

ANÁLISE DE USABILIDADE DE ALGORITMOS DE RASTREAMENTO E DETECÇÃO DE OBJETOS

USABILITY ANALYSIS OF OBJECT TRACKING AND DETECTION ALGORITHMS

Lucas B. Palácios¹, Matheus H. E. Candial², Jefferson A. R. Passerini³

¹Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, lucas.palacios@fatec.sp.gov.br

²Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, matheus.candial@fatec.sp.gov.br

³Faculdade de Tecnologia Prof. José Camargo – Fatec Jales, jefferson.passerini@fatec.sp.gov.br

Informação e Comunicação

Subárea: Processamento de Imagens e Visão Computacional

RESUMO

O artigo aborda a crescente integração da tecnologia à vida cotidiana, com enfoque nos setores de segurança, esportes e indústria, destacando o uso de algoritmos de rastreamento de objetos (Boosting, MedianFlow, CSRT, KCF, TLD, MOSSE e MIL) baseados em visão computacional para analisar registros de vídeo e extrair informações relevantes. O objetivo central é apresentar e analisar algoritmos de detecção e rastreamento de objetos, com ênfase em suas diferenças, vantagens e desvantagens. O OpenCV é citado como uma biblioteca crucial, com módulos específicos para visão computacional e aprendizado de máquina. A relevância da classificação na visão computacional é ressaltada, explicando-a como um processo para determinar a classe de uma entrada com base em categorias predefinidas. Destaca-se a importância dos classificadores em sistemas de visão computacional, como reconhecimento de imagens e detecção de defeitos. Foram testados os algoritmos de rastreamento: Boosting, MedianFlow, CSRT, KCF, TLD, MOSSE e MIL, evidenciando suas aplicações e características distintas. A análise dos resultados destaca o desempenho do algoritmo CSRT no rastreamento de objetos, inclusive em condições noturnas, embora enfrente dificuldades em casos de oclusão total. Sugere-se a integração de algoritmos de detecção nessas situações como crucial. Outros algoritmos, como KCF e MedianFlow, apresentaram resultados semelhantes, com ênfase na eficácia do KCF em condições desfavoráveis. A conclusão enfatiza a necessidade de uma análise cuidadosa da situação para escolher o algoritmo mais adequado, recomendando uma apresentação clara e objetiva dos resultados, incluindo ilustrações e tabelas estatísticas, juntamente com a consideração de observações de outros autores para enriquecer a discussão. Palavras-chave: rastreamento de objetos; visão computacional; CSRT; KCF; medianflow.

ABSTRACT

The article addresses the growing integration of technology in everyday life, with a focus on the security, sports, and industrial sectors, highlighting the use of vision-based object tracking algorithms (Boosting, MedianFlow, CSRT, KCF, TLD, MOSSE and MIL) computational to analyze video records and extract relevant information. The main goal is to present and analyze object detection and tracking algorithms, with an emphasis on their differences, advantages and disadvantages. OpenCV is cited as a crucial library, with specific modules for computer vision and machine learning. The relevance of classification in computer vision is highlighted, explaining it as a process to determine the class of an input based on predefined categories. The importance of classifiers in computer vision systems, such as image recognition and defect detection, is highlighted. The following tracking algorithms were tested: Boosting, MedianFlow, CSRT, KCF, TLD, MOSSE and MIL, highlighting their distinct applications and characteristics. Analysis of the results highlights the performance of the CSRT algorithm in object tracking, including in nighttime conditions, although it faces difficulties in cases of total

occlusion. The integration of detection algorithms in these situations is suggested as crucial. Other algorithms, such as KCF and MedianFlow, showed similar results, with an emphasis on the effectiveness of KCF in unfavorable conditions. The conclusion emphasizes the need for a careful analysis of the situation to choose the most appropriate algorithm, recommending a clear and objective presentation of the results, including illustrations and statistical tables, together with the consideration of observations from other authors to enrich the discussion.

Keywords: object tracking; computer vision; CSRT; KCF; medianflow.

1 INTRODUÇÃO

Com o avanço da tecnologia, estabeleceu-se um componente intrínseco da vida cotidiana, acarretando uma série de vantagens e promovendo uma reconfiguração dos procedimentos diários. Essa influência abarca a substituição da percepção visual humana por sistemas mecânicos em diversos âmbitos, abrangendo desde a segurança empresarial e residencial até o contexto esportivo, o gerenciamento do tráfego e as operações industriais, entre outros. Para a máquina, a tarefa de analisar registros de imagens e vídeos, a fim de extrair informações de relevância, desvela-se como um desafio de magnitude considerável.

Segundo Szeliski (2022), o desafio de capacitar algoritmos computacionais a perceberem estruturas tridimensionais, semelhantes à capacidade humana, identificando objetos diversos, compreendendo suas características e determinando sua relevância dentro do contexto observado, incluindo a habilidade de reconhecer outros indivíduos, atribuir-lhes nomes e interpretar expressões faciais para inferir emoções, apresenta-se como uma tarefa computacional de elevada complexidade. Além disso, o autor ressalta que os psicólogos perceptuais continuam a enfrentar dificuldades na busca pela elucidação dos princípios subjacentes à visão humana, bem como na proposição de uma solução abrangente para esse enigma.

A visão computacional é uma subárea da inteligência artificial, que como definido por Russell e Norvig (2013), são sistemas computacionais que agem como seres humanos ou apresentam comportamento similar, possuindo algumas capacidades específicas como o processamento e a representação do conhecimento; argumentação automatizada, e aprendizado de máquina, adaptando-se a novas circunstâncias. Lugger (2013) define a inteligência artificial como o ramo da ciência da computação que se ocupa da automação do comportamento inteligente. Assim a visão computacional é um dos componentes essenciais para a construção de sistemas inteligentes.

Szeliski (2022) aborda a visão computacional como um campo interdisciplinar da ciência da computação que se concentra no desenvolvimento de algoritmos e sistemas para permitir que computadores "vejam" e interpretem informações visuais a partir de imagens ou vídeos. Ela envolve a análise, processamento e compreensão de dados visuais, permitindo a detecção de objetos, reconhecimento de padrões, medição de distâncias, rastreamento de movimentos e uma variedade de outras tarefas relacionadas à interpretação de dados visuais por sistemas computacionais. A visão computacional tem aplicações em diversos campos, incluindo robótica, medicina, segurança, automação industrial e realidade aumentada, entre outros.

Para realizar a análise de imagens e vídeos de forma confiável e com desempenho satisfatório, emergiram os algoritmos de rastreamento e detecção de objetos. Estes algoritmos se baseiam em técnicas provenientes do campo da visão computacional, cujo propósito consiste em identificar e localizar objetos específicos em imagens ou vídeos. Contudo, não se pode simplesmente escolher um algoritmo qualquer sem levar em conta o contexto de utilização, a qualidade das imagens, iluminação, escala, entre outros. A detecção de objetos, por exemplo, representa uma abordagem que tem a capacidade de reconhecer todos os objetos-alvo presentes

na imagem, bem como determinar suas categorias e informações de localização, contribuindo para o aprimoramento da capacidade de compreensão visual das máquinas.

Os algoritmos de rastreamento de objetos, por outro lado, são usados para rastrear o movimento de um objeto específico em uma sequência de imagens ou vídeos. Juntos, esses algoritmos fornecem poderosas ferramentas para analisar e reter informações, melhorando a eficiência e a precisão na análise de dados de vídeo.

Nestas abordagens de detecção ou rastreamento de objetos, a abordagem a ser aplicada deve ser a utilização de algoritmos que nos permitam construir detectores de propósito específico, a fim de maximizar a detecção ou rastreamento que se deseja; a utilização de modelos genéricos estaria propensa a lentidão de processamento e a erros de detecção (Szeliski, 2022).

Este trabalho visa estudar a aplicação de algoritmos para detecção e rastreamento de pessoas em vídeos, demonstrando o desempenho de cada um e, ao mesmo tempo, discutir os desafios e oportunidades futuras.

2 REFERENCIAL TEÓRICO

Esta seção do trabalho visa demonstrar conceitos dos algoritmos de rastreamento estudados nesta pesquisa. Todos os modelos estudados foram aplicados a partir da biblioteca *Open Source Computer Vision Library* (OpenCV, 2023). A OpenCV é uma biblioteca de código aberto que oferece uma plataforma comum para aplicações de visão computacional e acelera a adoção de percepção de máquina em produtos comerciais. Criada com o objetivo de otimizar e facilitar o uso de algoritmos de visão computacional e aprendizado de máquina, a biblioteca possui mais de 2.500 algoritmos otimizados.

Esses algoritmos abrangem um conjunto completo de técnicas tradicionais e modernas de visão computacional e aprendizado de máquina. Eles são úteis para uma variedade de tarefas, como detecção e reconhecimento de faces, identificação de objetos, classificação de ações humanas em vídeos, rastreamento de movimentos de câmeras e objetos em movimento, produção de modelos 3D de objetos, geração de nuvens de pontos 3D a partir de câmeras estéreo, entre outras (OpenCV, 2023).

Existem muitos algoritmos de rastreamento disponíveis para serem usados em qualquer situação. É necessário, ainda assim, escolher bem qual utilizar, levando em conta a situação problema para a qual se deseja uma solução computacional, utilizando das vantagens específicas do algoritmo para aprimorar a entrega correta do resultado esperado.

A abordagem do algoritmo de rastreamento *Boosting*, baseada no classificador AdaBoost, é uma técnica valiosa no domínio da visão computacional. Este método requer treinamento em tempo de execução, empregando exemplos positivos e negativos para distinguir e rastrear objetos de interesse. A caixa de delimitação inicial, geralmente fornecida pelo usuário ou por outros algoritmos de detecção de objetos, é tratada como um exemplo positivo do objeto em questão, enquanto áreas fora da caixa são consideradas fundo. Quando um novo quadro de imagem é adquirido, o modelo computa pontuações para os pixels na vizinhança da posição anterior do objeto, com a nova posição sendo estabelecida no local onde a pontuação atinge seu valor máximo. No entanto, é importante notar que o algoritmo *Boosting* é considerado datado, com desempenho mediano, uma vez que não é capaz de identificar de forma confiável situações em que o rastreamento tenha falhado. Portanto, apesar de sua utilidade, a comunidade de pesquisa tem buscado alternativas mais eficazes e avançadas para aprimorar o rastreamento de objetos em cenários de visão computacional (Szeliski, 2022; Grabner; Grabner; Bischof, 2006).

O método de rastreamento *Multiple Instance Learning* (MIL) representa uma abordagem inovadora que redefine a noção de exemplos positivos e negativos em contextos de rastreamento de objetos. Em vez de especificar exemplos positivos e negativos, o MIL opera

com "bolsas" positivas e negativas, onde a bolsa positiva contém uma coleção de imagens, não necessariamente todas positivas, mas com a exigência de que pelo menos uma imagem na bolsa seja um exemplo positivo do objeto em questão. Isso permite que o rastreador explore uma vizinhança da posição atual do objeto, incluindo múltiplos *patches*, ampliando a capacidade de manter um rastreamento preciso, mesmo quando a posição do objeto varia. O rastreador MIL oferece um desempenho satisfatório, com menor suscetibilidade a desvios em comparação com outras abordagens, tornando-se uma opção valiosa para o rastreamento de objetos em cenários de visão computacional. No entanto, é importante observar que ele não relata de maneira confiável falhas no rastreamento e não é eficaz na recuperação de objetos completamente ocultos. Portanto, o MIL é uma alternativa comprovada e eficaz para o rastreamento de objetos, particularmente em ambientes que não requerem tolerância a oclusões completas (Babenko; Yang; Belongie, 2009).

O algoritmo MedianFlow é amplamente adotado devido à sua eficiência e desempenho sólido no rastreamento de objetos que exibem mudanças de direção rápidas. Essa abordagem rastreia os pontos de interesse ao longo de um período de tempo e emprega a mediana dos movimentos rastreados para estimar o deslocamento do objeto. Quando o movimento do objeto se afasta significativamente da mediana, o algoritmo identifica isso como uma falha no rastreamento. O MedianFlow destaca-se em situações em que a previsibilidade do movimento é desafiadora e oferece uma solução valiosa para o rastreamento eficaz de objetos em cenários de visão computacional (Kalal; Mikolajczyk; Matas, 2010; Mallick, 2017).

O algoritmo *Minimum Output Sum of Squared Error* (MOSSE) emprega correlação adaptativa para o rastreamento de objetos, resultando na criação de filtros de correlação estáveis a partir de uma única imagem inicial. O MOSSE demonstra robustez diante de variações de iluminação, escala, pose e deformações não rígidas, além de detectar oclusões com base na relação pico-lóbulo lateral, permitindo pausar e retomar o rastreamento quando o objeto reaparece. Esse rastreador opera em alta taxa de quadros por segundo, com um desempenho semelhante aos rastreadores mais complexos, porém com implementação mais simples e maior velocidade. No entanto, em comparação com os rastreadores baseados em aprendizado profundo, o MOSSE possui desempenho inferior (Bolme *et al.*, 2010).

O TLD, cuja sigla se refere a *Tracking, Learning, e Detection* (Rastreamento, Aprendizado e Detecção), é um algoritmo de rastreamento que aborda a tarefa de acompanhamento de objetos a longo prazo, dividindo-a em três componentes distintos: rastreamento de curto prazo, aprendizado e detecção. Esse rastreador segue o objeto de quadro a quadro, enquanto o detector localiza todas as aparências anteriormente observadas, corrigindo o rastreamento conforme necessário (Kalal; Mikolajczyk; Matas, 2012).

Além disso, o componente de aprendizado estima os erros do detector e os atualiza para evitar esses erros no futuro. Embora o TLD possa apresentar alguma oscilação em sua saída, ele se destaca ao rastrear objetos em diversas escalas, movimentos e situações de ocultação, sendo particularmente útil em cenários em que o objeto alvo pode ficar temporariamente obstruído por outros elementos da cena. No entanto, é importante observar que esse rastreador tende a gerar um número considerável de falsos positivos, o que pode impactar sua utilidade em algumas situações de rastreamento (Kalal; Mikolajczyk; Matas, 2012).

O rastreador *Kernelized Correlation Filters* (KCF) é uma evolução que se baseia em conceitos apresentados em rastreadores anteriores. O diferencial do KCF reside na exploração de regiões de sobreposição nas múltiplas amostras positivas, como as utilizadas no rastreador MIL (Multiple Instance Learning). Essa sobreposição de dados proporciona propriedades matemáticas vantajosas que este rastreador aproveita para otimizar tanto a velocidade quanto a precisão do rastreamento. Uma característica notável do KCF é sua habilidade de relatar falhas no rastreamento de forma mais eficiente em comparação com os rastreadores BOOSTING e MIL. Portanto, quando se utiliza o OpenCV 3.1 ou versões posteriores, o KCF é uma escolha

recomendada para a maioria das aplicações de rastreamento. No entanto, é importante observar que o KCF não é capaz de se recuperar de situações de ocultação total, o que deve ser considerado ao escolher a técnica de rastreamento mais adequada para cenários específicos (Henriques *et al.*, 2014).

O algoritmo de rastreamento CSRT aplica o Filtro de Correlação Discriminativa com Confiabilidade Espacial e de Canal (DCF-CSR), que apresenta uma abordagem inovadora no rastreamento de objetos ao incorporar o mapa de confiabilidade espacial para ajustar o suporte do filtro à parte específica da região selecionada no quadro de rastreamento. Esse método garante uma ampliação e localização precisas da região de interesse, tornando-o particularmente eficaz no rastreamento de objetos com formatos não retangulares (Lukezic *et al.*, 2019).

O DCF-CSR se destaca por sua simplicidade, fazendo uso apenas de duas características padrão, HoGs e Colornames, para extrair informações relevantes. Apesar de operar a uma taxa de quadros por segundo comparativamente mais baixa, a precisão que oferece no rastreamento de objetos é notavelmente superior. Essa abordagem representa uma opção promissora para cenários nos quais a localização precisa e a rastreabilidade de objetos com formas variadas são fundamentais, mesmo que à custa de uma taxa de quadros reduzida (Lukezic *et al.*, 2019).

3 METODOLOGIA

A metodologia empregada neste estudo tem como objetivo aprofundar e aprimorar o conhecimento no campo do rastreamento e detecção de objetos, em resposta aos desafios apresentados na introdução, com foco especial na análise de dados de vídeo. Inicialmente, definimos o rastreamento de objetos como a tarefa essencial de monitorar em tempo real a trajetória de um objeto. Abordamos essa tarefa por meio de diversas técnicas que preveem a posição subsequente do objeto no próximo quadro, avaliando suas discrepâncias, vantagens e desvantagens. Nosso propósito é evidenciar as abordagens de cada um dos modelos estudados por meio de experimentação e estudos de caso.

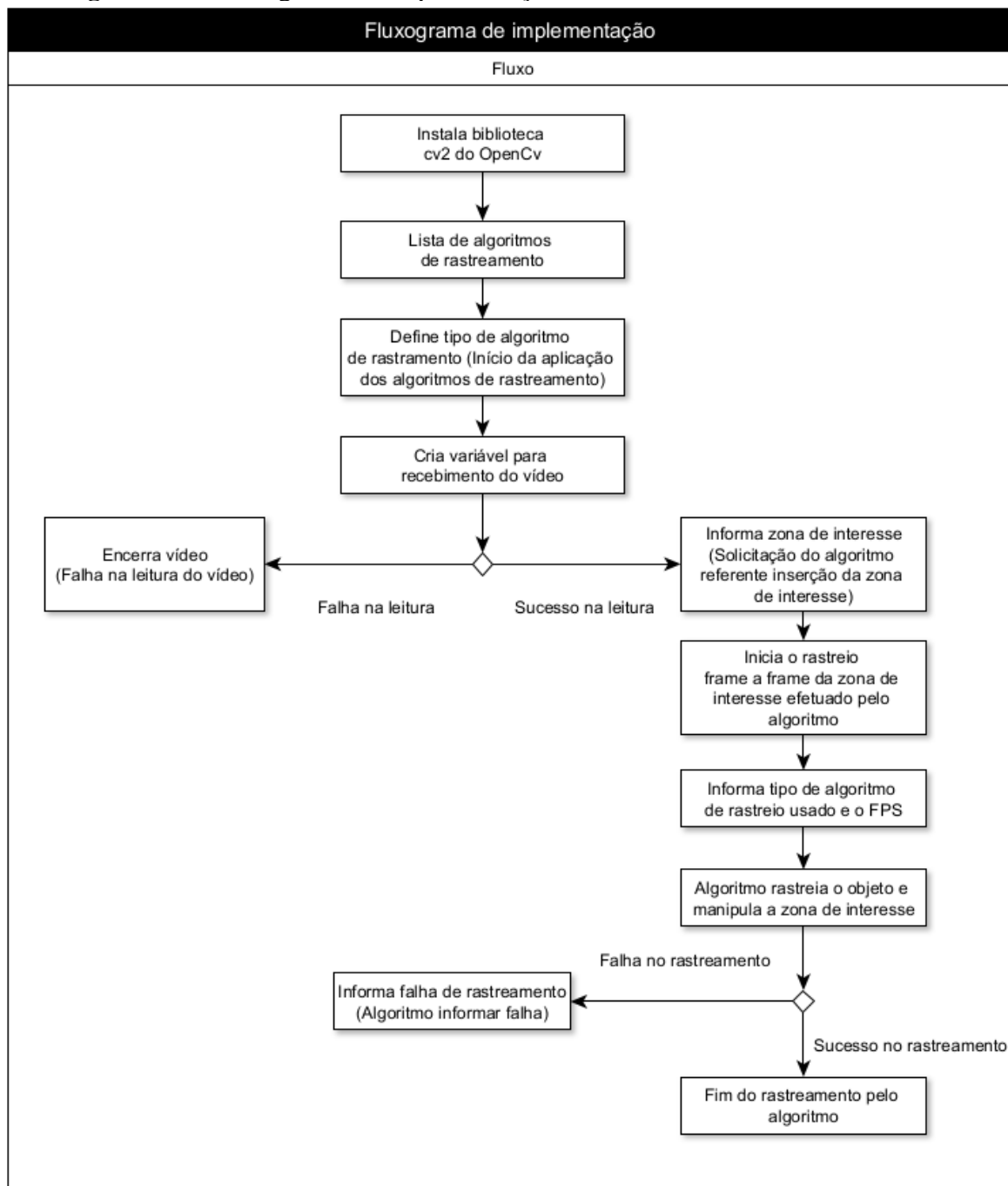
Para a implementação da metodologia, optou-se pela linguagem de programação Python e utilizou-se a biblioteca OpenCV (versão 2023) para o processamento de imagens e vídeos. Para conduzir os testes, foram empregados vídeos capturados em ambientes diversos, englobando condições de iluminação variadas, desde ambientes com pouca luz até ambientes muito iluminados. Para a análise, foram escolhidos três vídeos¹, todos registrando atividades de pessoas em movimento. É importante ressaltar que um dos vídeos foi gravado ao ar livre durante a noite, apresentando uma oclusão na qual um indivíduo passa por trás de um poste durante a gravação. A inclusão desses cenários diversos proporciona uma avaliação abrangente da eficácia dos algoritmos de rastreamento e detecção de objetos.

Na análise dos algoritmos estudados neste trabalho, é necessário considerar que um algoritmo de rastreamento eficaz é caracterizado pela incorporação de uma ampla gama de informações, tais como a aparência do objeto, sua posição atual, trajetória e outras características relevantes. Além disso, é importante destacar que é amplamente reconhecido que a caixa delimitadora pode acumular imprecisões ao longo do tempo, o que resulta em perdas de acurácia no rastreamento.

No Fluxograma 1 pode-se compreender o fluxo de implementação do modelo proposto neste trabalho, onde demonstra-se todas as etapas de pré-processamento até o resultado da aplicação das abordagens de algoritmos propostos.

¹ Link para acesso aos vídeos: <https://11nk.dev/CLhyh>

Fluxograma 1 – Fluxograma de implementação



Fonte: Elaborado pelos autores.

Foram aplicados testes nos modelos Boosting (Szeliski, 2022; Grabner; Grabner; Bischof, 2006), MIL (Babenko; Yang; Belongie, 2009), KCF (Henriques *et al.*, 2014), TLD (Kalal; Mikolajczyk; Matas, 2012), MedianFlow (Kalal; Mikolajczyk; Matas, 2010; Mallick, 2017), MOSSE (Bolme *et al.*, 2010), CSRT (Lukezic *et al.*, 2019), onde cada um dos modelos foi atribuído e um vetor e foi testado individualmente em cada um dos vídeos estudados. Os modelos foram utilizados a partir da biblioteca OPENCV.

Para a execução dos testes, cada um dos vídeos foi lido individualmente e aplicou-se cada um dos algoritmos. Quando do sucesso da leitura do vídeo, o usuário destacou a zona de interesse que define o objeto a ser rastreado. Neste contexto, “frame” representa o primeiro

quadro do vídeo, enquanto “bbox” corresponde à região de interesse (ROI), então o rastreador é inicializado, fornecendo como parâmetros o quadro do vídeo e a zona de interesse.

Informa-se ainda como parâmetros a taxa de quadros por segundo, com o objetivo de comparar a performance entre os diferentes algoritmos. Todos os modelos foram instanciados a partir de sua implementação na biblioteca OpenCV (2023) com as configurações padrão propostas pela documentação.

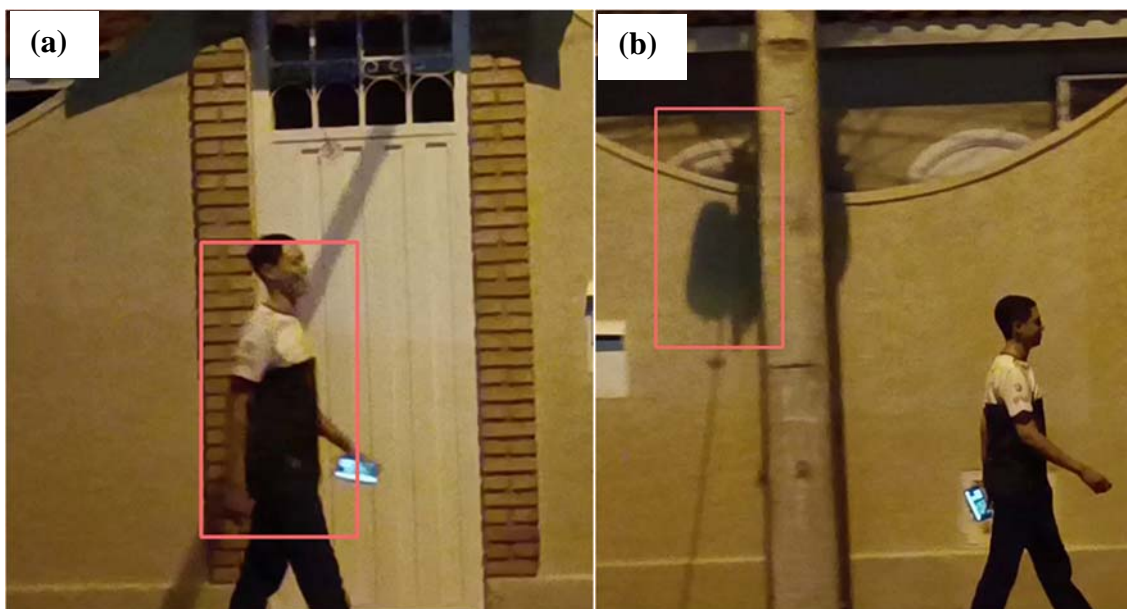
Dentro de um laço "while", o vídeo é reproduzido, e o rastreador acompanha o objeto, ajustando continuamente a posição da região de interesse. A caixa delimitadora (ROI) segue o objeto selecionado, cessando somente ao término do vídeo ou na eventualidade de perda de vista do objeto. Implementou-se um tratamento para o cenário de perda do objeto em cena, com a aplicação notificando sobre a ocorrência de uma falha no rastreamento. Ao pressionar a tecla "ESC" ou aguardar o fim do vídeo, o programa é encerrado.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Para análise dos resultados é apresentado o comportamento de cada um dos algoritmos aplicados ao vídeo, no qual uma pessoa caminha em linha reta sob condições noturnas. A baixa iluminação torna o rastreamento parcialmente desafiador. Em uma etapa do vídeo, um objeto obstrui completamente a pessoa, um cenário intencionalmente planejado para avaliar a resposta dos algoritmos quando enfrentam uma oclusão total do objeto alvo.

De acordo com Lukezic *et al.* (2019), apesar do algoritmo CSRT operar a uma taxa de quadros por segundo comparativamente mais lenta em relação aos demais algoritmos, a precisão que oferece no rastreamento de objetos é notavelmente superior.

Figura 1 – Abordagem de Identificação com o algoritmo CSRT: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

O algoritmo CSRT (Figura 1) realizou um rastreamento eficaz até o objeto sofrer uma oclusão. Posteriormente, identificou uma região de fundo que apresentava uma pequena semelhança com o objeto alvo, com uma cor preta predominante e tons de branco ligeiramente amarelados devido a iluminação do local. O algoritmo manteve-se funcional, aplicando a região

de interesse a uma região que exibia uma pequena semelhança com a que o usuário havia selecionado inicialmente.

O vídeo foi então apresentado ao rastreador MOSSE (Figura 2), que obteve um desempenho eficaz no rastreamento. Diferentemente do CSRT, quando o objeto foi ocultado, o algoritmo MOSSE manteve a região de interesse no último ponto conhecido do objeto. Em determinados cenários, essa abordagem pode ser vantajosa, pois mesmo que o alvo tenha sido perdido, ainda é possível saber a última região por onde o objeto passou.

Figura 2 – Abordagem de Identificação com o algoritmo MOSSE: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborado pelos autores.

Ao aplicar o algoritmo MedianFlow (Figura 3), desde o início do rastreamento, a região de interesse sofreu alterações no tamanho, embora o alvo tenha mantido suas proporções praticamente constantes. Além disso, o rastreamento, mesmo sem falhas imediatas, desviou-se para áreas sem justificativa aparente, evidenciando uma precisão insatisfatória nesses tipos de ambientes. Com base nos resultados obtidos, pode-se inferir que o algoritmo MedianFlow tem dificuldades para manter a precisão do rastreamento em ambientes com iluminação noturna e condições de luz precárias.

Figura 3 – Abordagem de Identificação com o algoritmo MedianFlow: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

O algoritmo TLD (Figura 4) em parte funciona identificando o objeto com base em sua aparência previamente analisada. Em um determinado ponto, o algoritmo mudou seu foco para uma área que exibia uma correlação com o objeto inicialmente selecionado. Isso pode ser interpretado como um falso positivo - uma situação na qual o algoritmo continuou a persistir. Uma análise mais detalhada da área em foco revela que seu esquema de cores é similar ao do

objeto-alvo, com um tom marrom predominante na parte superior, branco no meio do quadro e preto na parte inferior.

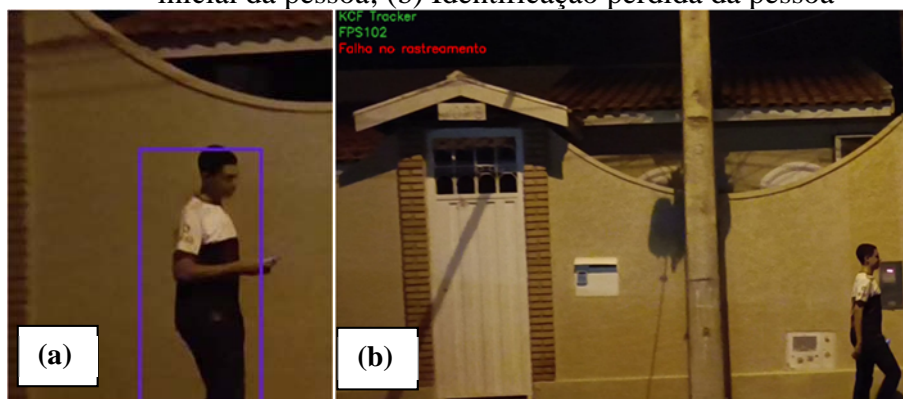
Figura 4 – Abordagem de Identificação com o algoritmo TLD: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

O algoritmo KCF (Figura 5) rastreou o objeto com eficácia até a oclusão total. Embora a falha no rastreamento seja indesejada, pode ser útil em determinados contextos. Por exemplo, pode-se determinar quanto tempo um objeto esteve em cena antes de ser ocluído. Quando o KCF falha, sabemos que o objeto foi perdido, informação que pode ser conveniente dependendo da situação.

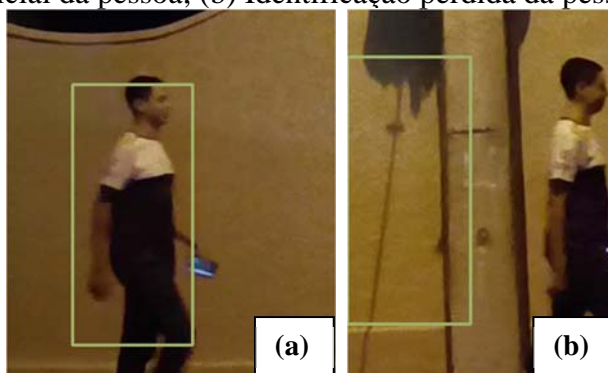
Figura 5 – Abordagem de Identificação com o algoritmo KCF: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

O algoritmo MIL (Figura 6), ao ser utilizado, apresentou resultados similares aos de outros algoritmos previamente analisados. Ele rastreou eficientemente o objeto com uma taxa de quadros satisfatória, marcando a última posição conhecida quando perdeu o alvo. Esta observação confirma que o algoritmo MIL não gerencia adequadamente situações de oclusão total do objeto.

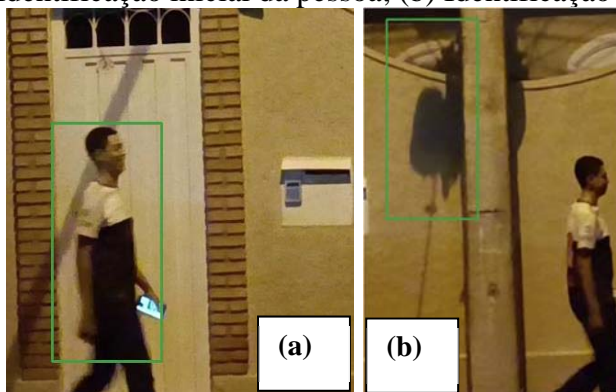
Figura 6 – Abordagem de Identificação com o algoritmo MIL: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

O algoritmo Boosting (Figura 7), conhecido por seu rápido desempenho, demonstrou resultados impressionantes no rastreamento do objeto. Mesmo em situações em que o objeto se perdeu, o algoritmo conseguiu localizar um ponto semelhante e manter-se nele. Esta habilidade notável também foi observada no algoritmo CSRT, que, apesar de ser mais pesado e utilizar técnicas de rastreamento mais avançadas, apresentou um desempenho similar. Estes resultados enfatizam a crucial importância de realizar testes rigorosos nos algoritmos antes de colocá-los em uso. Através desses testes, pode-se assegurar que o algoritmo escolhido é o mais adequado para a tarefa em questão, otimizando assim o desempenho e a eficácia do sistema.

Figura 7 – Abordagem de Identificação com o algoritmo Boosting: (a) Identificação inicial da pessoa; (b) Identificação perdida da pessoa



Fonte: Elaborada pelos autores.

Uma maneira de otimizar o desempenho desses algoritmos é a combinação de um algoritmo de detecção com um de rastreamento. Desta forma, quando o objeto de interesse reaparece, o algoritmo de detecção pode capturar a região relevante e fornecê-la ao algoritmo de rastreamento. No entanto, é crucial realizar uma análise cuidadosa para determinar qual algoritmo de detecção é mais adequado. A escolha do algoritmo de detecção deve ser feita de tal forma que maximize a eficácia e a performance geral do sistema, combinando eficientemente os algoritmos de rastreamento e detecção.

Salienta-se que, de modo geral, os algoritmos de rastreamento evidenciam um desempenho superior em comparação aos algoritmos de detecção puramente implementados. Esta superioridade é atribuída à capacidade dos algoritmos de rastreamento em reter informações sobre a aparência do objeto, sua posição anterior, direção e velocidade. Por outro lado, os algoritmos de detecção, ao reiniciarem a cada quadro, enfrentam o desafio de lidar com variações de escala em imagens.

No Quadro 1 pode-se verificar as observações em relação ao rastreamento executado por cada um dos algoritmos estudados quando expostos às amostras de vídeo estudadas neste trabalho.

Quadro 1 – Descrição resumida do resultado de cada algoritmo de rastreamento nos respectivos vídeos

Algoritmos e vídeos	Amostra de pessoa se aproximando com foco na pessoa e oclusão parcial	Amostra de pessoa se aproximando com foco em um objeto	Amostra de uma pessoa andando com oclusão total
CSRT	Rastrou o alvo até o fim, adaptando a região de interesse corretamente.	Obteve êxito ao rastrear. Gerou uma falha de rastreamento esperada apenas no fim do vídeo, onde o objeto sai completamente de cena.	Realizou um rastreamento eficaz até o objeto ser perdido de vista. Após, focou em um ponto incorreto.
MOSSE	Gerou uma falha no rastreamento logo no começo, por fim focou em uma parte errada.	Não conseguiu rastrear o objeto, provando ser uma escolha não boa para quando o alvo muda sua escala.	Conseguiu rastrear bem o objeto até a oclusão. Por fim, a região de interesse ficou localizada no último ponto de conhecimento do algoritmo.
MedianFlow	Acompanhou corretamente o alvo até o fim do vídeo, com uma precisão muito satisfatória.	Teve um bom resultado, modificou a região de interesse devido a mudança de escala do alvo.	Teve dificuldades para manter a precisão do rastreamento, alterando muito a região de interesse e perdendo rapidamente o objeto.
TLD	Rodou muito lentamente e na maioria do tempo não estava rastreando o alvo selecionado.	Logo no começo já perdeu o objeto de vista. Não gerou uma falha, mas o seu resultado não teve valor nenhum.	Não apresentou falha no rastreamento. Contudo, no começo, já mudou o foco do objeto selecionado para uma região não esperada e nela permaneceu.
KCF	Rodou com uma excelente performance. Contudo, durante uma oclusão parcial, perdeu o alvo e gerou uma falha de rastreamento.	Foi perceptível que a falha no rastreamento aconteceu assim que o objeto mudou rapidamente de posição, no começo do vídeo.	Rastrou com boa performance até a oclusão total. Esse algoritmo demonstrou ser uma boa escolha para essas condições.
MIL	Manteve o rastreio muito bem até a oclusão parcial. Após, capturou uma pequena parte do alvo selecionado, diminuindo a região de interesse.	Ao decorrer do vídeo o rastreador muda a região de interesse aos poucos, de tal modo que no final já se encontrava distante do que foi selecionado no começo.	Apresentou um bom rastreamento junto de uma boa taxa de quadros enquanto o objeto era visível.
Boosting	Conseguiu um bom rastreio, porém ficou perceptível a dificuldade em lidar com a alteração da escala.	Lidou bem com uma mudança rápida de posição. Ao decorrer perdeu o objeto e rastrou uma parte não selecionada. Manteve o vídeo numa taxa de quadros interessante.	Lidou muito bem com as condições em que o vídeo se encontra, unindo uma boa performance com um rastreamento preciso e confiável, apesar de perder o objeto após oclusão total.

Fonte: Elaborado pelos autores.

5 CONSIDERAÇÕES FINAIS

O manuseio de algoritmos de rastreamento de objetos se revela mais complexo do que pode parecer à primeira vista. Não se trata de uma simples escolha de ferramenta e implementação, mas sim de um processo que exige uma abordagem cuidadosa. Como

evidenciado, os algoritmos apresentam variações significativas em seus modos de funcionamento, cada um com suas próprias peculiaridades, que os tornam mais adequados para determinadas situações e menos eficazes em outras.

A atividade de rastrear um objeto em sua essência é a capacidade de acompanhar um objeto previamente selecionado, evitando que ele se perca ou que o foco seja desviado para outros pontos. Portanto, espera-se que um algoritmo que se propõe exclusivamente a essa tarefa que seja confiável e preciso. O algoritmo *Channel and Spatial Reliability Tracking* (CSRT) mostrou-se, de acordo com os testes apresentados e percepção humana, o melhor algoritmo de rastreamento.

Uma situação desfavorável se manifesta ao implementar exclusivamente um algoritmo de rastreamento sem um estudo e análise devida. A perda de rastreamento do objeto torna-se uma questão de tempo, mesmo para os algoritmos mais robustos, como o CSRT. Afinal, esses algoritmos são projetados para rastrear objetos, porém não são imunes às inúmeras dificuldades que podem surgir, como oclusões totais ou parciais do alvo. Pode-se pensar que para contornar estes obstáculos basta utilizar um algoritmo de detecção. Contudo, simplesmente incorporar um algoritmo de detecção não garante resultados satisfatórios; é necessário treinar o classificador adequadamente para se adaptar às condições específicas presentes no vídeo ou na imagem em questão, exigindo ainda mais uma série de análises e uma boa compreensão do seu funcionamento.

A análise dos resultados de cada um dos modelos foi observacional. Deve-se buscar como complemento do trabalho formas de sintetizar a acurácia obtida na detecção do objeto rastreado no vídeo frame-a-frame.

É notório que a manipulação de algoritmos de rastreamento de objetos é uma tarefa complexa e repleta de desafios. Isso demanda um profundo entendimento das ferramentas disponíveis e uma aplicação cuidadosa. A seleção apropriada e a execução precisa desses algoritmos desempenham um papel vital no êxito de qualquer aplicação de visão computacional.

REFERÊNCIAS

BABENKO, B.; YANG, M. H.; BELONGIE, S. Visual tracking with online multiple instance learning. *In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, 2009, Miami. **Proceedings** [...]. Miami: IEEE, 2009. p. 983-990. Disponível em: <https://faculty.ucmerced.edu/mhyang/papers/cvpr09a.pdf>. Acesso em: 3 nov. 2023.

BOLME, D. S. *et al.* Visual object tracking using adaptive correlation filters. *In: COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*, 2010, San Francisco. **Proceedings** [...]. San Francisco: IEEE, 2010. p. 2544-2550.

GRABNER, H.; GRABNER, M.; BISCHOF, H. Real-time tracking via on-line boosting. *In: BRITISH MACHINE VISION CONFERENCE – BMVC*, 5., 2006, Edinburgh. **Proceedings** [...]. Edinburgh, 2006. p. 47-56.

HENRIQUES, J. F. *et al.* High-speed tracking with kernelized correlation filters. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 37, n. 3, p. 583-596, 2014.

KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Forward-backward error: automatic detection of tracking failures. *In: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION*, 20th, 2010, Istanbul. **Proceedings** [...]. Istanbul, 2010. p. 2756-2759.

KALAL, Z; MIKOLAJCZYK, K.; MATAS, J. Tracking-learning-detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 34, n. 7, p. 1409-1422, 2012.

LUGGER, G. F. **Inteligência artificial**. 6. ed. São Paulo: Pearson, 2013.

LUKEZIC, A. *et al.* Discriminative correlation filter with channel and spatial reliability. **Computer Vision and Pattern Recognition**, v. 3, jan. 2019. Disponível em: <https://arxiv.org/pdf/1611.08461.pdf>. Acesso em: 30 out. 2023.

MALLICK, S. **Object tracking using OpenCV: C++/Python**. 2017. Disponível em: <https://learnopencv.com/object-tracking-using-opencv-cpp-python/>. Acesso em: 30 out. 2023.

OPEN SOURCE COMPUTER VISION LIBRARY – OPENCV. **OpenCV**. Disponível em: <https://docs.opencv.org/4.7.0/d0/de3/citelist.html>. Acesso em: 29 out. 2023.

RUSSELL, S. J.; NORVIG, P. **Inteligência artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

SZELISKI, R. **Computer vision: algorithms and applications**. 2. ed. New York: Springer, 2022. Disponível em: <https://szeliski.org/Book/>. Acesso em: 3 nov. 2023.