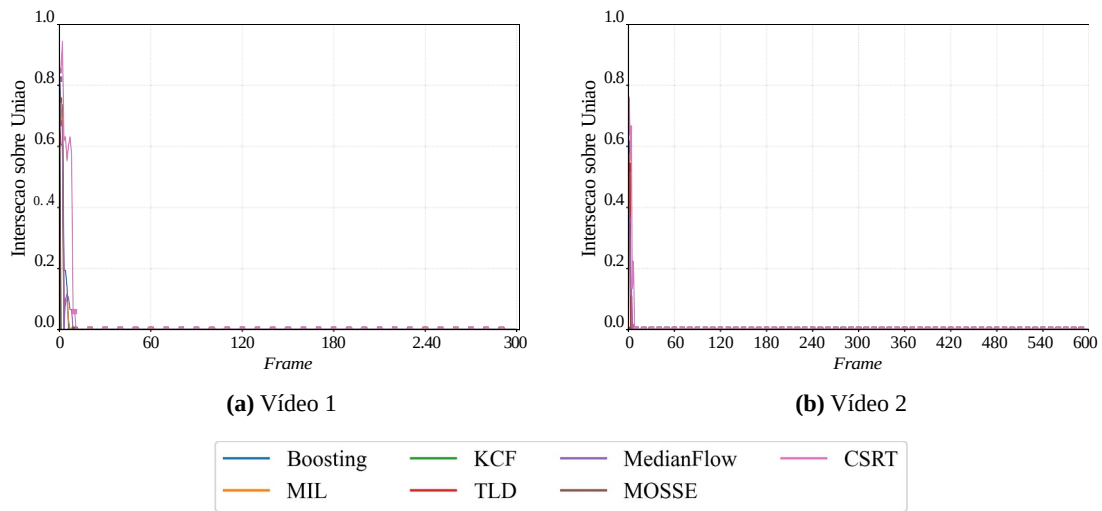
Fonte: Elaborado pelo autor (2025).

Na Tabela 3, são apresentados os vídeos de partidas profissionais. Esses vídeos refletem um cenário com maior complexidade, marcado por um fundo com maior densidade de elementos, maior velocidade no movimento dos objetos e, consequentemente, maior grau de deformação da bola.

5.2.1 Resultados da Interseção sobre União

Os algoritmos demonstraram um desempenho insatisfatório em vídeos profissionais; isso se deve à maior complexidade dos fundos, ocasionada pela presença de plateia, à altavelocidade dos objetos e, consequentemente, às rápidas mudanças de trajetória. Na Figura 18 são apresentados os resultados obtidos com a aplicação da métrica IoU para os vídeos de partidas profissionais.

Figura 18: IoU dos rastreadores ao longo dos *frames* dos vídeos de partidas profissionais.



Fonte: Elaborado pelo autor (2025).

Na Figura 18 é possível observar que o algoritmo CSRT apresentou o melhor desempenho, principalmente no gráfico 18a com 1280×720 px de dimensão, mas ainda insuficiente para garantir a consistência do rastreamento. Por outro lado, os algoritmos MOSSE e KCF obtiveram os piores resultados, com o MOSSE sendo o de pior desempenho geral. Enquanto o CSRT, que iniciou com IoU próxima a 1.0, decai ao longo dos quadros seguintes, os rastreadores com pior desempenho perdem o objeto logo após o início do rastreamento.

Em contraste com os vídeos profissionais, na Tabela 4 são apresentados os detalhes técnicos dos vídeos de partidas amadoras, gravados em ambientes controlados. Esses vídeos possuem características distintas, como fundos menos complexos e movimentos mais previsíveis.

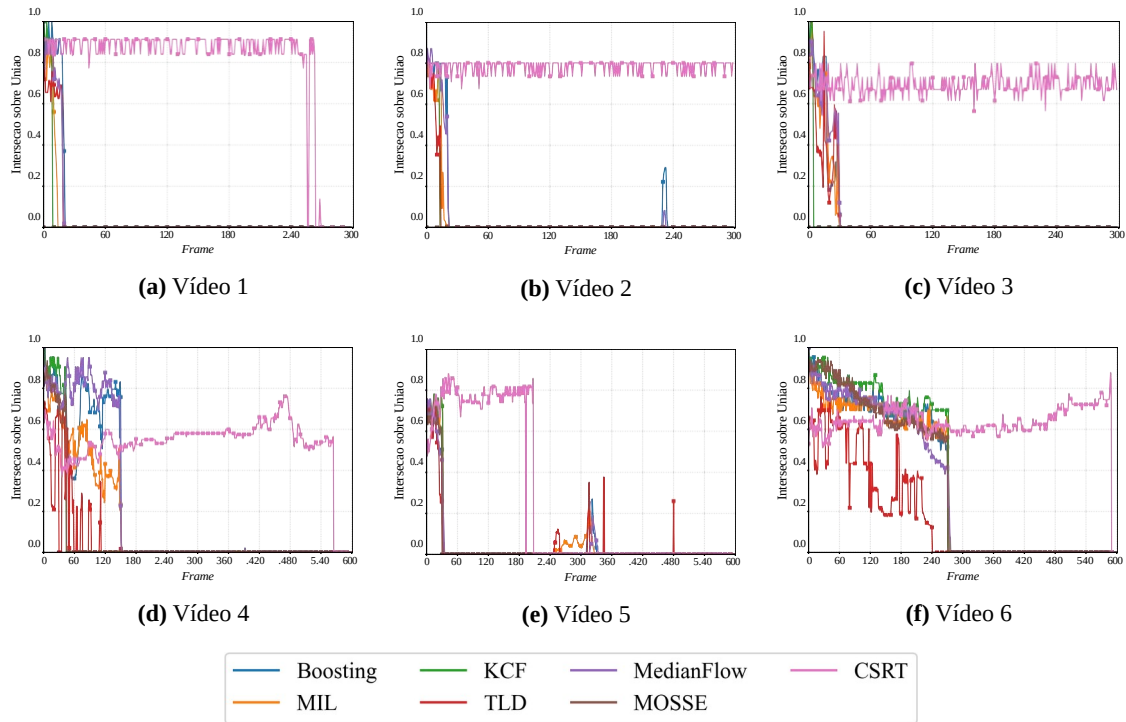
Tabela 4: Descrição dos vídeos de partidas amadoras.

Vídeo	FPS	Resolução (px)	Tamanho do Objeto (px)	Cor da Bola
1	30	1280×720	15×15	Azul
2	30	1280×720	18×18	Laranja
3	30	1280×720	16×16	Verde
4	60	1920×1080	35×35	Azul
5	60	1920×1080	34×34	Laranja
6	60	1920×1080	30×30	Verde

Fonte: Elaborado pelo autor (2025).

Os resultados obtidos para os vídeos expostos nos dados da Tabela 4 são ilustrados na Figura 19. Nesses vídeos, os algoritmos demonstraram melhor desempenho, rastreando os objetos por mais quadros. O algoritmo CSRT, mais uma vez, apresentou o melhor desempenho, seguido pelos algoritmos Boosting e MedianFlow, que, embora inferiores ao CSRT, obtiveram resultados melhores que os demais. Por fim, MOSSE, TLD e KCF apresentaram os piores desempenhos.

Figura 19: IoU dos rastreadores ao longo dos frames dos vídeos de ambiente controlado.



Fonte: Elaborado pelo autor (2025).

Ao analisar os gráficos apresentados na Figura 19, observa-se que o algoritmo CSRT apresenta um desempenho significativamente superior em comparação aos demais, mantendo uma IoU consistente na maioria dos vídeos. Seu desempenho mais preciso, aproximadamente 90%, é apresentado no gráfico da Subfigura 19a, enquanto o desempenho com menor variação, ou seja, com menos erros, ocorre no Vídeo 2 (Subfigura 19b), com média de 0.8 de IoU. Em contrapartida, o pior resultado é observado no gráfico da Subfigura 19e, onde ocorre uma perda precoce do objeto a ser rastreado.

Contudo, mesmo em condições controladas, os algoritmos MOSSE e KCF mantiveram um desempenho insatisfatório, evidenciando limitações em sua aplicação prática, sendo o MOSSE incapaz de manter o rastreamento mesmo por um número pequeno de quadros. O KCF, no entanto, apresentou um desempenho levemente superior, principalmente em vídeos de melhor qualidade. Outro desempenho insatisfatório é observado no TLD, que apresenta taxas baixas de IoU e inconsistência no rastreamento.

5.2.2 Resultados de Quadros por Segundo (Frames Per Second - FPS)

Os resultados em termos de FPS indicam que os algoritmos de rastreamento mais precisos tendem a ser mais lentos na execução de suas tarefas. O CSRT, por exemplo, que apresenta a maior taxa de IoU na maioria dos cenários, alcançou uma média de 24 FPS em vídeos de 30 FPS com resolução de 1280×720 px e apenas 18 FPS em vídeos de 60 FPS com resolução de 1920×1080 px. Evidencia-se que, embora seja mais preciso, o CSRT sacrifica parte de sua velocidade para garantir um rastreamento mais robusto.

Para além disso, é possível observar que a natureza da partida, seja profissional ou gravada em ambiente controlado, não influencia significativamente na taxa de FPS, já que o desempenho está mais relacionado à resolução e à taxa de quadros do vídeo.

Por outro lado, o rastreador MOSSE, que obteve o pior desempenho em termos de IoU, foi o que apresentou os melhores resultados em relação à velocidade. Esse comportamento pode ser atribuído à sua simplicidade e à ausência de um rastreamento adequado do objeto, o que reduz significativamente o tempo de processamento. Em vídeos de 30 FPS, o MOSSE alcançou taxas de processamento de 90 FPS, enquanto, em vídeos de 60 FPS, obteve médias próximas ao limite máximo de FPS do vídeo, variando entre 50 e 60 FPS.

Os rastreadores KCF e MedianFlow apresentaram desempenhos intermediários, com taxas de FPS superiores às do CSRT (70 FPS e 50 FPS, respectivamente), mas inferiores às do MOSSE.

Esses algoritmos conseguem equilibrar de forma mais eficiente a relação entre precisão e velocidade, sendo adequados para cenários em que o rastreamento não exige extrema precisão, mas ainda assim demanda um desempenho confiável.

O rastreador MIL, embora mais lento que o MOSSE e o KCF, apresentou uma taxa de FPS consistente em diferentes resoluções e taxas de quadros, o que o torna uma opção viável para aplicações que demandam estabilidade no desempenho. Já o TLD apresentou uma velocidade inferior, especialmente em vídeos de maior resolução (1920×1080 px), o que pode limitar sua aplicabilidade em cenários que exigem alta taxa de atualização.

Adiante, o Boosting apresentou médias de FPS que indicam uma performance intermediária, sendo mais rápido que rastreadores como CSRT e TLD, mas menos eficiente que opções como KCF e MOSSE. Os resultados detalhados estão apresentados na Tabela 5.

Tabela 5: Médias de FPS por rastreador e resolução.

Rastreador	30 FPS (1280 × 720)	60 FPS (1920 × 1080)
Boosting	35	25
CSRT	24	18
KCF	70	45
MedianFlow	50	35
MIL	40	30
MOSSE	90	55
TLD	25	20

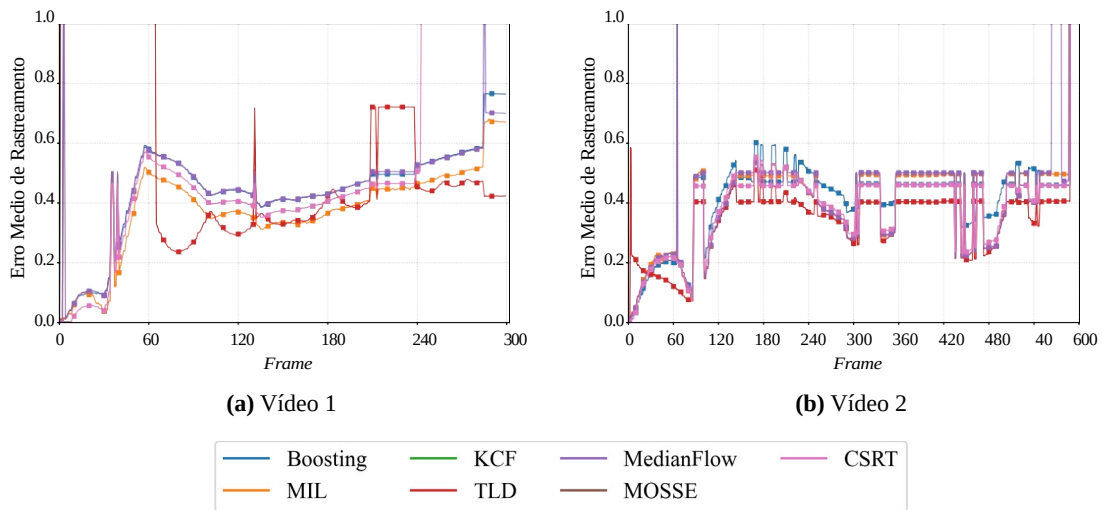
Fonte: Elaborado pelo autor (2025).

Os dados apresentados na Tabela 5 evidenciam a relação inversa entre precisão e velocidade nos algoritmos de rastreamento. Enquanto o CSRT prioriza a robustez do rastreamento, o MOSSE se destaca pela alta velocidade. Essa troca entre precisão e velocidade reflete diretamente nas aplicações práticas, onde a escolha do algoritmo deve considerar as demandas específicas do cenário. Em situações que exigem alta confiabilidade no rastreamento, como análises detalhadas de movimentos em vídeos, algoritmos mais robustos como o CSRT são preferíveis, mesmo com a penalização na velocidade.

5.2.3 Resultados do Erro Médio de Rastreamento

Em termos de erro, os resultados para os vídeos de partidas profissionais são apresentados nos gráficos da Figura 20, onde é possível identificar quais algoritmos falharam com maior frequência ao longo dos frames.

Figura 20: Erro Médio de Rastreamento em vídeos de partidas profissionais.

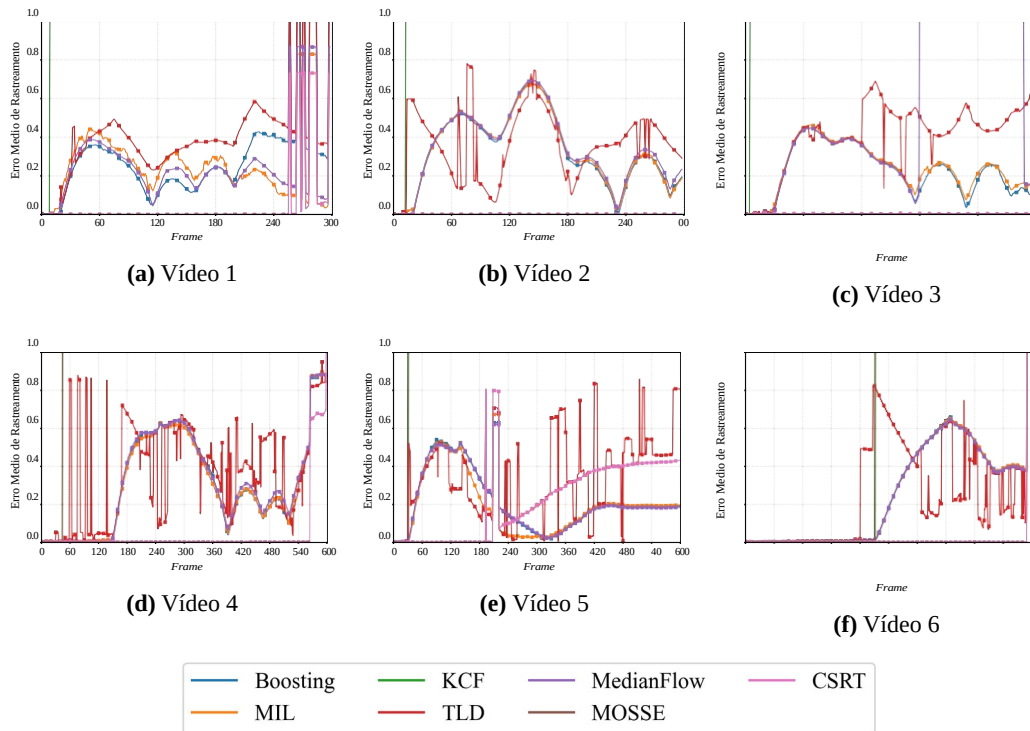


Fonte: Elaborado pelo autor (2025).

Nos vídeos de partidas profissionais, conforme os resultados expressos nos gráficos da Figura 20, o CSRT apresenta um dos menores erros médios, com valores próximos a 0.5, destacando-se como um dos algoritmos mais precisos e consistentes. No entanto, em cenários mais complexos, principalmente em exemplos como o Vídeo 2 (Subfigura 20b), o desempenho de outros algoritmos, como o TLD, é comprometido devido à maior variabilidade e aos picos de erro.

O KCF, por sua vez, apresenta dificuldades em manter o rastreamento, resultando em erros elevados e inconsistentes, uma taxa média de aproximadamente 100% de erro. Por fim, o MIL, apesar de ser robusto em alguns cenários, também demonstra maior sensibilidade a mudanças nos quadros, o que resulta em picos de erro. Na Figura 21, são apresentados os gráficos com os resultados dos vídeos de ambiente controlado.

Figura 21: Erro Médio de Rastreamento em vídeos de partidas com ambiente controlado.



Fonte: Elaborado pelo autor (2025).

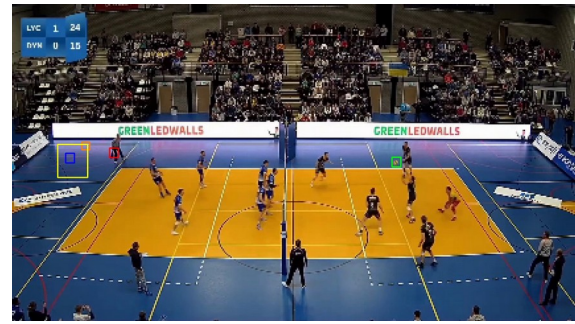
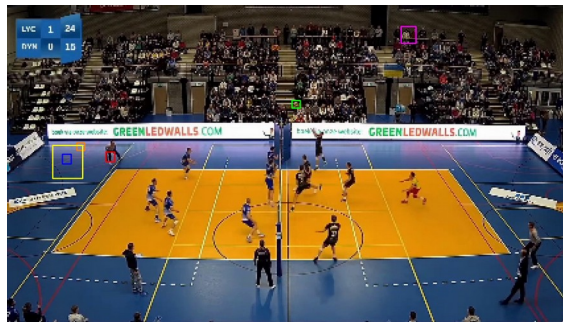
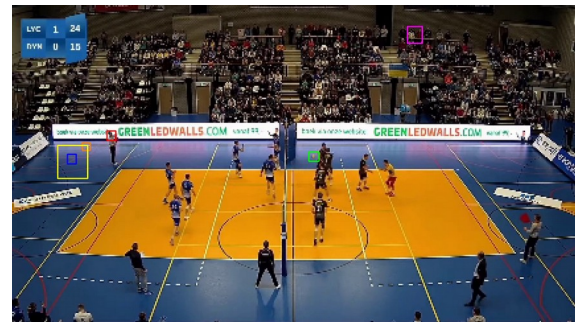
Nos vídeos do cenário controlado, conforme ilustrado na Figura 21, o CSRT novamente se destaca, especialmente em vídeos como o Vídeo 1 (Subfigura 21a) e o Vídeo 2 (Subfigura 21b) onde sua média de erro é quase nula. O Boosting apresenta um desempenho mediano em alguns vídeos, como o Vídeo 3 (Subfigura 21c), mas demonstra uma tendência de aumento no erro médio em quadros posteriores, o que limita sua eficácia em rastreamentos de longo prazo.

Por outro lado, o TLD apresenta maior variabilidade e picos de erro em vídeos como os dos gráficos da Subfigura 21d e Subfigura 21f, comprometendo sua confiabilidade. O KCF, assim como nos vídeos profissionais, demonstra pouca expressividade. Esses resultados destacam a superioridade do CSRT como o algoritmo mais confiável e preciso na maioria dos cenários. Os demais algoritmos apresentam desempenho inferior, sendo mais adequados para cenários específicos ou menos exigentes.

5.3 ANÁLISE VISUAL

A Análise Visual foi realizada em todos os vídeos apresentados nas Tabelas 3 e 4. No entanto, com o objetivo de garantir maior clareza e objetividade na apresentação dos resultados, optou-se por exibir e discutir os resultados da análise visual de apenas dois vídeos: o Vídeo 1 do cenário profissional e o Vídeo 2 do cenário controlado. A seleção desses vídeos foi baseada nos gráficos de resultados quantitativos.

Os resultados obtidos em um cenário profissional, ilustrados na Figura 22, destacam a complexidade adicional introduzida por elementos como iluminação variável e movimentação mais rápida dos objetos, o que impacta diretamente o desempenho dos algoritmos avaliados.

Figura 22: Análise visual do vídeo de partida profissional.**(a)** Frame 1**(b)** Frame 60**(c)** Frame 120**(d)** Frame 180

— ground truth	— CSRT	— MOSSE	— MIL
— Boosting	— MedianFlow	— KCF	— TLD

Fonte: Adaptado de Volleyball Playoffs (2024).

Os resultados apresentados na Figura 22 evidenciam os desafios enfrentados pelos algoritmos de rastreamento em uma partida de vôlei profissional. A movimentação rápida dos jogadores e a complexidade do cenário, com múltiplos elementos em movimento, impactam diretamente a precisão e a robustez dos métodos avaliados.

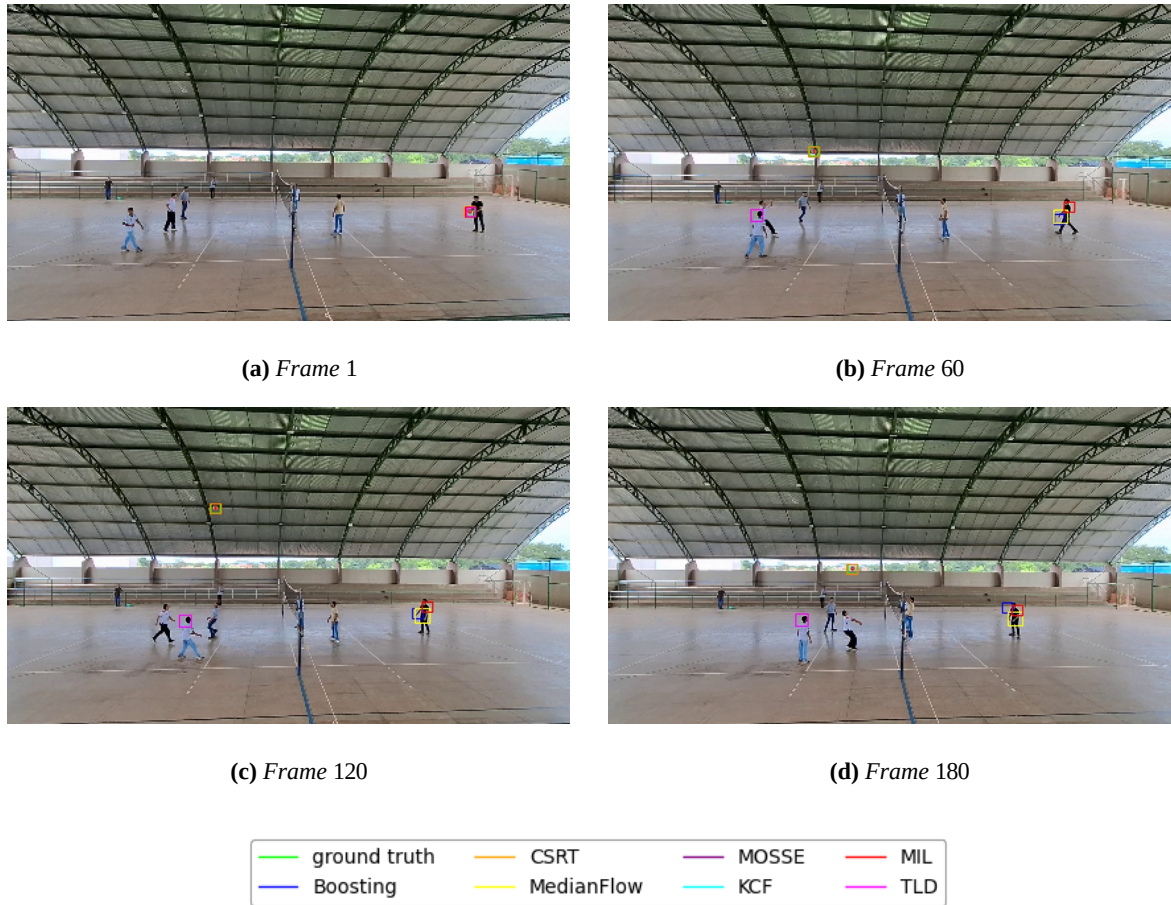
Mesmo o CSRT, que apresentou desempenho superior em cenários mais simples, mostrou limitações significativas ao lidar com as condições desafiadoras deste ambiente. No *Frame 1*, ilustrado na Subfigura 22a, todos os rastreadores detectam o objeto-alvo. No entanto, em um intervalo de apenas 60 *frames*, todos perdem o rastreamento, como é possível perceber nas Subfiguras 22b, 22c e 22d.

Os rastreadores Boosting e MedianFlow mantiveram suas caixas delimitadoras na posição inicial, assim como o CSRT, que, devido à posição da caixa, ainda conseguiu seguir o objeto por alguns quadros. Por outro lado, os rastreadores MIL e TLD, na tentativa de recuperar o objeto, acabaram se fixando em outros elementos do cenário, como indicado nas Subfiguras 22c e 22d. Já os rastreadores MOSSE e KCF desapareceram por completo.

Já os resultados da análise visual em ambiente controlado (Vídeo 2), apresentados na Figura 23, ilustram o desempenho dos algoritmos de OT em um cenário mais simplificado.

Através dessas amostras, é possível observar como cada algoritmo se comporta ao rastrear o objeto de interesse.

Figura 23: Análise visual do vídeo de cenário controlado.



Fonte: Elaborado pelo autor (2025).

Na Figura 23, é possível observar que os algoritmos apresentam variações significativas em termos de precisão e estabilidade no rastreamento do objeto. O algoritmo CSRT, por exemplo, demonstra maior aderência à trajetória real do objeto, enquanto outros, como o MedianFlow e o TLD, apresentam desvios mais acentuados.

No *Frame 1*, ilustrado na Subfigura 23a, todos os rastreadores apresentam suas respectivas caixas delimitadoras corretamente posicionadas ao redor do objeto-alvo. Entretanto, ao longo dos *frames*, apenas o CSRT consegue manter a precisão no rastreamento, acompanhando a bola por, no mínimo, 180 *frames*, como indicado na Subfigura 23d. Os demais rastreadores obtiveram resultados semelhantes aos apresentados na análise da Figura 22, perdendo o objeto rapidamente.

Os fatores limitantes evidenciados nos resultados da análise visual destacam a necessidade de algoritmos mais adaptativos e resilientes, capazes de lidar de forma eficaz com situações dinâmicas e imprevisíveis, como as observadas nos exemplos apresentados.

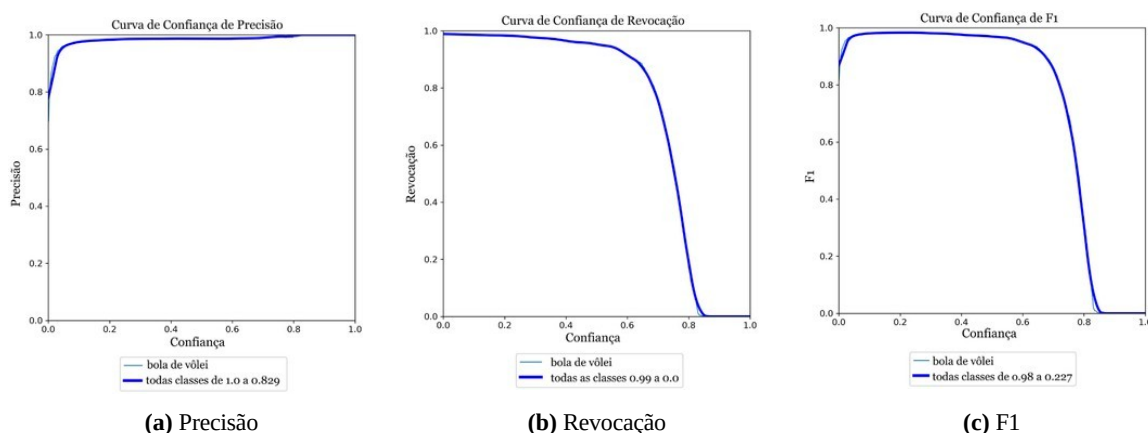
5.4 RESULTADOS DO YOLOV8

O modelo YOLOv8 foi treinado utilizando uma parcela do *dataset* criado, com um total de 1.600 amostras, ao longo de 300 épocas. Embora o *dataset* completo contenha um número significativamente maior de amostras, essa seleção limitada foi utilizada com o objetivo de realizar testes iniciais e experimentos preliminares.

Durante o treinamento, métricas como Precisão (*Precision*), Revocação (*Recall*) e F1 (*F1-score*) foram monitoradas para avaliar o desempenho do modelo. É importante destacar que a limitação no número de amostras pode ter influenciado os resultados, especialmente em termos de generalização para cenários mais complexos.

As curvas de Precisão, Revocação e F1, ilustradas na Figura 24, permitem avaliar diferentes aspectos do desempenho do modelo. A curva de Precisão mede a proporção de predições corretas entre todas as predições realizadas pelo modelo, sendo essencial para cenários onde falsos positivos podem comprometer os resultados. Já a curva de Revocação avalia a proporção de objetos corretamente detectados em relação ao total de objetos presentes no *dataset*, enquanto a curva F1 evidencia o equilíbrio entre precisão e revocação, combinando ambas as métricas em uma única medida.

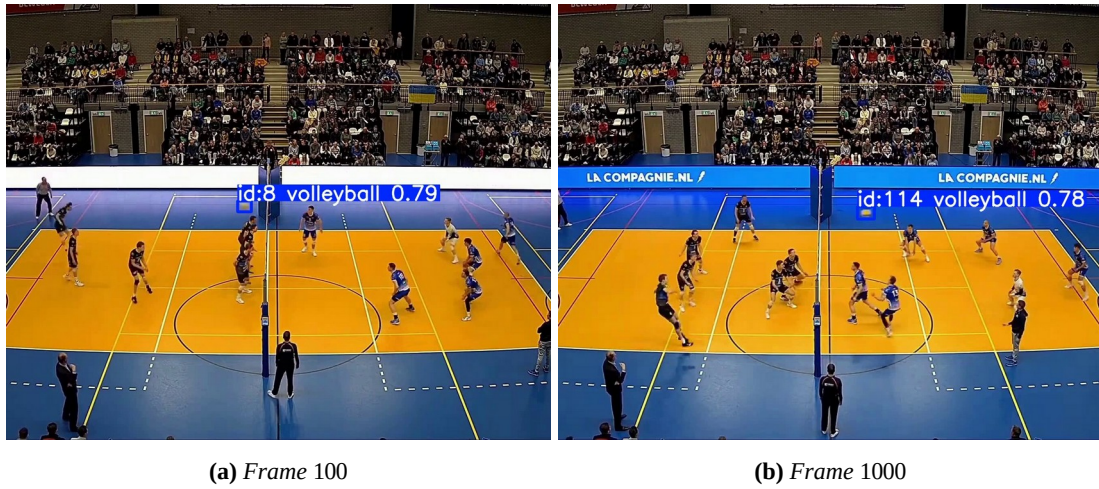
Figura 24: Curvas de Precisão, Revocação e F1.



Fonte: Elaborado pelo autor (2025).

Na curva de Precisão, observa-se que o modelo atinge precisão máxima de 1.00 para a classe "volleyball" em um limiar de confiança de 0.829, demonstrando alta confiabilidade nas detecções realizadas. Já na curva de Revocação, os valores permanecem elevados (próximos de 0.99) para limiares baixos, mas apresentam uma queda acentuada à medida que o limiar aumenta, indicando que o modelo perde capacidade de identificar todos os alvos em limiares mais altos. Por fim, a curva F1 evidencia um valor máximo de 0.98 em um limiar de confiança de 0.227, demonstrando um bom equilíbrio entre precisão e revocação.

Apesar do resultado alcançado, o modelo apresentou limitações significativas ao ser aplicado para VOT. Entre os problemas observados, destaca-se a dificuldade em manter a continuidade do rastreamento, resultando em trajetórias inconsistentes. Além disso, ocorreram trocas frequentes de IDs, como mostrado na Figura 25, indicando falhas na preservação da identidade dos objetos rastreados.

Figura 25: Troca de ID ao longo dos *frames*.

Fonte: Adaptado de Volleybal Playoffs (2024).

Na Figura 25, são ilustradas as trocas de IDs observadas ao longo dos *frames*. No *Frame* 100, o objeto identificado com o ID 8 é rastreado, mas no *Frame* 1000, o mesmo objeto é identificado como ID 114, evidenciando a inconsistência no rastreamento. Esse comportamento pode estar relacionado ao uso de um número limitado de amostras durante o treinamento, o que pode ter impactado a capacidade do modelo de generalizar adequadamente para cenários mais complexos. Em adição, o tempo de execução elevado inviabiliza o uso do modelo em aplicações de tempo real. Esses fatores limitam a aplicabilidade do modelo em cenários que exigem rastreamento contínuo e em tempo real.

5.5 SÍNTESE DOS RESULTADOS

Na Tabela 6 é apresentada a síntese geral dos resultados discutidos ao longo deste capítulo no cenário profissional.

Tabela 6: Síntese dos resultados em cenário profissional.

Rastreador	IoU Médio (%)	FPS Médio	MTE Médio (%)
Boosting	4.17	35.26	53.45
CSRT	5.84	20.13	55.91
KCF	0.24	67.84	99.83
MedianFlow	4.31	41.99	52.05
MIL	5.39	13.82	49.40
MOSSE	0.24	86.53	99.83
TLD	1.37	3.64	52.42

Fonte: Elaborado pelo autor (2025).

No cenário profissional, como é possível observar na Tabela 6, os rastreadores apresentaram valores baixos de IoU Médio, com destaque para o KCF e o MOSSE, que obtiveram o menor valor, com 0,24%. O melhor desempenho em termos de IoU Médio foi alcançado pelo CSRT, com 5,84%, seguido pelo MIL, com 5,39%. Em relação ao FPS Médio, que mede a velocidade de processamento, o MOSSE foi o mais rápido, atingindo 86,53 FPS, seguido pelo KCF, com 67,84 FPS. O pior desempenho nesse aspecto foi do TLD, com apenas 3,64 FPS.

Quanto ao MTE Médio, que mede o erro médio de rastreamento, os valores foram elevados para todos os rastreadores. O KCF e o MOSSE apresentaram o maior erro, com 99,83%. O melhor desempenho em termos de MTE Médio foi do MIL, com 49,40%, seguido pelo MedianFlow, com 52,05%.

Esses resultados refletem o desempenho dos rastreadores em um cenário profissional desafiador, onde a precisão é reduzida, mas a velocidade de processamento pode variar significativamente entre os métodos. O MOSSE se destacou como o mais rápido, enquanto o CSRT obteve a melhor precisão em termos de IoU Médio, e o MIL apresentou o menor erro de rastreamento. Na Tabela 7 são apresentados os resultados de desempenho dos rastreadores no ambiente controlado.

Tabela 7: Síntese dos resultados em ambiente controlado.

Rastreador	IoU Médio (%)	FPS Médio	MTE Médio (%)
Boosting	28.29	30.25	20.77
CSRT	84.92	18.93	5.61
KCF	11.02	57.48	88.81
MedianFlow	26.38	34.10	25.87
MIL	28.48	12.92	20.21
MOSSE	8.98	73.44	90.41
TLD	20.14	5.94	27.50

Fonte: Elaborado pelo autor (2025).

Na Tabela 7, onde são apresentados os resultados de desempenho dos rastreadores em um ambiente controlado, é possível observar que nesse cenário, os rastreadores apresentaram um desempenho superior em comparação ao cenário profissional, com destaque para o CSRT, que alcançou o maior valor de IoU Médio, com 84,92%. O pior desempenho em termos de IoU Médio foi do MOSSE, com 8,98%.

Em relação ao FPS Médio, que mede a velocidade de processamento, o MOSSE foi o mais rápido, atingindo 73,44 FPS, seguido pelo KCF, com 57,48 FPS. O pior desempenho nesse aspecto foi do TLD, com apenas 5,94 FPS. Quanto ao MTE Médio, que mede o erro médio de rastreamento, o CSRT apresentou o menor erro, com 5,61%, indicando maior precisão nesse cenário. O pior desempenho em termos de MTE Médio foi do MOSSE, com 90,41%.

Esses resultados mostram que o ambiente controlado favorece uma maior precisão e estabilidade, permitindo que os rastreadores mantenham o objeto rastreado com mais consistência. O CSRT se destacou como o mais preciso em termos de IoU Médio e MTE Médio, enquanto o MOSSE foi o mais rápido em termos de FPS Médio.

Conforme apresentado nas Tabelas 6 e 7, os rastreadores avaliados demonstraram diferentes níveis de desempenho, com destaque para o CSRT, que obteve os melhores resultados em termos de precisão, e para o MOSSE, que se destacou pela alta velocidade, embora com menor acurácia. Já o YOLOv8, apesar de não estar incluído na tabela, apresentou excelente desempenho na detecção de objetos, com alta precisão e revocação. No entanto, ao ser utilizado para rastreamento, enfrentou limitações como trocas de IDs, além de um tempo de execução elevado, o que restringe sua aplicação em tempo real. Esses resultados reforçam a necessidade de ajustes e otimizações para melhorar a robustez e a eficiência do modelo em tarefas de rastreamento. As discussões sobre os resultados apresentados neste capítulo serão aprofundadas no Capítulo 6, dedicado às Considerações Finais.

6 CONSIDERAÇÕES FINAIS

O presente estudo teve como objetivo principal avaliar e identificar os algoritmos mais adequados para o desenvolvimento de um sistema de rastreamento visual preciso e adaptável para partidas de vôlei, com foco em aprimorar tanto a análise de desempenho quanto a arbitragem no esporte. Para isso, foram analisados diversos algoritmos de rastreamento visual em ambiente controlado e em partidas profissionais, caracterizadas por desafios como iluminação variável, movimentação rápida e frequentes oclusões.

Os resultados obtidos evidenciaram que o algoritmo CSRT apresentou o melhor desempenho geral, destacando-se em termos de precisão (IoU) e estabilidade no rastreamento, especialmente em cenários controlados. No entanto, mesmo o CSRT demonstrou limitações significativas em partidas profissionais, onde a maior complexidade do ambiente impactou negativamente sua robustez e consistência. Outros algoritmos, como Boosting e MedianFlow, apresentaram desempenhos intermediários, enquanto MOSSE e KCF obtiveram os piores resultados, com baixa precisão e maior suscetibilidade a falhas.

A análise revelou uma clara troca entre precisão e eficiência computacional. Algoritmos mais precisos, como o CSRT, tendem a ser mais lentos, enquanto os mais rápidos, como o MOSSE, apresentam desempenho inferior em termos de rastreamento. Esse equilíbrio entre precisão e velocidade é um fator crítico a ser considerado no desenvolvimento de sistemas em tempo real, como os destinados à arbitragem e análise de desempenho.

Adicionalmente, esses resultados demonstram que rastreadores com baixo desempenho em termos de precisão não devem ser avaliados exclusivamente com base na velocidade. Embora o MOSSE seja o mais rápido, sua baixa precisão compromete sua utilidade em aplicações que exigem rastreamento confiável. Por outro lado, o CSRT destaca-se por oferecer a maior precisão entre os algoritmos avaliados, ao mesmo tempo em que mantém uma taxa de FPS moderada, demonstrando que é possível alcançar um equilíbrio entre precisão e velocidade, mesmo com alguma perda em desempenho.

Acrescenta-se que os resultados obtidos com o modelo treinado do YOLO, embora promissores para a detecção, ainda precisam de melhorias para que possa realizar o rastreamento de modo mais confiável. Outro problema identificado é a generalização do modelo, que requer uma maior quantidade de amostras para um treinamento mais eficaz.

Os desafios enfrentados pelos algoritmos em cenários profissionais reforçam a necessidade de soluções mais adaptativas e resilientes, capazes de lidar com situações dinâmicas e imprevisíveis. Fatores como oclusões frequentes, variações de iluminação e a alta velocidade da bola destacam as limitações dos algoritmos clássicos, indicando que abordagens baseadas em DL podem oferecer maior capacidade de generalização e robustez em ambientes complexos.

Por fim, a possibilidade de realizar pré-processamento nas imagens, como normalização de cores, equalização de histograma ou remoção de ruídos, pode ser explorada para verificar melhorias no desempenho dos algoritmos. No entanto, é importante ressaltar que tais etapas de pré-processamento podem aumentar significativamente o tempo de execução, o que pode ser um fator limitante em aplicações que demandam alta velocidade, como o rastreamento em tempo real.

6.1 TRABALHOS FUTUROS

Neste trabalho, foi realizada a avaliação de rastreadores clássicos, que, apesar de apresentarem algumas vantagens em cenários controlados, demonstraram limitações significativas em situações mais complexas, como partidas profissionais de vôlei. Para complementar a análise, foi utilizado o YOLO como um adicional para explorar o potencial de algoritmos mais avançados em cenários dinâmicos. O YOLO demonstrou maior capacidade de lidar com desafios como oclusões frequentes, variações de iluminação e alta velocidade da bola, destacando-se como uma alternativa promissora para o rastreamento visual em ambientes complexos.

A análise realizada fornece uma base inicial para o aprimoramento de sistemas de rastreamento aplicados ao vôlei, contribuindo para a compreensão dos desafios específicos do esporte. Ademais, a criação de um conjunto de imagens para treinamento, teste e validação representa um passo importante para o desenvolvimento de soluções mais robustas e adaptáveis.

Com base nos objetivos definidos, algumas direções futuras podem ser destacadas:

- **Integração de técnicas baseadas em DL:** CNNs e modelos de rastreamento baseados em DL, como Siamese Networks e Transformers, podem ser explorados para superar as limitações dos algoritmos clássicos.
- **Desenvolvimento de sistemas híbridos:** a combinação de algoritmos clássicos com técnicas de DL pode equilibrar precisão e eficiência computacional.
- **Aprimoramento do conjunto de dados:** a expansão do conjunto de imagens com mais exemplos de situações desafiadoras, como oclusões e variações de iluminação, pode melhorar o treinamento e a validação de novos modelos.
- **Avaliação em cenários reais:** testes em partidas profissionais com sistemas integrados de rastreamento e análise podem validar a aplicabilidade das soluções propostas e identificar novos desafios a serem enfrentados.
- **Exploração de tecnologias complementares:** o uso de chips para monitorar dados específicos, como velocidade e aceleração de objetos, pode ser integrado a sistemas de Visão Computacional para fornecer informações pontuais e contextuais. Essa abordagem híbrida pode unir a precisão dos dados isolados fornecidos pelos chips com a análise global e dinâmica proporcionada pela CV, ampliando as possibilidades de rastreamento e análise no esporte.

REFERÊNCIAS

- AHARON, N.; ORFAIG, R.; BOBROVSKY, B.-Z. Bot-sort: Robust associations multi-pedestrian tracking. **arXiv preprint arXiv:2206.14651**, 2022.
- ALI, A.; JALIL, A.; NIU, J.; ZHAO, X.; RATHORE, S.; AHMED, J.; IFTIKHAR, M. A. Visual object tracking—classical and contemporary approaches. **Frontiers of Computer Science**, Springer, v. 10, p. 167–188, 2016.
- BABENKO, B.; YANG, M.-H.; BELONGIE, S. Robust object tracking with online multiple instance learning. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 33, n. 8, p. 1619–1632, 2010.
- BASHAR, M.; ISLAM, S.; HUSSAIN, K. K.; HASAN, M. B.; RAHMAN, A.; KABIR, M. H. Multiple object tracking in recent times: A literature review. **arXiv preprint arXiv:2209.04796**, 2022.
- BATISTA, I. K. T. Avaliação da qualidade de experiência em transmissão de vídeo com codec h. 264 utilizando métricas objetivas de referência completa e sem referência. Brasil, 2024.
- BENHAJYOUSSEF, A.; SAIDANI, A. Recent advances on image edge detection. In: CUEVAS, F.; MAZZEO, P. L.; BRUNO, A. (Ed.). **Digital Image Processing**. Rijeka: IntechOpen, 2024. cap. 3. Disponível em: <<https://doi.org/10.5772/intechopen.1003763>>.
- BEWLEY, A.; GE, Z.; OTT, L.; RAMOS, F.; UPCROFT, B. Simple online and realtime tracking. In: IEEE. **2016 IEEE international conference on image processing (ICIP)**. Phoenix, Arizona, USA, 2016. p. 3464–3468.
- BOLME, D. S.; BEVERIDGE, J. R.; DRAPER, B. A.; LUI, Y. M. Visual object tracking using adaptive correlation filters. **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, p. 2544–2550, 2010. Disponível em: <<https://api.semanticscholar.org/CorpusID:2451356>>.
- BRADSKI, G. The OpenCV Library. **Dr. Dobb's Journal of Software Tools**, 2000.
- BRDJANIN, A.; DARDAGAN, N.; DZIGAL, D.; AKAGIC, A. Single object trackers in opencv: A benchmark. In: IEEE. **2020 International Conference on Innovations in Intelligent Systems and Applications (INISTA)**. Novi Sad, Serbia, 2020. p. 1–6.
- CARDOSO, V. A. Identificação de placas veiculares utilizando processamento de imagens. Edifes, 2023.
- CHEN, F.; WANG, X.; ZHAO, Y.; LV, S.; NIU, X. Visual object tracking: A survey. **Computer Vision and Image Understanding**, Elsevier, v. 222, p. 103508, 2022.
- Confederação Brasileira de Voleibol. **Regras Oficiais de Voleibol 2021-2024**. Rio de Janeiro, Brasil: Confederação Brasileira de Voleibol, 2021. Acesso em: 14 dez. 2024. Disponível em: <<https://www.cbv.com.br>>.
- DANTAS, C. A. A. **Uma proposta para o ensino do movimento retilíneo uniforme utilizando fotoanálise e videoanálise**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2022.

- DIWAN, T.; ANIRUDH, G.; TEMBHURNE, J. V. Object detection using yolo: Challenges, architectural successors, datasets and applications. **multimedia Tools and Applications**, Springer, v. 82, n. 6, p. 9243–9275, 2023.
- FREUND, Y.; SCHAPIRE, R.; ABE, N. A short introduction to boosting. **Journal-Japanese Society For Artificial Intelligence**, JAPANESE SOC ARTIFICIAL INTELL, v. 14, n. 771-780, p. 1612, 1999.
- FREUND, Y.; SCHAPIRE, R. E. *et al.* Experiments with a new boosting algorithm. In: CITESEER. **icml**. [S.l.], 1996. v. 96, p. 148–156.
- FU, Y.; QIN, P.; ZHANG, J.; LU, Z. Joint ai inference and target tracking at network edge: A hybrid offline-online design for uav-enabled network. **IEEE Transactions on Wireless Communications**, IEEE, 2024.
- Fédération Internationale de Volleyball. **History of Volleyball**. 2025. Acessado em: 17 fev. 2025. Disponível em: <<https://www.fivb.com/volleyball/the-game/history/>>.
- GARCIA, R. M.; MEIRELES, C. H. A. de; PEREIRA, E. G. B. Evolução e adaptação histórica do voleibol. **Lecturas: Educación Física Y Deportes**, v. 26, n. 281, p. 183–203, 2021.
- GHOSH, I.; RAMAMURTHY, S. R.; CHAKMA, A.; ROY, N. Sports analytics review: Artificial intelligence applications, emerging technologies, and algorithmic perspective. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, Wiley Online Library, v. 13, n. 5, p. e1496, 2023.
- GIRSHICK, R.; DONAHUE, J.; DARRELL, T.; MALIK, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Columbus, OH, USA: [s.n.], 2014. p. 580–587.
- GONZALEZ, R.; WOODS, R. **Digital Image Processing, Global Edition**. Pearson Education, 2018. ISBN 9781292223070. Disponível em: <<https://books.google.com.br/books?id=p74oEAAAQBAJ>>.
- GRABNER, H.; GRABNER, M.; BISCHOF, H. Real-time tracking via on-line boosting. In: CITESEER. **British Machine Vision Conference (BMVC)**. Edinburgh, United Kingdom, 2006. v. 1, n. 5, p. 6.
- GUDAUSKAS, J.; MATUSEVICIUS, Ž. Multiple object tracking for video-based sports analysis. In: CEUR-WS. **CEUR workshop proceedings: IVUS 2021: Information society and university studies 2021: Proceedings of the 26th international conference on information society and university studies (IVUS 2021)**. Kaunas, Lithuania, 2021. v. 2915, p. 1–10.
- HAN, X.; WANG, Q.; WANG, Y. Ball tracking based on multiscale feature enhancement and cooperative trajectory matching. **Applied Sciences**, MDPI, v. 14, n. 4, p. 1376, 2024.
- HARTMANN, C.; SILVA, A.; OLIVEIRA, S. L.; DA, J.; FILHO, S.; JOSE, G.; SANTOS. História do voleibol. 04 2022.
- HENRIQUES, J. F.; CASEIRO, R.; MARTINS, P.; BATISTA, J. Exploiting the circulant structure of tracking-by-detection with kernels. In: SPRINGER. **Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12**. Florence, Italy, 2012. p. 702–715.

- HENRIQUES, J. F.; CASEIRO, R.; MARTINS, P.; BATISTA, J. High-speed tracking with kernelized correlation filters. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 37, n. 3, p. 583–596, 2014.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.
- HUANG, G. An effective volleyball trajectory estimation and analysis method with embedded graph convolution. **International Journal of Distributed Systems and Technologies (IJDST)**, IGI Global, v. 14, n. 2, p. 1–13, 2023.
- JAVED, S.; DANELLJAN, M.; KHAN, F. S.; KHAN, M. H.; FELSBERG, M.; MATAS, J. Visual object tracking with discriminative filters and siamese networks: a survey and outlook. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 45, n. 5, p. 6552–6574, 2022.
- JOCHER, G.; CHAURASIA, A.; QIU, J. **Ultralytics YOLOv8**. 2023. Disponível em: <<https://github.com/ultralytics/ultralytics>>.
- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Forward-backward error: Automatic detection of tracking failures. In: **2010 20th International Conference on Pattern Recognition**. Istanbul, Turkey: IEEE, 2010. p. 2756–2759.
- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-learning-detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 34, n. 7, p. 1409–1422, 2011.
- 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>>.
- KAUR, R.; SINGH, S. A comprehensive review of object detection with deep learning. **Digital Signal Processing**, Elsevier, v. 132, p. 103812, 2023.
- KUMAR, A.; VOHRA, R.; JAIN, R.; LI, M.; GAN, C.; JAIN, D. K. Correlation filter based single object tracking: A review. **Information Fusion**, v. 112, p. 102562, 2024. ISSN 1566-2535. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1566253524003403>>.
- LIU, J. S.; CHEN, R. Sequential monte carlo methods for dynamic systems. **Journal of the American statistical association**, Taylor & Francis, v. 93, n. 443, p. 1032–1044, 1998.
- LIU, S.; LIU, D.; SRIVASTAVA, G.; POŁAP, D.; WOZ´ NIAK, M. **Overview of correlation filter based algorithms in object tracking**. **Complex Intell Syst**. 2020.
- LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. Ssd: Single shot multibox detector. In: SPRINGER. **Computer Vision–ECCV 2016: Proceedings of the 14th European Conference on Computer Vision, Part I**. Amsterdam, The Netherlands: Springer International Publishing, 2016. p. 21–37.
- LIU, X.; CAESAR, H. Offline tracking with object permanence. **arXiv.org**, abs/2310.01288, 2023.
- LUKEZIC, A.; VOJIR, T.; ZAJC, L. C.; MATAS, J.; KRISTAN, M. Discriminative correlation filter with channel and spatial reliability. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. Honolulu, Hawaii, USA: IEEE, 2017. p. 6309–6318.

- LUO, W.; XING, J.; MILAN, A.; ZHANG, X.; LIU, W.; KIM, T.-K. Multiple object tracking: A literature review. **Artificial intelligence**, Elsevier, v. 293, p. 103448, 2021.
- MARENGONI, M.; STRINGHINI, D. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**; Vol. 16, No 1 (2009); 125-160, v. 16, 03 2010.
- MEZZAROBBA, C.; PIRES, G. d. L. Breve panorama histórico do voleibol: do seu surgimento à espetacularização esportiva. **Atividade Física, Lazer & Qualidade de Vida: revista de educação física**, Curso de Educação Física da Universidade do Estado do Amazonas, 2011.
- NAIK, B. T.; HASHMI, M. F.; BOKDE, N. D. A comprehensive review of computer vision in sports: Open issues, future trends and research directions. **Applied Sciences**, MDPI, v. 12, n. 9, p. 4429, 2022.
- NGUYEN, D.; NGUYEN, C.; DUONG-BA, T.; NGUYEN, H.; NGUYEN, A.; TRAN, T. Joint network coding and machine learning for error-prone wireless broadcast. In: IEEE. **2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)**. Piscataway, NJ, USA: IEEE, 2017. p. 1–7.
- NOGUEIRA, G. R. P. Concepções metodológicas para o ensino-aprendizagem do voleibol. Brasil, 2024.
- NUNES, V. d. S. Tópicos em visão computacional: uma revisão sistemática com aplicações em economia 4.0. Serra, 2023.
- PEREIRA, L. V. L. Avaliação de modelos de detecção de objetos na identificação de doenças pulmonares e cardíacas em imagens de raio-x torácicos. 2024.
- REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 39, n. 6, p. 1137–1149, 2016.
- ROMÃO, B. Redes neurais convolucionais para a detecção de objetos. Pontifícia Universidade Católica de Campinas (PUC-Campinas), 2023.
- SARIGÜL, M. A survey on digital video stabilization. **Multimedia Tools and Applications**, Springer, v. 82, n. 26, p. 40181–40207, 2023.
- SILVA, R. B. d. Desenvolvimento de tecnologia de informação e comunicação (tic) para análise instantânea de desempenho de voleibol. Universidade Estadual Paulista (Unesp), 2021.
- Volleybal Playoffs. **Canal do YouTube: VolleybalPlayoffs**. 2024. <<https://www.youtube.com/@volleybalplayoffs>>. Acesso em: 9 nov. 2024.
- Wikipédia. **História do voleibol**. 2025. Acessado em: 17 fev. 2025. Disponível em: <https://pt.wikipedia.org/wiki/Hist%C3%B3ria_do_voleibol>.
- WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and realtime tracking with a deep association metric. In: IEEE. **2017 IEEE International Conference on Image Processing (ICIP)**. Beijing, China: IEEE, 2017. p. 3645–3649.
- WU, Y.; LIM, J.; YANG, M.-H. Online object tracking: A benchmark. In: IEEE. **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. Portland, Oregon, USA: IEEE, 2013. p. 2411–2418.

YARDIMCI, O.; TEKEREK, A. S., imşek. Performance comparison of online and offline tracking algorithms. In: **Signal Processing, Sensor/Information Fusion, and Target Recognition**

XXXI. Orlando, Florida, USA: SPIE, 2022. v. 12096, p. 1209604–1209604.

ZAIDI, S. S. A.; ANSARI, M. S.; ASLAM, A.; KANWAL, N.; ASGHAR, M.; LEE, B. A survey of modern deep learning based object detection models. **Digital**

Signal Processing, v. 126, p. 103514, 2022. ISSN 1051-2004. Disponível em:

<<https://www.sciencedirect.com/science/article/pii/S1051200422001312>>.

ZHANG, Y.; SUN, P.; JIANG, Y.; YU, D.; WENG, F.; YUAN, Z.; LUO, P.; LIU, W.; WANG, X.

Bytetrack: Multi-object tracking by associating every detection box. In: SPRINGER. **European Conference on Computer Vision (ECCV)**. Tel Aviv, Israel, 2022. p. 1–21.

Su Zhihui & Chen Sijing. **Offline method-based online tracking method and apparatus, computer device, and storage medium**. 2021.

ZOU, Z.; CHEN, K.; SHI, Z.; GUO, Y.; YE, J. Object detection in 20 years: A survey.

Proceedings of the IEEE, v. 111, n. 3, p. 257–276, 2023.

APÊNDICE A – IMAGENS E VÍDEOS DIGITAIS

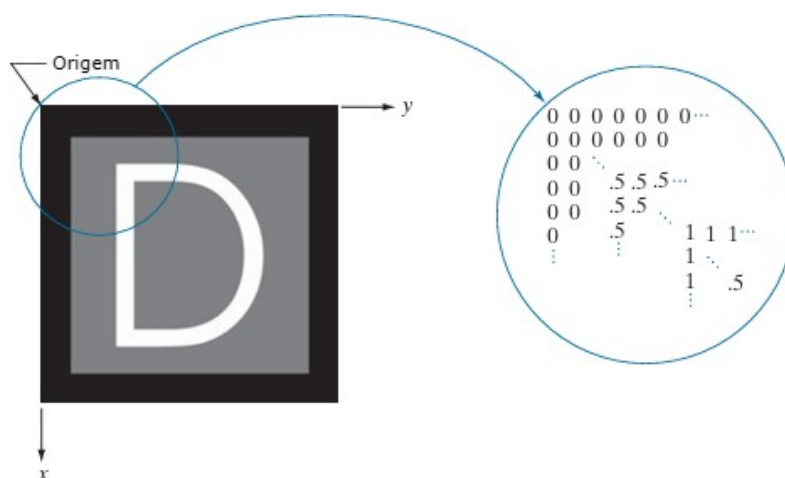
A.1 IMAGENS E VÍDEOS DIGITAIS

A era digital trouxe consigo uma transformação significativa na forma como capturamos, armazenamos e interpretamos dados visuais. As imagens e vídeos digitais desempenham um papel central em diversas áreas do conhecimento e da indústria, desde a comunicação e o entretenimento até a medicina e a segurança. Este apêndice explora os fundamentos das imagens e vídeos digitais, destacando suas características, formas de criação, processamento e aplicações.

A.1.1 Imagens Digitais

Uma imagem digital é uma representação bidimensional de dados visuais obtida através da captura, armazenamento ou geração de informações visuais em formato eletrônico. Cada imagem digital é composta por uma matriz de *pixels* (*px*), como exibido na Figura 26,

Figura 26: Representação de imagem digital.



Fonte: Adaptado de Gonzalez & Woods (2018, p. 67).

Na Figura 26 é apresentado o *pixel*, que corresponde à menor unidade de uma imagem digital. Cada *pixel* representa um ponto específico da imagem e armazena informações sobre cor e intensidade luminosa, sendo essencial para a composição visual e a análise de imagens (CARDOSO, 2023). Essas imagens digitais podem ser criadas a partir de diversas fontes, como câmeras digitais, escâneres, satélites e até mesmo por meio de softwares gráficos.

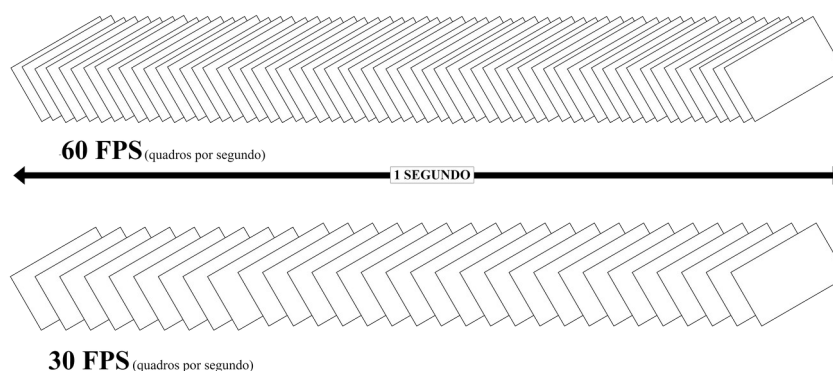
Segundo Gonzalez & Woods (2018), as imagens digitais são fundamentais para a análise e interpretação de dados visuais em diversas aplicações, desde o reconhecimento de padrões até a visualização científica. A criação e o processamento de imagens digitais envolvem diversas técnicas e algoritmos que permitem a manipulação das propriedades visuais para melhorar a

qualidade, extrair informações ou realizar análises detalhadas. A evolução das tecnologias de imagem digital tem revolucionado áreas como a fotografia, a medicina, a astronomia e a vigilância, permitindo uma maior precisão e detalhes na captura de imagens.

A.1.2 Vídeos Digitais

Um vídeo é composto por uma sequência de imagens estáticas exibidas em rápida sucessão, criando a ilusão de movimento contínuo. Ele é gerado a partir da captura sequencial das imagens refletidas na lente de uma câmera durante seu deslocamento no mundo real (SARIGÜL, 2023). Cada imagem individual em um vídeo é chamada de *frame*. A taxa de quadros (*Frames Per Second* - FPS) define quantos *frames* são exibidos por segundo, como ilustrado na Figura 27, influenciando diretamente a fluidez da reprodução do vídeo.

Figura 27: Representação de FPS.



Fonte: Elaborado pelo autor (2024).

Como ilustrado na Figura 27, a maioria dos vídeos modernos é gravada a taxas que variam entre 30 e 60 FPS, embora algumas aplicações, como a captura de vídeo de alta velocidade, possam usar taxas de quadros muito maiores para permitir a análise detalhada de movimentos rápidos. O FPS é importante, pois a percepção de um movimento contínuo depende da quantidade do número de quadros projetados em um determinado intervalo de tempo (DANTAS, 2022).

Os vídeos são amplamente utilizados em diversas áreas, incluindo entretenimento, comunicação, educação e vigilância. Com o avanço da tecnologia digital, a produção e distribuição de vídeos tornaram-se mais acessíveis, permitindo que indivíduos e organizações criem e compartilhem conteúdo audiovisual com facilidade. De acordo com Batista (2024), a codificação de vídeo, que envolve a compressão de dados visuais e de áudio, é essencial para a transmissão eficiente de vídeos na internet e em redes de transmissão.

Dessa forma, os vídeos desempenham um papel fundamental na sociedade moderna, não apenas como uma ferramenta de entretenimento, mas também como um recurso indispensável para a disseminação de informações, o aprendizado e a análise de eventos em tempo real.

APÊNDICE B – DETECÇÃO DE OBJETOS

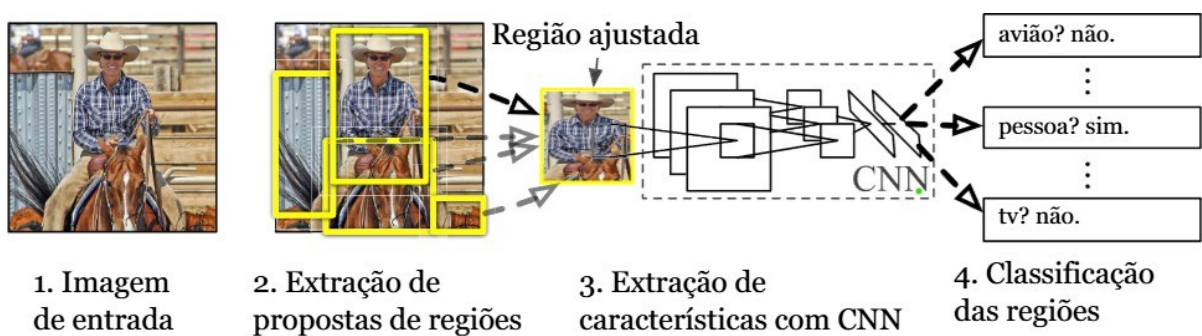
B.1 DETECÇÃO DE OBJETOS (*OBJECT DETECTION* - OD)

A detecção de objetos se destaca como um pilar fundamental da CV, possibilitando a extração de informações de imagens e vídeos. Essa área abrange uma ampla gama de técnicas e metodologias, desde abordagens clássicas até algoritmos modernos baseados em Aprendizado de Máquina (*Machine Learning* - ML), possibilitando a realização de diversas tarefas cruciais, como rastreamento de objetos, segmentação semântica (ZOU *et al.*, 2023), reconhecimento e classificação de objetos. A evolução contínua dessas metodologias tem ampliado significativamente as aplicações práticas e a eficácia em diferentes contextos.

B.1.1 Métodos Convencionais de Detecção de Objetos

Tradicionalmente, os métodos de detecção de objetos se baseiam na extração manual de características das imagens e no uso de classificadores simples para identificar e localizar objetos (KAUR; SINGH, 2023). Estes consistem, geralmente, em três etapas: a seleção de região, onde uma região específica da imagem é identificada e delimitada como uma área de interesse para análise subsequente; extração de características, onde características relevantes são identificadas na imagem; e classificação, onde um modelo é treinado para reconhecer e distinguir entre diferentes classes de objetos com base nas características selecionadas. Esse processo está ilustrado na Figura 28.

Figura 28: Etapas convencionais da detecção de objetos.



Fonte: Adaptado de Kaur & Singh (2023).

Na Figura 28, é apresentado o fluxo das etapas convencionais para a detecção de objetos em imagens. Inicialmente, uma imagem de entrada é fornecida ao sistema, que realiza a extração de propostas de regiões, delimitando áreas de interesse que podem conter objetos relevantes. Em seguida, essas regiões ajustadas passam por uma etapa de extração de características,

onde atributos importantes são identificados manualmente ou por métodos tradicionais, como descritores baseados em textura, cor ou forma.

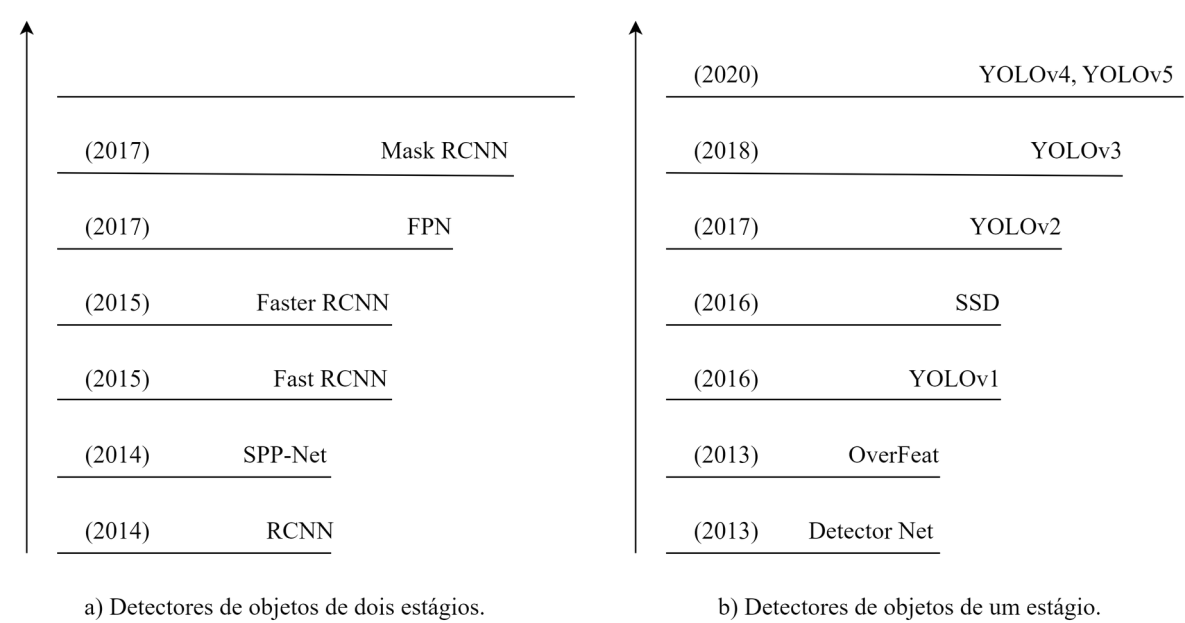
Por fim, ocorre a etapa de classificação, na qual as regiões são analisadas e categorizadas em diferentes classes com base nas características extraídas. Embora a figura ilustre o uso de Redes Neurais Convolucionais (*Convolutional Neural Networks* - CNNs) na extração de características, é importante destacar que este método não é considerado parte das abordagens convencionais, mas sim uma evolução moderna.

Esses métodos representam uma abordagem convencional, porém ainda relevante, na detecção de objetos em imagens, fornecendo os fundamentos para técnicas mais avançadas para aplicações práticas em diferentes domínios.

B.1.2 Métodos Contemporâneos de Detecção de Objetos

Os métodos contemporâneos de detecção de objetos evoluíram significativamente com o advento das Redes Neurais e do Aprendizado Profundo (*Deep Learning* - DL). Essas metodologias superam as abordagens tradicionais em termos de precisão e eficiência, principalmente devido à sua capacidade de aprender representações robustas e de alto nível de características em imagens, sem necessidade de intervenção manual. Alguns dos principais avanços e técnicas utilizadas nos métodos contemporâneos são os Detectores de Um Estágio (*One-Stage Detectors*) e Detectores de Dois Estágios (*Two-Stage Detectors*) amparados pelas CNNs, conforme classificado na Figura 29.

Figura 29: Classificação de modelos genéricos de detecção de objetos.



Fonte: Adaptado de Kaur & Singh (2023).

Na Figura 29, detectores de uma etapa, como YOLO e Single-Shot Detector (SSD), proposto por *Liu et al.* (2016), integram a geração de propostas de regiões e a classificação em uma única rede neural, possibilitando a detecção em tempo real (DIWAN; ANIRUDH; TEMBHURNE, 2023). Por outro lado, os detectores de duas etapas, como R-CNN (*GIRSHICK et al.*, 2014) e Faster R-CNN (*REN et al.*, 2016), geram propostas de regiões antes de realizar a classificação. As CNNs, base dos métodos modernos, aprendem hierarquias de características, desde bordas simples até padrões complexos (*ZOU et al.*, 2023).

Embora os avanços tenham sido significativos, desafios clássicos ainda persistem nessa área. Problemas como a rotação de objetos, mudanças de escala e a detecção de objetos pequenos continuam a impactar a precisão dos modelos. Além disso, a localização precisa de objetos em cenários complexos, bem como a detecção de objetos densamente agrupados ou parcialmente ocluídos, representam barreiras importantes. Outro desafio crítico é acelerar o processo de detecção para atender às demandas de aplicativos em tempo real, o que exige esforços contínuos em pesquisa e desenvolvimento.

APÊNDICE C – RASTREAMENTO DE OBJETOS

C.1 RASTREAMENTO DE OBJETOS (*OBJECT TRACKING* - OT)

O rastreamento se destaca como uma técnica complementar à detecção, concentrando-se em acompanhar a trajetória e o estado de um objeto específico ao longo de um fluxo de imagens (ZAIDI *et al.*, 2022). Além de identificar a presença de objetos, o rastreamento os acompanha em sua jornada, fornecendo informações sobre seu movimento, interações e mudanças de estado.

C.1.1 Abordagens Clássicas de Rastreamento de Objetos

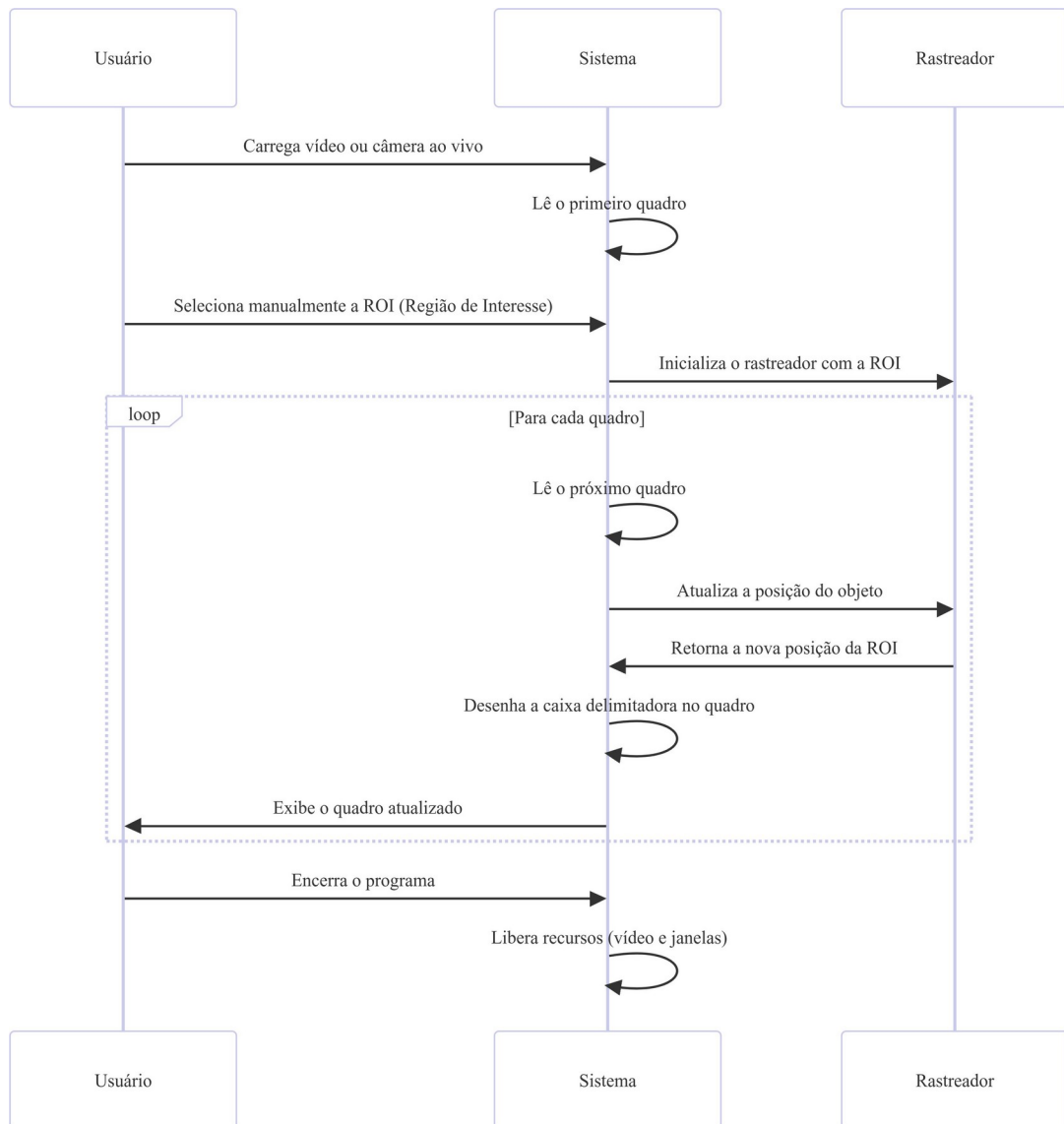
As abordagens tradicionais de Rastreamento de Objetos geralmente envolvem técnicas baseadas em algoritmos que utilizam características visuais dos objetos para segui-los ao longo do tempo em uma sequência de imagens ou vídeos. Popularmente, métodos baseados em características, modelos ou cor e aparência eram amplamente utilizados para OT (ALI *et al.*, 2016). Esses métodos utilizam atributos específicos dos objetos, como forma, textura, cor ou padrões de movimento, para identificar e seguir um objeto em uma sequência de imagens.

Os métodos baseados em características extraem pontos de interesse ou descritores, como cantos ou bordas, os quais são rastreados ao longo do tempo. Por outro lado, os métodos baseados em modelos utilizam representações predefinidas dos objetos, como formas geométricas ou modelos 3D, para prever e atualizar a posição do objeto em movimento. Já os métodos que se baseiam em cor e aparência utilizam informações visuais, como histogramas de cor ou modelos de aparência, para distinguir e rastrear os objetos.

Por outro lado, os modelos probabilísticos, especialmente os filtros, emergiram como uma abordagem poderosa e versátil nesse campo. Ao integrar conceitos fundamentais de probabilidade e estatística, esses filtros oferecem uma estrutura robusta para lidar com a incerteza e variabilidade inerentes aos sistemas dinâmicos. Usando técnicas como o Filtro de Kalman (KALMAN, 1960) e o Filtro de Partículas (LIU; CHEN, 1998), é possível estimar com precisão o estado dos objetos em movimento, mesmo em ambientes desafiadores e cenários complexos. Esses modelos não apenas fornecem uma estimativa da posição atual do objeto, mas também permitem prever sua trajetória futura, fornecendo percepções valiosas para aplicações que vão desde a navegação autônoma de veículos até a vigilância de segurança e o monitoramento de sistemas industriais.

Na Figura 30 é apresentado o diagrama de sequência que descreve o *pipeline* básico de OT utilizando OpenCV, que possui a implementação dos algoritmos clássicos, apresentados no Apêndice D, que utilizam as abordagens descritas acima para realizar o processo de acompanhamento.

Figura 30: Diagrama de sequência do *pipeline* de rastreamento de objetos.



Fonte: Elaborado pelo autor (2024).

O processo apresentado na Figura 30 é iniciado com o usuário, que pode carregar um vídeo previamente gravado ou ativar a câmera ao vivo. O sistema, então, lê o primeiro quadro e solicita ao usuário que selecione manualmente a região de interesse (*Region of Interest* - ROI) do objeto a ser rastreado. Esse processo também pode ser realizado de forma automática utilizando um modelo de detecção previamente treinado. Após a seleção, o rastreador é inicializado com a ROI fornecida.

Durante o *loop* de rastreamento, o sistema lê cada quadro subsequente e solicita ao rastreador que atualize a posição do objeto. O rastreador retorna a nova posição da ROI, que é usada pelo sistema para desenha a caixa delimitadora ao redor do objeto. O quadro atualizado é exibido ao usuário. O processo continua até que o usuário encerre o programa, momento em que

o sistema libera os recursos utilizados. Vale destacar que, em sistemas reais, esse procedimento pode variar conforme a implementação.

C.1.2 Abordagens Modernas de Rastreamento de Objetos

Métodos clássicos de OT, como filtros de Kalman e rastreamento baseado em correspondência de características, ainda são utilizados em diversos cenários. No entanto, nos últimos anos, surgiram abordagens modernas que oferecem maior precisão, robustez e eficiência, como CNNs e Redes Neurais Recorrentes (*Recurrent Neural Networks* - RNNs), modelos de movimento baseados em redes Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), entre outros.

Essas técnicas permitem acompanhar objetos específicos mesmo em cenários complexos, como quando eles se movem, mudam de forma ou são parcialmente ocluídos por outros objetos. Mesmo com toda a evolução dos métodos utilizados para o OT, o desempenho dos rastreadores é sensível a fatores em torno do alvo e à própria variabilidade do alvo, como a representação da aparência do alvo, mudanças de iluminação, velocidade de movimento do alvo, alterações do cenário, oclusão, estabilidade da câmera e velocidade de processamento dos rastreadores, como exposto por *Chen et al.* (2022).

Os rastreadores SORT (*BEWLEY et al.*, 2016) e DeepSORT (*WOJKE; BEWLEY; PAULUS*, 2017) representam avanços significativos em técnicas de OT, combinando a simplicidade e eficiência com uma precisão aprimorada. O SORT utiliza um filtro de Kalman para prever o movimento dos objetos e um algoritmo de associação baseado em distância, permitindo rastreamento em tempo real. Já o DeepSORT melhora esse processo ao incorporar CNNs para extrair características de aparência dos objetos, tornando o rastreamento mais robusto em situações de oclusão ou alta densidade de alvos semelhantes.

Recentemente, algoritmos ainda mais avançados têm sido empregados para melhorar o rastreamento em tempo real, como BoT-SORT (*AHARON; ORFAIG; BOBROVSKY*, 2022) e ByteTrack (*ZHANG et al.*, 2022), ambos compatíveis com o YOLO. O BoT-SORT aprimora o DeepSORT ao integrar refinamentos na associação de detecções, como modelos de reidentificação (ReID) e ajustes no algoritmo de correspondência. O ByteTrack, por sua vez, melhora a associação de detecções ao considerar não apenas as previsões de alta confiança, mas também as de baixa pontuação, garantindo uma melhor continuidade do rastreamento.

APÊNDICE D – ALGORITMOS DE RASTREAMENTO DE OBJETOS

D.1 ALGORITMOS DE RASTREAMENTO DE OBJETOS

Neste apêndice, é apresentada uma revisão da literatura sobre as metodologias de rastreamento de objetos, com ênfase em como os algoritmos clássicos operam e se diferenciam em suas abordagens. Existe na literatura uma ampla gama de técnicas de rastreamento, cada uma com características e aplicações distintas. Esta revisão busca descrever os princípios de funcionamento de algoritmos populares, como Boosting, Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF ou simplesmente CSRT), Kernelized Correlation Filters (KCF), MedianFlow, Multiple Instance Learning (MIL), Minimum Output Sum of Squared Error (MOSSE) e Tracking, Learning, and Detection (TLD), todos eles com implementações disponíveis na biblioteca OpenCV.

D.1.1 Boosting

O método de rastreamento Boosting (FREUND; SCHAPIRE *et al.*, 1996), implementado pela OpenCV, utiliza a versão apresentada por Grabner, Grabner & Bischof (2006), que se baseia no algoritmo AdaBoost (FREUND; SCHAPIRE; ABE, 1999) adaptado para seleção de características em tempo real. Caracterizada pela capacidade de treinamento *online*, essa implementação permite que o classificador se ajuste dinamicamente durante o rastreamento do objeto. Isso possibilita que o algoritmo lide de forma robusta com variações na aparência do objeto, como mudanças de iluminação, rotações fora do plano e oclusões parciais.

Além disso, o algoritmo utiliza informações do fundo como exemplos negativos, aumentando sua resistência ao problema de *drift*. Esse problema ocorre quando o rastreador perde precisão ao longo do tempo devido a mudanças na aparência ou na posição do objeto. Essa abordagem permite, ainda, a seleção de características mais discriminativas entre o objeto e o fundo.

A eficiência do Boosting é alcançada através do uso de características computacionalmente rápidas, como as *Haar-like wavelets*. Essas características são baseadas em variações de intensidade entre regiões adjacentes da imagem, capturando mudanças abruptas, como bordas e texturas. Ademais, são utilizados histogramas de orientação e padrões binários locais, o que permite a execução do método em tempo real.

A principal ideia do algoritmo *on-line boosting* é a introdução dos seletores, que são responsáveis por atualizar de forma incremental os classificadores fracos durante o processo de rastreamento. Cada seletor mantém um conjunto separado de classificadores fracos e, quando uma nova amostra de treinamento é recebida, esses classificadores são atualizados. O melhor classificador fraco, ou seja, aquele que apresenta o menor erro, é então selecionado. O erro de cada classificador fraco é estimado com base nas amostras já processadas, e o seletor escolhe

aquele com o menor erro estimado. Matematicamente, isso é expresso pela Equação D.1:

$$h^{\text{sel}}(x) = h_m^{\text{weak}}(x) \quad (\text{D.1})$$

Na Equação D.1, m é o classificador fraco selecionado de acordo com o critério de otimização, sendo $m = \arg \min_i e_i$, em que e_i é o erro estimado do classificador fraco h^{weak} .

A complexidade do processo é determinada pelo número de seletores, mas a maior parte do tempo de processamento é consumida pela atualização dos classificadores fracos. Para otimizar esse processo, uma modificação importante proposta é o uso de um único conjunto global de classificadores fracos (*pool*), compartilhado entre todos os seletores, em vez de usar *pools* separados para cada um. Isso permite que, para cada nova amostra, todos os classificadores fracos sejam atualizados apenas uma vez, economizando tempo computacional. Após a atualização, cada seletor escolhe o classificador fraco com o erro mais baixo, baseado no valor estimado de λ , e o peso de importância do classificador é passado para o próximo seletor. O classificador forte é então calculado, como mostrado na Equação D.2:

$$h^{\text{Strong}}(x) = \text{sign}(\text{conf}(x)) \quad (\text{D.2})$$

Na Equação D.2, $\text{conf}(x)$ é a combinação linear dos seletores, dada pela Equação D.3:

$$\text{conf}(x) = \sum_{n=1}^N \alpha_n \cdot h_n^{\text{el}}(x) \quad (\text{D.3})$$

A Equação D.3 exibe a fórmula na qual α_n são os pesos associados a cada seletor $h_n^{\text{el}}(x)$.

Esse ciclo de atualizações é repetido até que todos os seletores sejam atualizados, resultando em um classificador forte a cada passo do processo.

Por fim, para aumentar a diversidade entre os classificadores fracos e permitir que o algoritmo se adapte às mudanças no ambiente, o pior classificador fraco do *pool*, conjunto compartilhado de classificadores fracos, é substituído por um novo classificador escolhido aleatoriamente. Isso garante que o algoritmo permaneça flexível, mesmo diante de mudanças nas condições de rastreamento.

D.1.2 Discriminative Correlation Filter with Channel and Spatial Reliability (CSR-DCF ou CSRT)

O rastreador CSR-DCF, ou simplesmente CSRT, introduzido por Lukezic *et al.* (2017), representa um avanço significativo no uso de filtros de correlação discriminativos. Sua principal inovação é a incorporação de um mapa de confiabilidade espacial, que ajusta dinamicamente o filtro às regiões mais adequadas do objeto para rastreamento. Essa abordagem permite lidar com desafios como oclusões parciais, formas irregulares e mudanças no ambiente,

além de ampliar a região de busca sem comprometer a precisão do rastreamento. Diferentemente de métodos que

assumem alvos com formas retangulares alinhadas aos eixos, o CSRT evita que o filtro aprenda informações irrelevantes do fundo, superando essas limitações.

A formulação básica do rastreador é descrita pela Equação D.4, que define a probabilidade da posição do objeto x dado o conjunto de filtros h :

$$p(x|h) = \prod_{d=1}^{N_d} p(x|f_d)p(f_d) \quad (D.4)$$

em que f_d representa os canais de características, $p(x|f_d)$ é a convolução do mapa de características f_d com o filtro h_d , e $p(f_d)$ é um peso que reflete a confiabilidade do canal.

Os filtros h_d são aprendidos minimizando a soma dos erros quadráticos entre as saídas de correlação e a resposta desejada g , conforme a Equação D.5:

$$\arg \min_h \prod_{d=1}^{N_d} \|f_d * h_d - g\|^2 + \lambda \|h_d\|^2 \quad (D.5)$$

em que $*$ denota a operação de correlação, e λ é um parâmetro de regularização. Essa formulação é resolvida no domínio da frequência utilizando a Transformada de Fourier, o que permite uma solução eficiente.

Uma das principais inovações do CSRT é a introdução do mapa de confiabilidade espacial m , que restringe o suporte do filtro às regiões mais confiáveis do objeto. Isso é incorporado ao aprendizado do filtro por meio de uma restrição $h \equiv m \odot h$, na qual $\odot$ representa a multiplicação ponto a ponto. Essa abordagem é resolvida utilizando o método dos multiplicadores de Lagrange, conforme descrito na Equação D.6:

$$L(h_c, h, l|m) = \|h_c * f - g\|^2 + \lambda \|h_m\|^2 + l^H (h_c - h_m) + \mu \|h_c - h_m\|^2 \quad (D.6)$$

em que h_c é uma variável dual, $h_m = m \odot h$, l é o multiplicador de Lagrange, e μ é

um parâmetro de penalização.

Outro aspecto importante do CSRT é o uso de confiabilidade de canal, que atribui pesos aos diferentes canais de características com base na qualidade das respostas aprendidas. Essa ponderação reduz ruídos nas estimativas e melhora a precisão durante o processo de localização. O método foi projetado para ser eficiente, utilizando características padrão, como HoGs, técnica para extração de características baseada na análise da intensidade e direção dos gradientes de uma imagem, e ColorNames, representações baseadas em histogramas de cores que associam valores RGB a nomes de cores perceptíveis. Isso permite sua execução em tempo real, mesmo em processadores convencionais.

D.1.3 Kernelized Correlation Filters (KCF)

Apresentado por *Henriques et al.* (2014), o método KCF representa um avanço significativo no rastreamento visual de alta velocidade, integrando técnicas de aprendizado

discriminativo

com ferramentas analíticas do domínio da frequência. O KCF utiliza filtros de correlação otimizados por meio de regressão de *kernel*, permitindo que milhares de amostras transladas e escaladas sejam incorporadas de maneira eficiente, sem a necessidade de processá-las explicitamente. Essa eficiência é alcançada explorando propriedades de matrizes circulantes e a Transformada Discreta de Fourier, reduzindo drasticamente o custo computacional e o armazenamento, fatores limitantes em rastreadores tradicionais.

O funcionamento do KCF é descrito pela Equação D.7, que define o problema de otimização para aprender o filtro h :

$$h = \arg \min_h \|x * h - y\|^2 + \lambda \|h\|^2 \quad (\text{D.7})$$

em que x é o *patch* de entrada, y é a resposta desejada, $*$ denota a operação de correlação, e λ é um parâmetro de regularização.

A solução no domínio da frequência é dada pela Equação D.8:

$$\hat{h} = \frac{\hat{y} \odot \hat{x}^*}{\hat{x} \odot \hat{x}^* + \lambda} \quad (\text{D.8})$$

dentro da qual $\hat{h}$, $\hat{y}$ e $\hat{x}$ são as representações no domínio da frequência de h , y e x , respectivamente, $\odot$ denota a multiplicação ponto a ponto, e $\hat{x}^*$ é o conjugado complexo de $\hat{x}$. Essa solução aproveita a propriedade de diagonalização das matrizes circulantes no domínio da frequência, reduzindo significativamente o custo computacional.

Uma das inovações centrais do KCF é sua capacidade de manter a mesma complexidade computacional dos filtros de correlação lineares, mesmo ao integrar o método de kernel para aprendizado não linear. A versão kernelizada do KCF utiliza a Equação D.9 para calcular a resposta do rastreador:

$$f(z) = F^{-1} \hat{k}_{xz} \odot \hat{\alpha} \quad (\text{D.9})$$

no qual $\hat{k}_{xz}$ é o kernel no domínio da frequência entre o patch de entrada z e o patch de referência x , e $\hat{\alpha}$ é o filtro aprendido no domínio da frequência.

Além disso, o método permite a extensão para múltiplos canais de características, como no caso do *Dual Correlation Filter*, que combina as vantagens dos filtros de correlação lineares com o uso de *kernels* eficientes. Testado em um *benchmark* com 50 vídeos, o KCF superou rastreadores consagrados, como o TLD, apresentando alto desempenho em taxas de até centenas de quadros por segundo.

D.1.4 MedianFlow

O método MedianFlow, proposto por Kalal, Mikolajczyk & Matas (2010), introduz uma abordagem inovadora para detecção de falhas no rastreamento de objetos, baseando-se no erro *Forward-Backward* (FB). Esse erro avalia a consistência das trajetórias ao rastrear um ponto tanto no sentido temporal direto quanto no reverso. Caso as trajetórias apresentem discrepâncias

significativas, o rastreamento é considerado incorreto. Essa métrica permite identificar falhas com alta confiabilidade e selecionar trajetórias robustas em sequências de vídeo. Ademais, o método é complementar à Correlação Cruzada Normalizada (*Normalized Cross-Correlation* - NCC), comumente usada para validação de rastreamento.

O erro FB, que é a base do rastreador MedianFlow, é definido pela Equação D.10:

$$\text{Erro}_{FB} = \|p - p'\| \quad (\text{D.10})$$

na qual p representa a posição inicial de um ponto rastreado no sentido direto (*forward*) e p' é a posição do mesmo ponto após o rastreamento reverso (*backward*). O erro é calculado como a norma da diferença entre essas duas posições. Se o Erro_{FB} exceder um limiar pré-definido, o rastreamento é considerado falho.

Além disso, o rastreador utiliza a NCC para validar as trajetórias, conforme descrito na Equação D.11:

$$\text{NCC}(I, T) = \frac{\sum_{x,y} (I(x,y) - \bar{I})(T(x,y) - \bar{T})}{\sqrt{\sum_{x,y} (I(x,y) - \bar{I})^2} \sqrt{\sum_{x,y} (T(x,y) - \bar{T})^2}} \quad (\text{D.11})$$

em que $I(x, y)$ é a região da imagem, $T(x, y)$ é o modelo do objeto rastreado, e $\bar{I}$ e $\bar{T}$ são as médias de intensidade de I e T , respectivamente. A NCC mede a similaridade entre o modelo e a região da imagem, sendo usada para validar a qualidade do rastreamento.

Baseado no erro FB (Equação D.10), o rastreador MedianFlow foi desenvolvido para melhorar a precisão do rastreamento e a detecção de falhas. Ao medir as discrepâncias entre as trajetórias rastreadas para frente e para trás, o algoritmo pode identificar de maneira eficaz quando o rastreamento falha devido a oclusões, movimentos rápidos ou mudanças significativas na aparência do objeto. Essa abordagem fornece uma maneira robusta de selecionar as trajetórias mais confiáveis, eliminando aquelas que apresentam inconsistências, como o erro de *drift*.

D.1.5 Multiple Instance Learning (MIL)

O rastreador MILTrack, ou simplesmente MIL, proposto por Babenko, Yang & Belongie (2010), utiliza uma abordagem inovadora para o rastreamento de objetos, incorporando o conceito de aprendizado por múltiplas instâncias (*Multiple Instance Learning* - MIL). Diferente dos métodos tradicionais de aprendizado supervisionado, que exigem exemplos rotulados individualmente, o MILTrack permite que o sistema aprenda a partir de conjuntos de exemplos, conhecidos como *bags*, onde o rótulo é atribuído ao conjunto, não ao exemplo específico.

No contexto do MIL, a relação entre os rótulos dos *bags* e das instâncias individuais é descrita pela Equação D.12:

$$y = \max_i y_i \quad (\text{D.12})$$

na qual y é o rótulo do *bag*, e y_i são os rótulos das instâncias individuais. Um *bag* é considerado positivo ($y = 1$) se pelo menos uma de suas instâncias for positiva ($y_i = 1$); caso contrário, o *bag* é negativo ($y = 0$).

Para rastreamento, o MILTrack utiliza um modelo de aparência adaptativo baseado em um classificador discriminativo que estima a probabilidade $P(y|x)$, onde x é um *patch* de imagem e y é uma variável binária indicando a presença do objeto. A atualização do modelo é realizada com base em um algoritmo de *online boosting*, que otimiza a função de perda descrita na Equação D.13:

$$L = \sum_{j=1}^N \log \left(1 + \exp(-y_j \cdot \max_i f(x_{ij})) \right) \quad (D.13)$$

em que $f(x_{ij})$ é a resposta do classificador para a instância x_{ij} no *bag* j , e y_j é o rótulo do *bag*. Essa formulação permite que o rastreador lide com incertezas nos dados de treinamento, como oclusões parciais ou mudanças na aparência do objeto.

Além disso, o MILTrack utiliza um modelo de movimento simples para restringir a busca do objeto em cada quadro a uma região ao redor da posição estimada no quadro anterior. A nova posição do objeto é determinada pela maximização da probabilidade $P(y|x)$ sobre os *patches* candidatos, conforme a Equação D.14:

$$\ell_t = \arg \max_{x \in X_s} P(y|x) \quad (D.14)$$

no caso em que ℓ_t é a posição estimada do objeto no quadro t , e X_s é o conjunto de *patches* dentro de uma região de busca ao redor da posição anterior.

O MILTrack evita problemas comuns como o *drift* do objeto, proporcionando uma atualização mais robusta do modelo de aparência adaptativo, com um número mínimo de ajustes de parâmetros. O modelo adaptativo de aparência baseado em MIL, além de ser eficaz para rastreamento com baixa sobrecarga computacional, mostrou-se capaz de lidar com uma ampla variedade de condições dinâmicas, tornando-o uma solução poderosa para rastreamento de objetos em ambientes complexos.

D.1.6 Minimum Output Sum of Squared Error (MOSSE)

O MOSSE, conforme apresentado por Bolme et al. (2010), é uma técnica de rastreamento visual baseada em filtros de correlação adaptativos. Ao contrário de outros métodos tradicionais de rastreamento, que utilizam filtros estáticos ou são sensíveis a pequenas imprecisões nos exemplos de treinamento, o MOSSE adapta-se dinamicamente durante o processo de rastreamento, criando filtros robustos a mudanças na aparência do objeto, como variações de escala, rotação e iluminação.

A principal inovação do MOSSE é a capacidade de gerar filtros de correlação que minimizam a Soma dos Erros Quadráticos (*Sum of Squared Error* - SSE) entre a saída desejada

e a saída real da correlação. A formulação do problema de otimização é descrita pela Equação D.15:

$$\min_H \sum_i \|F_i H^* - G_i\|^2 \quad (\text{D.15})$$

na qual F_i e G_i são as FFTs do *patch* de entrada f_i e da saída desejada g_i , respectivamente, e H^* é o filtro de correlação no domínio da frequência. A solução para H^* é obtida de forma fechada, conforme a Equação D.16:

$$H^* = \frac{G_i F_i^*}{F_i F_i^*} \quad (\text{D.16})$$

Na Equação D.16, F_i^* é o conjugado complexo de F_i . Essa solução utiliza a relação elementar entre entrada e saída no domínio da frequência, garantindo eficiência computacional.

Durante o rastreamento, o MOSSE utiliza a correlação no domínio da frequência para localizar o objeto. A correlação é calculada pela Equação D.17:

$$G = F \odot H^* \quad (\text{D.17})$$

na qual G é a saída da correlação, F é a Transformada de Fourier do *patch* de entrada, H^* é o filtro de correlação, e $\odot$ denota a multiplicação ponto a ponto. A saída G é então transformada de volta para o domínio espacial utilizando a Transformada inversa de Fourier.

Uma característica importante do MOSSE é a detecção de falhas no rastreamento utilizando a razão Pico-para-Lóbulos Laterais (*Peak-to-Sidelobe Ratio* - PSR). O PSR é calculado pela Equação D.18:

$$\text{PSR} = \frac{g_{\max} - \mu_{\text{sidelobe}}}{\sigma_{\text{sidelobe}}} \quad (\text{D.18})$$

em que $g_{\max}$ é o valor máximo da saída de correlação, μ_{sidelobe} e σ_{sidelobe} são a média e o desvio padrão dos valores fora da região do pico, respectivamente. Um PSR baixo indica uma possível falha no rastreamento, permitindo que o rastreador pause e retome o processo quando o objeto reaparece.

O MOSSE é altamente eficiente, com uma complexidade computacional de $O(P \log P)$, onde P é o número de *pixels* no *patch* de rastreamento. Isso permite que o método processe até 669 quadros por segundo (BOLME et al., 2010).

D.1.7 Tracking, Learning and Detection (TLD)

Sugerido por Kalal, Mikolajczyk & Matas (2011), o TLD é um *framework* projetado para rastreamento de longo prazo de objetos desconhecidos em fluxos de vídeo. A principal inovação do TLD está na decomposição da tarefa de rastreamento em três componentes complementares: rastreamento, aprendizado e detecção. O rastreador estima o movimento do objeto entre quadros consecutivos, assumindo que as mudanças de movimento sejam limitadas e que

o objeto esteja visível. A detecção, por outro lado, trata cada quadro de forma independente, localizando todas as aparições aprendidas do objeto ao longo do tempo. Já o componente de aprendizado monitora o desempenho do rastreador e do detector, identificando erros e atualizando o modelo de detecção para corrigir falhas futuras.

Para garantir a robustez do sistema, o TLD utiliza o paradigma de aprendizado P-N (*Positive-Negative*), no qual dois avaliadores avaliam os erros do detector: o *P-Expert*, responsável por identificar detecções perdidas, e o *N-Expert*, que detecta falsos positivos. Esses avaliadores atualizam o conjunto de treinamento do detector, conforme descrito na Equação D.19:

$$T_{\text{new}} = T_{\text{old}} \cup T_{\text{P-Expert}} \setminus T_{\text{N-Expert}} \quad (\text{D.19})$$

Na Equação D.19, T_{new} é o conjunto de treinamento atualizado, T_{old} é o conjunto de treinamento anterior, $T_{\text{P-Expert}}$ são as detecções perdidas identificadas pelo *P-Expert*, e $T_{\text{N-Expert}}$ são os falsos positivos identificados pelo *N-Expert*.

O rastreamento no TLD é realizado utilizando o rastreador MedianFlow, que avalia a consistência do movimento do objeto entre quadros consecutivos. A detecção, por sua vez, utiliza um classificador treinado para localizar o objeto em cada quadro. A integração entre rastreamento e detecção é feita por meio de um mecanismo de validação cruzada, onde o detector pode corrigir falhas do rastreador e vice-versa.

Em seguida, o TLD incorpora estratégias de recuperação em casos de falha, como a detecção de oclusões ou quando o objeto sai do campo de visão. A detecção de falhas é baseada no erro *Forward-Backward*, calculado pela Equação D.20:

$$\text{FB Error} = \|\ell_{\text{fwd}} - \ell_{\text{bwd}}\| \quad (\text{D.20})$$

na qual ℓ_{fwd} é a posição estimada pelo rastreador no sentido direto (quadro t para $t + 1$) e ℓ_{bwd} é a posição estimada no sentido inverso (quadro $t + 1$ para t).

Essa métrica avalia o erro no nível global, considerando a posição do objeto como um todo. No entanto, o TLD também utiliza o erro *Forward-Backward* no nível local, como definido na Equação D.10, para pontos rastreados individualmente. A diferença entre essas abordagens está no nível de análise. Enquanto a Equação D.20 verifica a consistência do rastreamento global do objeto, a Equação D.10 avalia a confiabilidade de pontos específicos. Ambas as métricas são complementares e contribuem para a robustez do rastreador.

O TLD é capaz de operar em tempo real, lidando com desafios como mudanças de escala, iluminação, oclusões parciais e movimentos complexos. Testes extensivos em cenários desafiadores mostraram que o TLD supera métodos convencionais, destacando-se como uma solução eficiente e adaptativa para rastreamento de longo prazo. Apesar de suas limitações, como dificuldades com rotações fora do plano e objetos articulados, o TLD oferece um equilíbrio robusto entre rastreamento, aprendizado contínuo e detecção precisa.

APÊNDICE E – MÉTRICAS UTILIZADAS

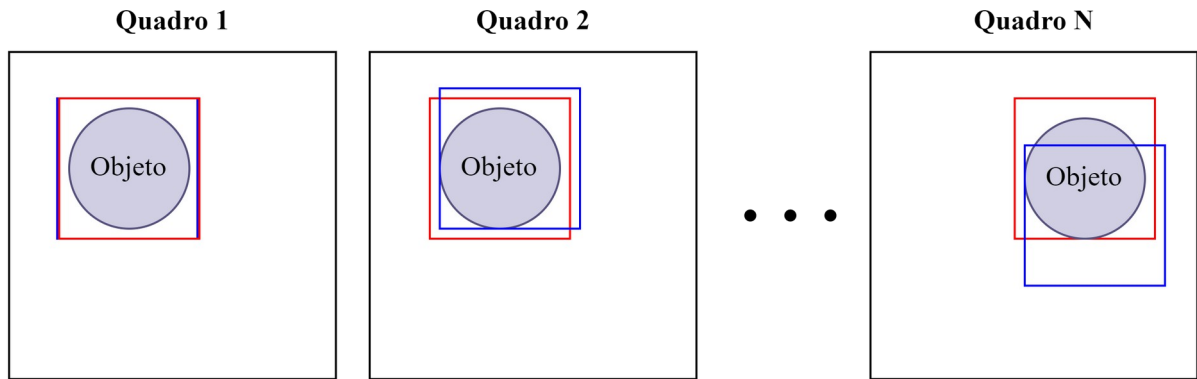
E.1 MÉTRICAS UTILIZADAS

A avaliação de algoritmos de rastreamento de objetos é um aspecto essencial para determinar sua eficácia e adequação em aplicações práticas. A seguir, são descritas as principais métricas utilizadas neste trabalho.

E.1.1 Interseção sobre União (*Intersection over Union - IoU*)

A métrica *Intersection over Union* (IoU) é amplamente utilizada para avaliar a qualidade de detecções e rastreamento de objetos, medindo a sobreposição entre a caixa delimitadora prevista (*predicted bounding box*) e a caixa delimitadora real (*ground truth bounding box*) (BRDJANIN *et al.*, 2020; ROMÃO, 2023). Essa métrica é particularmente útil para quantificar a precisão do rastreamento, ao comparar a proximidade entre a previsão do modelo e a posição real do objeto. A Figura 31 ilustra o conceito de IoU.

Figura 31: Representação do processo de IoU.



Fonte: Elaborado pelo autor (2024).

Na Figura 31, a linha vermelha representa a caixa delimitadora real (*ground truth*), enquanto a linha azul representa a caixa delimitadora gerada pelo rastreador. Nos primeiros dois quadros, a IoU apresenta valores elevados, indicando uma sobreposição significativa entre as caixas, o que sugere que o rastreador está funcionando corretamente. No entanto, à medida que o vídeo avança, a IoU pode diminuir, como no quadro *N*, indicando possíveis falhas no rastreamento devido a fatores como oclusões, mudanças de iluminação ou deformações do objeto.

A métrica IoU é formalmente definida pela Equação E.1, que calcula a razão entre a área de interseção (A_{inter}) das duas caixas e a área de união (A_{union}).

$$\text{IoU} = \frac{A_{\text{inter}}}{A_{\text{union}}} \quad (\text{E.1})$$

Nessa equação, A_{inter} corresponde à área comum entre a caixa prevista (B_{pred}) e a caixa real (B_{gt}), enquanto A_{union} é obtida somando as áreas das caixas previstas e reais e subtraindo a área de interseção, como mostrado na Equação E.2.

$$A_{\text{union}} = A_{\text{pred}} + A_{\text{gt}} - A_{\text{inter}} \quad (\text{E.2})$$

Na Equação E.2, A_{pred} representa a área da caixa prevista (B_{pred}) e A_{gt} a área da caixa real (B_{gt}).

Os valores de IoU variam entre 0 e 1. IoU igual a 1 indica uma sobreposição perfeita entre as caixas, ou seja, o rastreador está completamente alinhado com a posição real do objeto. IoU igual a 0 indica que não há sobreposição entre as caixas, sugerindo que o rastreador falhou em localizar o objeto.

E.1.2 Erro Médio de Rastreamento (*Mean Tracking Error - MTE*)

O Erro Médio de Rastreamento (*Mean Tracking Error - MTE*) é uma métrica que quantifica a precisão geométrica de um rastreador ao longo do tempo. Ele mede o desvio médio entre a posição estimada do objeto e sua posição real em uma sequência de quadros. Essa métrica é derivada do Erro de Rastreamento (*Tracking Error - TE*), que avalia o erro em um único quadro. O TE é formalmente definido pela Equação E.3.

$$\text{TE}_i = \|\mathbf{p}_{\text{estimada},i} - \mathbf{p}_{\text{real},i}\| \quad (\text{E.3})$$

Na Equação E.3, $\mathbf{p}_{\text{estimada},i}$ representa o vetor de posição estimada do objeto no quadro i , enquanto $\mathbf{p}_{\text{real},i}$ representa o vetor de posição real do objeto no mesmo quadro. O termo $\|\mathbf{p}_{\text{estimada},i} - \mathbf{p}_{\text{real},i}\|$ corresponde à norma euclidiana da diferença entre os dois vetores, que calcula a distância entre a posição estimada e a posição real no quadro i . O TE fornece uma medida pontual do erro de rastreamento em um único quadro.

O MTE, por sua vez, é a média dos valores de TE ao longo de todos os quadros analisados. Ele é formalmente definido pela Equação E.4.

$$\text{MTE} = \frac{1}{N} \sum_{i=1}^N \text{TE}_i = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_{\text{estimada},i} - \mathbf{p}_{\text{real},i}\| \quad (\text{E.4})$$

Na Equação E.4, N representa o número total de quadros analisados na sequência. O termo TE_i é o erro de rastreamento no quadro i , conforme definido na Equação E.3. O MTE fornece uma medida escalar que varia de 0 a valores positivos, onde valores mais baixos indicam maior precisão no rastreamento.

O MTE é amplamente utilizado para avaliar o desempenho de algoritmos de rastrea-

mento, especialmente em cenários desafiadores, como mudanças de iluminação, oclusões ou

deformações do objeto. Ele é uma métrica robusta para comparar diferentes abordagens de rastreamento, fornecendo uma visão quantitativa da precisão geométrica ao longo do tempo.

E.1.3 Quadros por Segundo (*Frames Per Second* - FPS)

O FPS mede o desempenho computacional do algoritmo de rastreamento em termos de quadros processados por segundo, como apresentado na Equação E.5. Essa métrica é fundamental para avaliar a viabilidade do algoritmo em aplicações em tempo real, onde é necessário processar e atualizar a posição do objeto em alta velocidade.

$$\text{FPS} = \frac{N_{\text{quadros}}}{T_{\text{processamento}}} \quad (\text{E.5})$$

A Equação E.5 define N_{quadros} como o número total de quadros processados pelo algoritmo, e $T_{\text{processamento}}$ como o tempo total necessário para processar esses quadros. Quanto mais altos os valores de FPS, indicam maior eficiência computacional, sendo particularmente importantes em cenários com restrições de tempo real, como monitoramento de vídeo ao vivo ou aplicações em sistemas embarcados.

Apesar de sua utilidade, o número de quadros por segundo, por si só, não é suficiente para demonstrar resultados completos, pois apresenta algumas limitações elencadas a seguir:

- **Não reflete a qualidade do rastreamento:** o FPS mede apenas o desempenho computacional, mas não leva em consideração a precisão ou a eficácia do rastreamento. É possível que um rastreador com FPS alto tenha um desempenho ruim.
- **Variação com a complexidade do algoritmo:** algoritmos de rastreamento mais complexos podem apresentar uma taxa de FPS menor, mas isso não significa necessariamente que sejam piores. A complexidade pode estar associada a melhores resultados em termos de precisão e robustez.
- **Dependência do hardware:** o FPS é fortemente influenciado pela capacidade do hardware em que o algoritmo está sendo executado. Isso pode gerar resultados inconsistentes quando comparados entre diferentes plataformas, dificultando uma análise justa de desempenho.
- **Não considera a adaptação a diferentes cenários:** um algoritmo pode ter uma alta taxa de FPS em um ambiente controlado, mas essa métrica não captura como ele se adapta a diferentes tipos de movimento, condições de iluminação ou interferências externas.

E.1.4 Robustez

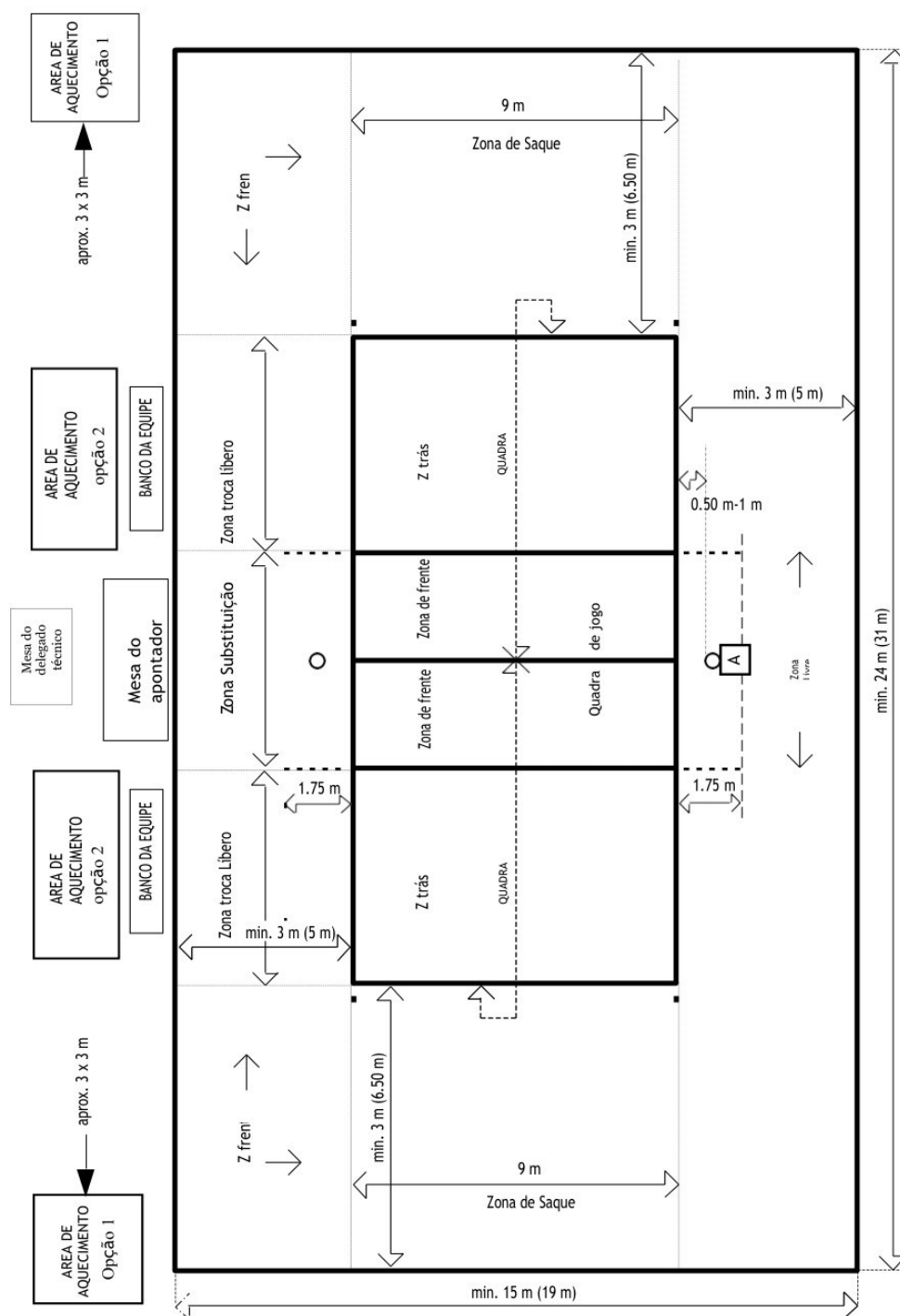
A robustez de um algoritmo de rastreamento mede sua capacidade de lidar com cenários de alta complexidade e situações adversas, como mudanças abruptas de iluminação, oclusões parciais ou totais, variações de escala e objetos em movimento rápido. Essa característica é fundamental para avaliar a adaptabilidade do algoritmo, especialmente em ambientes dinâmicos e imprevisíveis, onde o rastreador precisa continuar funcionando de maneira precisa e confiável.

A robustez será avaliada por meio de análise visual, em que observações qualitativas serão feitas ao longo dos vídeos de teste. Nessa avaliação, serão verificadas situações como:

- **Mudanças de iluminação:** a capacidade do rastreador de manter o objeto rastreado apesar de variações na intensidade e na direção da luz.
- **Oclusões:** como o rastreador lida com o objeto parcialmente oculto ou totalmente fora do campo de visão por algum período de tempo.
- **Movimento rápido:** a habilidade do algoritmo de acompanhar objetos em movimento rápido sem perder o rastreamento ou introduzir erros significativos.
- **Variações de escala e rotação:** a capacidade de adaptação do rastreador quando o objeto muda de tamanho ou posição no campo de visão.

ANEXO A – ESPECIFICAÇÕES DA QUADRA DE VÔLEI

Figura 32: Especificações da Quadra de Vôlei.



Fonte: Confederação Brasileira de Voleibol (2021).