



Ramicés Moisés do Nascimento

Avaliação de algoritmos de rastreamento no problema de detecção de pessoas no mar

Recife

2023

Ramicés Moisés do Nascimento

Avaliação de algoritmos de rastreamento no problema de detecção de pessoas no mar

Monografia apresentada ao Curso de Bacharelado em Ciências da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Ciência da Computação

Orientador: Valmir Macário Filho

Recife

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- R173a Nascimento, Ramicés
Avaliação de algoritmos de rastreamento no problema de detecção de pessoas no mar / Ramicés Nascimento. - 2023.
52 f. : il.
- Orientador: Valmir Macario Filho.
Inclui referências.
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco,
Bacharelado em Ciência da Computação, Recife, 2023.
1. algoritmo. 2. rastreamento. 3. genético. 4. avaliação. I. Filho, Valmir Macario, orient. II. Título

CDD 004



**MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Ramicés Moisés do Nascimento às 14h do dia 13 de setembro de 2023, no link <https://meet.google.com/avm-aeve-wic>, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado “Avaliação de algoritmos de rastreamento no problema de detecção de pessoas no mar”, orientado por Valmir Macario Filho e aprovado pela seguinte banca examinadora:

Valmir Macario Filho

João Paulo Silva do Monte Lima
DC/UFRPE

À minha mãe, meu pai e minha avó, que tanto me ajudaram nessa jornada. À minha mulher que me incentivou a terminar minha monografia e ao meu orientador, que teve uma paciência gigantesca comigo

Agradecimentos

Agradeço ao meu orientador, que muito insistiu pra que eu finalizasse essa jornada. Agradeço a minha mãe, meu pai e minha avó, que muito lutaram e se esforçaram para que eu chegasse até aqui e agradeço a minha mulher por ameaçar não casar caso eu não terminasse a monografia.

*“É terra de meu país; desejo que seja posta em meu caixão, se eu morrer longe de
minha pátria”
(Dom Pedro II)*

Resumo

Sabe-se que nas praias de Pernambuco, ataques de tubarões são um medo constante da população. Muitos desses ataques são fatais, o que faz necessário que alguma atitude seja tomada, visto que as praias de Pernambuco atraem milhares de turistas por ano. Assim, pesquisadores da UFRPE iniciaram uma pesquisa visando desenvolver um sistema de rastreamento de pessoas no mar, que torna possível avisar aos salvavidas quando as pessoas ultrapassarem uma área considerada segura na praia, além de poder alocar uma quantidade maior desses profissionais em lugares com maior número de pessoas.

O sistema foi dividido em três etapas: segmentação da imagem, detecção dos banhistas e rastreamento dos mesmos. Este trabalho foca na terceira etapa. Rastrear pessoas é uma tarefa complexa e com custo computacional alto. Problemas com mudanças de iluminação do ambiente, mudança na direção dos alvos e mudanças no plano de fundo são apenas algumas das dificuldades que podem ser citadas. Assim, o trabalho desenvolvido tem como objetivo avaliar seis algoritmos de rastreamento de pessoas presentes na literatura em imagens de praia.

Primeiro, rotulou-se manualmente uma base de dados de dez vídeos gravados na praia de Boa Viagem, em Pernambuco. Depois foram escolhidos seis algoritmos para serem avaliados. Depois, foi comparado o resultado de cada quadro dado como resposta pelo algoritmo, com o que tinha sido anteriormente rotulado e tirou-se uma média. Então, tirou-se uma média geral para saber a taxa de acerto do algoritmo e o tempo de execução do mesmo. Por fim, o melhor algoritmo foi escolhido para otimização através de um algoritmo genético, e verificou-se se houve alguma melhora no resultado. O CSRT foi o algoritmo que obteve o melhor resultado e após a otimização com o algoritmo genético, obteve-se uma melhoria de 20% na sua acurácia

Palavras-chave: algoritmo, rastreamento, genético, avaliação

Abstract

It is known that shark attacks are a constant fear for the population on the beaches of Pernambuco. Many of these attacks are fatal, which calls for some action to be taken, considering that Pernambuco beaches attract thousands of tourists each year. Therefore, researchers from UFRPE initiated a study aiming to develop a system for tracking people in the sea, which would make it possible to alert lifeguards when individuals exceed a designated safe area on the beach, as well as allocate a greater number of these professionals in areas with a higher concentration of people.

The system was divided into three stages: image segmentation, detection of beachgoers, and tracking of individuals. This work focuses on the third stage. Tracking people is a complex task with high computational costs. Problems such as changes in lighting conditions, alterations in the direction of targets, and variations in the background are just a few of the difficulties that can be mentioned. Thus, the objective of this research is to evaluate six people tracking algorithms found in the literature using beach images.

Firstly, a database of ten videos recorded at Boa Viagem beach in Pernambuco was manually labeled. Then, six algorithms were selected for evaluation. Subsequently, the output of each frame provided by the algorithm was compared with the previously labeled data, and an average was calculated. Overall averages were then obtained to assess the algorithm's accuracy and execution time. Finally, the best algorithm was chosen for optimization using a genetic algorithm, and any improvements in the results were verified. CSRT was the algorithm that obtained the best result and after optimization with the genetic algorithm, an improvement of 20% in its accuracy was obtained

Keywords: algorithm, tracking, genetic, performance evaluation

Lista de ilustrações

Figura 1 – Placa de aviso de ataque de tubarões em Boa Viagem	13
Figura 2 – Exemplo de rastreamento de um dos vídeos dessa base	14
Figura 3 – Fluxograma do Boosting Tracking	26
Figura 4 – Basicamente, o MIL utiliza um conjunto de características como exemplo e quando um novo objeto é utilizado, ele é classificado como positivo ou negativo baseado nesse conjunto.	27
Figura 5 – Arquitetura do Median Flow	27
Figura 6 – Exemplo do MOSSE sendo utilizado em diversas situações, como rotação, oclusão e deslocamento	29
Figura 7 – Exemplo de pontos chaves durante a detecção de uma pessoa usando o CppMT	30
Figura 8 – A arquitetura do treinamento off-line. Na fase de treino, a rede neural aprende a relação objeto e movimento e durante a fase de testes avalia-se o resultado	32
Figura 9 – Arquitetura do GOTURN	32
Figura 10 – Aqui, é apresentada a construção da região de interesse. Da esquerda para a direita: O objeto a ser rastreado, a extração de características, como cor e forma, e o filtro que é aplicado para definir a região de busca final.	33
Figura 11 – Primeiro vídeo	36
Figura 12 – Segundo vídeo	36
Figura 13 – Terceiro vídeo	37
Figura 14 – Quarto vídeo	37
Figura 15 – Quinto vídeo	38
Figura 16 – Sexto vídeo	38
Figura 17 – Sétimo vídeo	39
Figura 18 – Oitavo vídeo	39
Figura 19 – Nono vídeo	40
Figura 20 – Décimo vídeo	40
Figura 21 – Pessoas Rotuladas	45
Figura 22 – O resultado do rastreamento. Em rosa, a marcação manual da base e em vermelho, o rastreamento do algoritmo. Nos estágios iniciais, o rastreamento é muito preciso, porém vai perdendo essa acurácia com o tempo	49

Lista de tabelas

Tabela 1 – Acurácia média de rastreamento	46
Tabela 2 – Tempo de rastreamento em minutos	47
Tabela 3 – Média de acurácia do CSRT com otimização	48

Lista de abreviaturas e siglas

CSRT	Correlation Filter with Channel and Spatial Reliability
KFC	Kernelized Correlation Filter
MIL	Multiple Instance Learning
CMIT	Comitê Estadual de Monitoramento de Incidentes com Tubarões
Caffe	Convolutional architecture for fast feature embedding
GOTURN	Generic Object Tracking Using Regression Networks
CppMT	C++ Multi-Object Tracking
MOSSE	Minimum Output Sum of Squared Errors)

Sumário

	Lista de ilustrações	9
1	INTRODUÇÃO	13
2	TRABALHOS RELACIONADOS	18
3	CONCEITOS BÁSICOS	25
3.0.1	RASTREAMENTO	25
3.1	ALGORITMOS DE RASTREAMENTO	25
3.1.1	BOOSTING TRACKING	25
3.1.2	MIL	26
3.1.3	Median Flow	27
3.1.4	Mosse	28
3.1.5	KCF	29
3.1.6	CPPMT	30
3.1.7	GOTURN	31
3.1.8	CSRT	33
3.2	Algoritmo Genético	33
4	METODOLOGIA	35
4.1	ROTULAÇÃO DA BASE DE DADOS	35
4.2	ESCOLHA DOS ALGORITMOS	41
4.3	AVALIAÇÃO DOS ALGORITMOS	41
4.4	OTIMIZAÇÃO COM O ALGORITMO GENÉTICO	42
4.5	MÁQUINA DE TESTES	44
5	RESULTADOS	45
6	CONCLUSÃO	50
	REFERÊNCIAS	52

1 Introdução

Segundo os registros oficiais do Comitê Estadual de Monitoramento de Incidentes com Tubarões (Cemit) , nos últimos 25 anos, foram registrados cerca de cento e um ataques de tubarões nas praias de Pernambuco. Destes ataques, vinte e cinco resultaram em vítimas fatais. No entanto, é importante ressaltar que esse fenômeno não é exclusivo de Pernambuco, pois ataques de tubarões também são registrados em outras áreas ao longo da costa brasileira. Contudo, Pernambuco é responsável por mais da metade dos ataques no país, o que tem levado à sua reputação internacional pela presença de tubarões em suas águas costeiras.



Figura 1 – Placa de aviso de ataque de tubarões em Boa Viagem

Dentre todas as praias do estado, destaca-se a praia de Boa Viagem, que apresenta um número significativo de ataques. Com o intuito de abordar os problemas mencionados, pesquisadores da Universidade Federal Rural de Pernambuco iniciaram um projeto financiado pela FACEPE e SDS-PE, voltado para o desenvolvimento de um sistema de detecção e rastreamento de banhistas no mar. O objetivo desse sistema é

auxiliar os salva-vidas na monitoração de banhistas em áreas consideradas de risco, além de facilitar a alocação eficiente desses profissionais. Dessa forma, áreas com maior concentração de pessoas na praia poderão contar com um maior número de salva-vidas em comparação a áreas menos frequentadas.

O sistema é dividido em três partes principais: segmentação, detecção de banhistas e rastreamento. A etapa de segmentação consiste em separar a área da areia do mar e dividir a imagem em regiões de interesse. Dessa forma, o processamento posterior será realizado apenas na área relevante para o sistema, ou seja, nas pessoas dentro da água. A detecção de banhistas refere-se à identificação das pessoas que estão no mar. Por fim, o rastreamento tem como objetivo acompanhar as pessoas detectadas na etapa anterior até o momento em que elas saíam da água ou saíam do alcance de visão da câmera. Este trabalho concentra-se especificamente na última etapa descrita acima.

O rastreamento é a capacidade de acompanhar um objeto em um vídeo ao longo dos quadros subsequentes. O algoritmo deve ser capaz de acompanhar o movimento dos objetos no vídeo, mesmo que eles estejam parcialmente ocultos, e parar de rastreá-los quando eles saírem do campo de visão. O rastreamento é utilizado para diversos propósitos, como vigilância e monitoramento e devem se adaptar aos mais diversos desafios (ABDULGHAFOOR, 2022).



Figura 2 – Exemplo de rastreamento de um dos vídeos dessa base

Tão importante quanto os algoritmos em si, são as métricas utilizadas para avaliá-los. Essas métricas fornecem meios de avaliar o desempenho dos algoritmos em relação ao acompanhamento de objetos em imagens subsequentes. É necessário avaliar se a forma e a posição do objeto rastreado são consistentes com sua velocidade, mesmo quando essa velocidade é nula, levando em consideração fatores como oclusões parciais. Além disso, é possível avaliar o tempo necessário para que os algoritmos concluam essa tarefa.

Durante muito tempo, métodos manuais foram utilizados para avaliação. Nesses métodos, os próprios pesquisadores observavam os resultados dos algoritmos e faziam avaliações manuais. Embora essa abordagem seja funcional, ela demanda um tempo considerável para ser concluída. O ideal é que a avaliação ocorra de forma automática. Dessa forma, o tempo que seria gasto com avaliação manual pode ser direcionado para testar os algoritmos em uma variedade mais ampla de situações.

O objetivo principal desse trabalho é avaliar os resultados obtidos no rastreamento de pessoas no mar utilizando *hardwares de baixo custo*. Para isso, foram escolhidos seis algoritmos que estão disponíveis através do *OpenCV*: o *Boosting* (GRABNER; GRABNER; BISCHOF, 2006a), o CSRT (LUKEZIC et al., 2018), o KCF (HENRIQUES et al., 2012), o MIL (BABENKO; YANG; BELONGIE, 2009), o Moose (BOLME et al., 2010) e o *Median Flow* (KALAL; MIKOLAJCZYK; MATAS, 2010). Os mesmos serão avaliados utilizando uma base de dados de vídeos que foram filmados na praia de Boa Viagem em Pernambuco, por pesquisadores e alunos da UFRPE. Essa base é composta por dez vídeos filmados em posições e horários do dia diferentes, para garantir uma maior variabilidade de dados para testes. Ela foi totalmente rotulada para ser utilizada nesse trabalho utilizando uma ferramenta chamada VATIC, sendo assim uma base nova e uma contribuição para a literatura, visto que ela pode ser usada em trabalhos futuros por outros pesquisadores.

Para avaliar os algoritmos, será comparada a resposta gerada pelos algoritmos após a execução com a resposta esperada da base de dados rotulada. A resposta é fornecida na forma de um polígono de quatro lados, assim como a rotulação da base. Para realizar essa avaliação, será utilizada a fórmula da interseção sobre a união (IOU), que consiste em calcular a razão entre a área da união e a área da interseção dos dois polígonos, a fim de determinar o grau de sobreposição entre eles. Esse valor é dado em forma de porcentagem, indicando a taxa de acurácia do resultado obtido com o que foi rotulado.

Depois disso, o tempo necessário para concluir o rastreamento em cada vídeo também será avaliado. Essa etapa é de extrema importância, uma vez que o objetivo final da aplicação é ser um sistema em tempo real. Não é útil para um salva-vidas perceber, vários minutos depois, que um banhista se deslocou para uma área mais

profunda do mar. Como os vídeos foram gravados em uma taxa de quadros muito alta (360 quadros por segundo), o que garante alta precisão devido à disponibilidade de mais informações para o processamento, mas com maior tempo de processamento, os algoritmos também foram testados pulando quadros em números específicos: trinta, sessenta e noventa. Isso foi feito para verificar se a redução no tempo de processamento compensa a perda de precisão, pois quanto mais rápido e preciso for o algoritmo, melhor será o desempenho.

Além disso, é importante observar que todos os algoritmos utilizados nos testes possuem parâmetros internos em seu código-fonte que definem seu comportamento e tempo de execução. Esses parâmetros incluem características como a área de busca (que delimita o espaço em que o algoritmo procurará um objeto rastreado) e a quantidade de características utilizadas na busca, entre outros. Como os algoritmos foram treinados para uma classe genérica de problemas, surgiu a ideia de otimizá-los para o problema específico do rastreamento no mar. Após a realização dos testes para encontrar o melhor algoritmo, o mesmo será escolhido para ter seus parâmetros otimizados por meio de um algoritmo genético. Embora outras classes de algoritmos, como redes neurais, pudessem ter sido utilizadas para tentar essa otimização, este trabalho se concentra especificamente nessa abordagem, deixando espaço para investigar outras soluções em trabalhos futuros.

O presente trabalho está organizado em diversas seções, cada uma abordando aspectos específicos relacionados ao tema proposto. A seguir, apresentamos uma breve descrição de cada seção:

Introdução: Nesta parte inicial, oferecemos uma visão panorâmica do cenário abordado no trabalho, esclarecemos o problema central que será investigado, delineamos os propósitos da pesquisa, justificamos a relevância do estudo e estabelecemos os limites do escopo a ser explorado. Adicionalmente, esmiuçamos a organização do trabalho, proporcionando uma visão antecipada das seções que virão a seguir.

Revisão da Literatura: Neste segmento, procedemos a uma análise aprofundada das teorias centrais, conceitos fundamentais e estudos prévios vinculados ao tópico do trabalho. Investigamos minuciosamente o ambiente em que o problema se encontra, evidenciando trabalhos de destaque e identificando áreas de conhecimento que carecem de informações, o que respalda a pertinência da pesquisa em questão.

Metodologia: Aqui são descritos os métodos e procedimentos utilizados para conduzir a pesquisa. São apresentados a base de dados utilizada, os algoritmos criados e a métrica análise.

Resultados: Nesta seção, são apresentados os resultados obtidos, qual o melhor algoritmo e qual o resultado a otimização com o algoritmo genético.

Conclusão: A conclusão desempenha o papel de condensar os aspectos cruciais explorados ao longo do trabalho, ao mesmo tempo em que oferece respostas aos objetivos preestabelecidos. Nesse contexto, realçam-se as valiosas contribuições do estudo para o aprofundamento do entendimento acerca do tema em análise. Com uma sessão final que explica um pouco dos possíveis trabalhos que podem ser desenvolvidos no futuro

Por fim, os resultados obtidos nesse trabalho mostram que o CSRT é o algoritmo mais eficiente dentre os escolhidos e que a otimização através do algoritmo genético gera resultados melhores, tendo espaço para trabalhos futuros com novas otimizações.

2 Trabalhos Relacionados

Para escolher boas métricas de avaliação dos algoritmos utilizados neste trabalho e obter um entendimento mais aprofundado do problema a ser resolvido, foi realizada uma pesquisa na literatura sobre métodos de rastreamento de pessoas. O objetivo era verificar as melhores técnicas de rastreamento que poderiam ser utilizadas em um *hardware* de baixo custo. A seguir será apresentado um pequeno estudo de caso com todos os trabalhos pesquisados.

No trabalho de Tao Yang, Stan Li, Quan Pan e Jing Li (YANG et al., 2004), é proposto um sistema de rastreamento em tempo real que não requer conhecimento prévio da forma ou velocidade de deslocamento do objeto no vídeo. O sistema é capaz de fornecer resultados precisos mesmo em cenas dinâmicas ou com objetos parcialmente ocluídos. O método assume que a velocidade de dois objetos durante a oclusão é a mesma para todo o grupo, permitindo recuperar as trajetórias individuais de cada objeto e determinar suas posições exatas durante e após a oclusão, desde que os objetos tenham sido corretamente rotulados anteriormente. Esse sistema é relevante para o projeto em questão, pois ambos lidam com fundos dinâmicos, possibilitando a aplicação de técnicas similares para aprimorar os algoritmos de rastreamento selecionados para avaliação no projeto. No entanto, os resultados descritos no artigo indicam que o tempo de resposta é demorado, mesmo em vídeos de baixa resolução. Além disso, não foi encontrada nenhuma implementação disponível dessa abordagem, tornando necessário realizar a implementação manualmente.

Já o trabalho de Bo Wu e Ram Nevatia(WU; NEVATIA, 2006) propõe uma abordagem que detecta e rastreia automaticamente pessoas que possivelmente estejam ocluídas e em movimento. Nessa abordagem são combinados os resultados de dois tipos de detecção: a da pessoa como um todo, e de partes da mesma. Assim, por exemplo, caso um indivíduo esteja com as pernas ocluídas, o sistema ainda será capaz de detectá-la de maneira correta. Uma das principais vantagens desse método é que ele realiza a detecção de pessoas de maneira automaticamente. Além disso, ele é capaz rastrear múltiplas pessoas ao mesmo tempo e possui bons resultados quando o objeto alvo está ocluído. Porém o algoritmo não foi testado em ambientes dinâmicos, apenas em ambientes fechados.

No trabalho de Quming Zhoua e J.K. Aggarwa (ZHOU; AGGARWAL, 2006) foi criado um algoritmo de rastreamento para ambientes externos que integra a posição espacial, forma e cor do objeto alvo. Essa integração torna o algoritmo robusto a mudança de orientação do objeto, mudanças no fundo e a interrupção no movimento.

Além disso, foi criada uma versão do algoritmo que utiliza múltiplas câmeras para realizar o rastreamento. Dessa forma, fundindo as trajetórias obtidas por elas, o sistema torna-se robusto a oclusão. Isso é obtido através do filtro de Kalman, que torna possível alternar entre as câmeras assim que o objeto é ocluído. Porém, o sistema carece de testes em ambientes extremamente dinâmicos. Além disso, o algoritmo não consegue obter um bom resultado, quando o objeto muda de direção rapidamente.

No trabalho desenvolvido por Spampinato, Palazzo e Giordano ([SPAMPINATO; PALAZZO; GIORDANO, 2011](#)), eles propõem um método de avaliação para rastreamento de objetos que combina informações de forma, aparência e movimento, utilizando um classificador Naive Bayes. O método é autoavaliativo, permitindo que o próprio algoritmo identifique e teste resultados incorretos. Os experimentos realizados no estudo utilizaram três conjuntos de dados distintos: cinco vídeos de tráfego real, cinco vídeos da base de dados CAVIAR e cinco vídeos da base de dados do projeto *Fish4Knowledge*. O objetivo principal do trabalho foi avaliar o desempenho dos algoritmos de rastreamento por meio dessa abordagem de avaliação integrada.

Jan Hosang, Rodrigo Benenson e Bernt Schiele, no artigo intitulado "*How good are detection proposals, really?*" ([HOSANG; BENENSON; SCHIELE, 2013](#)), conduziram um estudo para avaliar a eficácia das propostas de detecção em tarefas de reconhecimento de objetos. O estudo foi realizado no Instituto Max Planck de Informática, em Saarbrücken, Alemanha, no ano de 2013. O trabalho aborda a importância das propostas de detecção na detecção de objetos e apresenta uma análise comparativa de diferentes algoritmos nesse contexto. Esse trabalho foi muito importante, pois ele apresenta uma metodologia para avaliar diferentes algoritmos, semelhante ao que está sendo feito neste artigo, embora tenha sido feito em um ambiente fechado.

Já no trabalho de Wu, Lim e Yang ([WU; LIM; YANG, 2013](#)), foram feitas duas contribuições importantes para a literatura de rastreamento de objetos. Primeiramente, eles criaram uma base de dados com mais de cinquenta vídeos rotulados, bem como uma biblioteca com mais de vinte e nove algoritmos de rastreamento. Além disso, eles padronizaram as entradas e saídas dos algoritmos, facilitando sua utilização e comparação. Essa base de dados e biblioteca foram utilizadas para realizar um grande número de testes e avaliar o desempenho dos algoritmos. Esse trabalho foi relevante para a presente pesquisa, uma vez que ambos têm o objetivo de realizar testes de desempenho em algoritmos de rastreamento. No entanto, não foi possível encontrar a biblioteca mencionada contendo os algoritmos.

No trabalho de Longyin Wen et al. ([WEN et al., 2014](#)), eles propõem um método de busca nas vizinhanças, construído a partir de um grafo que é criado hierarquicamente. Propriedades como aparência e movimento também estão integradas nesse grafo. Além disso, o algoritmo foi desenvolvido para rastrear múltiplos objetos. Com-

parações com outros algoritmos do estado da arte mostram que ele possui um bom desempenho. Uma das principais vantagens é que, é possível rastrear múltiplos alvos, porém, não foi encontrada a implementação desse algoritmo na internet.

Outro trabalho que também foi útil para esta pesquisa foi o de Wu, Lim e Yang (WU; LIM; YANG, 2015). Nele, foram realizados experimentos de larga escala para testar algoritmos de rastreamento. A partir desses testes, eles chegaram a duas conclusões interessantes sobre o problema de rastreamento. A primeira conclusão é que as informações sobre o plano de fundo de um vídeo são extremamente importantes para um rastreamento eficiente. A segunda conclusão é que o modelo de movimento desempenha um papel essencial no rastreamento de objetos, especialmente quando ocorrem deslocamentos abruptos. Esse artigo foi muito relevante, pois apresenta um trabalho semelhante ao realizado nesta pesquisa, permitindo uma comparação de desempenho entre alguns algoritmos amplamente reconhecidos na literatura. No entanto, é importante destacar os testes foram feitos todos dentro de um ambiente controlado.

No artigo "*Real-time insect tracking and monitoring with computer vision and deep learning*"(BJERGE; MANN; HøYE, 2021) é abordado o uso de visão computacional e aprendizagem de máquina para o rastreamento de insetos em tempo real. Basicamente uma rede neural do tipo YOLOv3 foi treinada e acoplada em um servidor. Depois, uma câmera fica filmando e mandando imagens em tempo real para esse servidor e lá a rede treinada realiza o rastreamento e a classificação dos insetos. Os resultados do experimentos são positivos, sendo que foi possível detectar e classificar diferentes espécies de insetos, sendo uma aplicação interessante de se usar na agricultura.

O artigo "*A novel enhanced algorithm for efficient human tracking*"(GHEISARI et al., 2022) apresenta um novo algoritmo aprimorado para o rastreamento de pessoas. O algoritmo proposto utiliza técnicas avançadas de processamento de imagem para segmentação e aprendizado de máquina para identificar e rastrear com precisão as pessoas em tempo real. O método proposto mostra resultados promissores, superando os algoritmos existentes em termos de acurácia e eficiência de processamento. Além disso, também foram feitos experimentos e comparações detalhadas com outros métodos para validar a eficácia do novo algoritmo e os resultados sugerem que o algoritmo proposto é altamente eficiente e preciso, tornando-o uma solução viável para aplicações de rastreamento de pessoas em tempo real. Porém, para o seu correto funcionamento seria necessário reescrever toda a parte de segmentação que foi desenvolvida nesse projeto, visto que ele tem um algoritmo de segmentação próprio, o que foge da proposta inicial de avaliar apenas a acurácia dos algoritmos.

Já no trabalho "*A novel real-time multiple objects detection and tracking framework for different challenges*"(ABDULGHAFOOR, 2022) apresenta um novo fra-

network de detecção e rastreamento em tempo real de múltiplos objetos, capaz de rastrear em diferentes ambientes com diferentes desafios. Um grande desafio encontrado nos sistemas existentes são as variações de iluminação, oclusões, mudanças de perspectiva e movimentos rápidos, que podem comprometer a precisão e o desempenho geral. Para tentar criar uma nova ferramenta que consigo enfrentar esses desafios, o sistema utiliza uma abordagem baseado em "deep learning", combinando redes neurais convolucionais (CNNs) e técnicas de filtragem Kalman para detectar e rastrear objetos de forma eficiente. O trabalho, apesar de apresentar resultados com um ganho de acuracia no rastreamento se comparado com os algoritmos atuais, usa uma técnica computacionalmente custosa, o que foge do objetivo desse trabalho

O estudo "*Radar-Based Robust People Tracking and Consumer Applications*" (NINOS et al., 2022) aborda a aplicação de radar no rastreamento de pessoas como uma nova metodologia. Ele aponta as dificuldades encontradas na área, como oclusão, iluminação e movimentos rápidos, e propõem a utilização de radar como uma solução robusta para o rastreamento de pessoas. O estudo apresenta uma metodologia detalhada para o rastreamento de pessoas baseado nessa abordagem, descrevendo cada etapa do processo, desde a detecção inicial até o rastreamento. O trabalho, possui uma limitação em termos de equipamento, uma vez que requer o uso de um radar para sua aplicação.

No artigo "*Human and Object Detection using Machine Learning Algorithm*" (AHAMMED; GHOSH; ASHIK, 2022), é abordado o uso de algoritmos de aprendizado de máquina para a detecção de pessoas e objetos. Os autores propõem o uso de algoritmos como redes neurais convolucionais (CNNs) e algoritmos de detecção baseados em características para superar desafios como oclusão e variação de iluminação. Embora o artigo não aborde especificamente a detecção de pessoas no mar, ele explora diversas técnicas de pré-processamento de dados que podem melhorar a detecção de pessoas, que é uma etapa importante do pré-processamento e pode impactar no resultado final. O artigo é interessante e apresenta abordagens que podem ser consideradas em estudos futuros.

No trabalho de Yuhan Wang e Han Yang (WANG; YANG, 2022) é abordado a implementação de um sistema de rastreamento de pessoas no trânsito utilizando as redes neurais YOLOv5 e DeepSORT. O YOLOv5 é uma rede neural capaz de detectar e localizar pedestres em imagens em tempo real, enquanto o DeepSORT é um algoritmo de rastreamento que atribui identidades aos alvos rastreados. O estudo descreve o passo a passo do treinamento do modelo YOLOv5 com um conjunto de dados apropriado e a integração com o DeepSORT para o rastreamento das pessoas. Também são abordadas técnicas de pré-processamento de dados e pós-processamento para melhorar a precisão e a robustez do sistema. Os resultados mostraram um de-

sempenho satisfatório, com altas taxas de detecção e rastreamento de pedestres em diferentes cenários e condições ambientais.

O artigo "SMILEtrack: SiMilarity LEarning for Occlusion-Aware Multiple Object Tracking"(HSIANG et al., 2022) aborda os desafios no rastreamento de múltiplos objetos em vídeos, como ocultação, objetos similares e cenas complexas. Para enfrentar esses desafios, os autores propõem o SMILEtrack, um sistema de rastreamento de objetos inovador. O SMILEtrack combina um detector eficiente de objetos com um Módulo de Aprendizado de Similaridade baseado em Redes Siamesas (SLM). O SLM calcula a similaridade de aparência entre objetos, superando as limitações dos descritores de características em modelos convencionais. Ele utiliza um bloco de Autoatenção de Patches (PSA) inspirado na Visão Transformacional para gerar características confiáveis para a correspondência de similaridade. Além disso, o SMILEtrack apresenta um Módulo de Correspondência de Similaridade (SMC) com uma nova função GATE para a associação robusta de objetos em quadros de vídeo consecutivos. Essas inovações permitem ao SMILEtrack alcançar um equilíbrio aprimorado entre velocidade de execução e precisão de rastreamento. Nos testes realizados em conjuntos de dados como MOT17 e MOT20, o SMILEtrack supera métodos de referência, incluindo o BYTETrack, em termos de pontuações MOTA (Medida de Avaliação de Rastreamento de Múltiplos Objetos) e HOTA (Medida de Avaliação de Rastreamento de Alta Ordem).

Já o artigo "*Multi-object Crowd Real-time Tracking in Dynamic Environment Based on Neural Network*"(ZHANG; MA, 2022) aborda um método de rastreamento em tempo real de múltiplos objetos em ambientes dinâmicos de multidões. Nele é feita uma abordagem que utiliza uma rede neural treinada para aprender a representação e o movimento dos objetos na cena. Esse processo é dividido em duas etapas principais: detecção e rastreamento. Na etapa de detecção, a rede neural é usada para identificar os objetos de interesse na imagem de entrada. Em seguida, na etapa de rastreamento, a rede neural rastreia os objetos ao longo das sequências de vídeo, utilizando informações passadas para prever suas posições futuras. No artigo é demonstrado a eficácia do método proposto em diversos cenários de rastreamento em tempo real de múltiplos objetos em ambientes dinâmicos. Os resultados mostraram alta precisão e robustez na detecção e rastreamento de objetos, mesmo em situações desafiadoras. Além disso, apesar do sistema proposto ter obtido um resultado acima do que o modelo Yolov3, eles ainda estão inferiores a redes do tipo Yolov4.

No artigo "Analysis of Waves and Human Interaction Using Beach Webcams," publicado em 2023, aborda a importância da monitorização automática de transmissões de vídeo em praias para melhorar a segurança, monitorização ambiental, investigação e educação relacionadas às atividades na praia. Os autores propõem uma abordagem inovadora para monitorar a água por meio da análise de transmissões de vídeo de

praias. Diferentemente de trabalhos anteriores, essa abordagem analisa não apenas o comportamento dos da água ou dos seres humanos nos vídeos da praia, mas também as interações entre eles. A metodologia emprega redes neurais profundas para analisar as transmissões de vídeo de *webcams* de praia existentes. A análise inclui a detecção e o rastreamento de pessoas e ondas, bem como análises de nível superior. O estudo utiliza a caracterização das condições para o surf como um estudo de caso e demonstra a capacidade de estimar os valores de parâmetros-chave que podem ajudar a determinar a qualidade de um local de surf em um determinado momento. A abordagem apresentada combina a detecção e o rastreamento de pessoas em ambientes costeiros com a análise e o rastreamento de ondas. Isso permite uma compreensão mais profunda da "cena da praia" ao analisar simultaneamente pessoas e ondas. Os autores destacam a importância de ferramentas tecnológicas para analisar e prever o comportamento dos corpos de água, contribuindo para a segurança humana, especialmente em relação a riscos como ondas grandes e correntes perigosas.

Por fim, o artigo "SparseTrack: Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth" (LIU et al., 2023) aborda os desafios no rastreamento de múltiplos objetos em ambientes congestionados e com ocorrência frequente de oclusão. Os métodos de rastreamento existentes, embora eficazes, ainda enfrentam problemas nessas situações. A proposta do artigo é realizar uma decomposição esparsa em cenas densas para melhorar o desempenho na associação de alvos ocultos. Para isso, eles apresentam um método de estimativa de pseudo-profundidade para obter a profundidade relativa de alvos a partir de imagens 2D. Em seguida, desenvolvem um algoritmo chamado "depth cascading matching" (DCM) que utiliza as informações de profundidade para converter um conjunto de alvos densos em vários subconjuntos esparsos e realizar a associação de dados nesses subconjuntos em ordem da proximidade para a distância. Ao integrar o método de pseudo-profundidade e a estratégia DCM no processo de associação de dados, os autores propõem um novo rastreador chamado "SparseTrack". Este método oferece uma nova perspectiva para resolver o desafiador problema de rastreamento de objetos em ambientes congestionados. Usando apenas correspondência IoU, o SparseTrack alcança um desempenho comparável com os melhores métodos do estado-da-arte em conjuntos de dados de referência, como MOT17 e MOT20.

Na avaliação de diversos trabalhos, é fundamental identificar os pontos fortes e adaptá-los conforme a necessidade. Optar por algoritmos que não dependam de dispositivos adicionais, como radares, é uma consideração relevante. A seleção também deve levar em conta a viabilidade de execução em *hardwares* acessíveis em termos de custo, juntamente com considerações de eficiência computacional e restrições técnicas.

Nesse contexto, a habilidade de combinar e adaptar diferentes abordagens é muitas vezes imperativa para alcançar os resultados mais satisfatórios. É importante destacar que muitos dos algoritmos analisados foram desenvolvidos para rastrear um único indivíduo por vez, o que pode ser uma limitação em cenários mais complexos.

Além disso, alguns desses algoritmos empregaram técnicas que demandam recursos computacionais substanciais. Entretanto, ao explorar uma gama de métodos, os pesquisadores têm contribuído ao apresentar métricas de avaliação de algoritmos, bem como procedimentos para coleta e análise de dados. Vale ressaltar que diversos algoritmos foram projetados e testados em ambientes controlados e fechados, o que pode impactar sua aplicabilidade em situações do mundo real.

É crucial considerar as limitações e vantagens de cada algoritmo à luz das demandas do contexto específico e dos recursos disponíveis. Dessa forma, a seleção do método mais apropriado deve ser pautada em uma avaliação abrangente que considere tanto as capacidades técnicas quanto as limitações das diferentes abordagens.

3 Conceitos Básicos

3.0.1 RASTREAMENTO

Dado um objeto de interesse em um quadro, o objetivo de rastreá-lo significa localizar o dado objeto em quadros subsequentes, independente da velocidade do objeto, mudança de luz, mudanças do ponto de vista ou outras invariantes como oclusão (NEBEHAY; PFLUGFELDER, 2015a). O Modelo de movimento descreve como a imagem de um alvo pode mudar para diferentes movimentos do objeto (BABENKO; YANG; BELONGIE, 2009), enquanto o modelo de aparência é um algoritmo para casar um modelo estatístico da forma e aparência de um objeto, para uma nova imagem (ZHANG; ZHANG; YANG, 2012)

3.1 ALGORITMOS DE RASTREAMENTO

3.1.1 BOOSTING TRACKING

Esse algoritmo (GRABNER; GRABNER; BISCHOF, 2006b) é um método de rastreamento em tempo real desenvolvido com base em uma versão modificada do algoritmo AdaBoost. Nele, as características dos objetos são atualizadas em tempo de execução, permitindo que o algoritmo lide com mudanças nos objetos ao longo do tempo.

A figura abaixo apresenta uma visão geral de como o *Boosting Tracking* funciona:

A ideia principal desse algoritmo gira em torno de um classificador de boosting. Inicialmente, os classificadores fracos são inicializados aleatoriamente e cada um deles possui um conjunto de características da região-alvo, geradas por meio de várias técnicas. A taxa de erro de cada classificador fraco deve ser menor que 50%. Quando um novo exemplo chega, que pode ser tanto do objeto-alvo (exemplo positivo) quanto do fundo da imagem (exemplo negativo), os classificadores fracos de cada classificador de boosting são atualizados. O classificador com a menor taxa de erro é selecionado e os pesos do classificador forte são atualizados a cada iteração. No final, é feita uma combinação linear de todos os classificadores fracos.

A parte mais custosa desse processo é a atualização dos classificadores fracos. Para diminuir o tempo de processamento, foi introduzido um classificador global que é compartilhado por todos os classificadores de boosting. Essa abordagem oferece a vantagem de atualizar todos os classificadores fracos apenas uma vez quando um

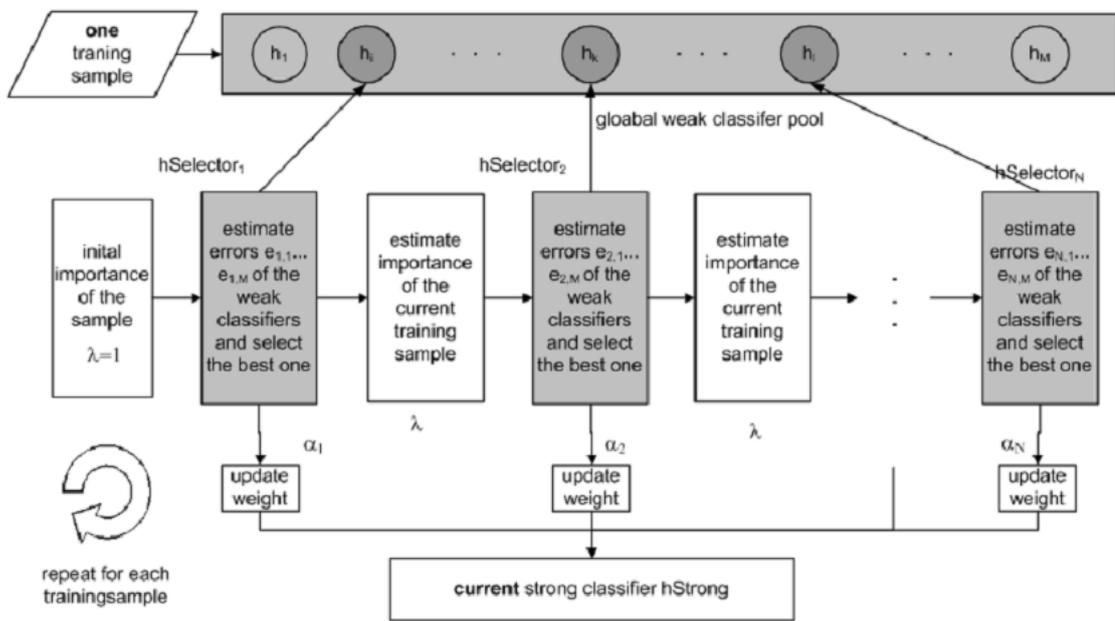


Figura 3 – Fluxograma do Boosting Tracking

novo exemplo chega.

Esse algoritmo possui características úteis para este projeto, pois é capaz de realizar o rastreamento em tempo real e foi projetado para lidar com diferentes tipos de fundos. Então agora será testado se ele consegue se adaptar ao problema de rastreamento de pessoas no mar.

3.1.2 MIL

O MIL (BABENKO; YANG; BELONGIE, 2009) é um algoritmo que se baseia em uma abordagem com aprendizado de máquina que é utilizada para lidar com problemas em que apenas informações parciais sobre os dados estão disponíveis. Ele é frequentemente aplicado em problemas de classificação em que as características estão disponíveis apenas para grupos ou conjuntos de instâncias, em vez de instâncias individuais. A imagem abaixo demonstra como esse modelo funciona:

O algoritmo envolve o conceito de "instâncias múltiplas", ou seja, em vez de rotular cada instância individualmente, o MIL trabalha com um conjunto que contém várias instâncias. Cada conjunto é rotulado como positivo se pelo menos uma instância dentro dela pertencer à classe positiva, caso contrário, é rotulada como negativa.

Durante o treinamento, o algoritmo MIL utiliza estratégias para inferir informações sobre as instâncias individuais com base nas informações de rotulagem dos conjuntos. Uma das abordagens comuns é a "bag-level", onde as características são extraídas e utilizadas para treinar um classificador. Esse classificador é então usado para

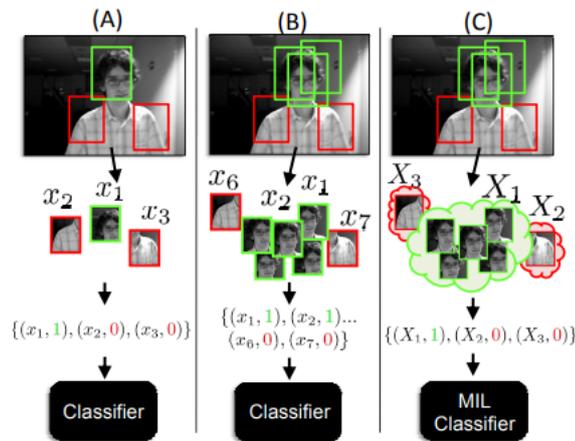


Figura 4 – Basicamente, o MIL utiliza um conjunto de características como exemplo e quando um novo objeto é utilizado, ele é classificado como positivo ou negativo baseado nesse conjunto.

inferir os rótulos das instâncias dentro do conjunto.

3.1.3 Median Flow

O *Median Flow* (KALAL; MIKOLAJCZYK; MATAS, 2010) é um algoritmo que utiliza uma combinação de características espaciais e temporais para realizar o rastreamento dos objetos. A imagem abaixo mostra a arquitetura do modelo:

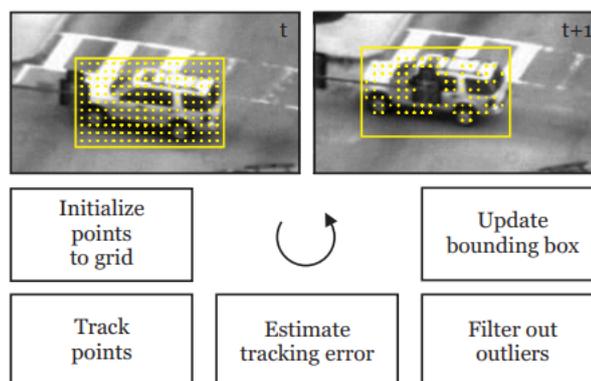


Figura 5 – Arquitetura do Median Flow

O funcionamento do *Median Flow* pode ser resumido da seguinte forma: Primeiro, o algoritmo começa selecionando uma região de interesse contendo o objeto a ser rastreado no primeiro quadro do vídeo. Essa região é usada como a detecção inicial do objeto. Depois, as características visuais são extraídas da região de interesse como descritores de características. Essas características servirão como representação do objeto a ser rastreado. Nos próximos quadros do vídeo, as características extraídas no passo anterior são comparadas com as características na nova região candidata.

A correspondência entre as características é estabelecida através de métodos como o casamento de pontos de interesse ou correspondência de descritores. Depois, o algoritmo utiliza uma filtragem temporal para suavizar as correspondências entre as características ao longo do tempo. Isso ajuda a remover possíveis falsos positivos e a manter uma boa acurácia no rastreamento. Com base nas correspondências estabelecidas e na filtragem temporal, o algoritmo estima o movimento do objeto entre os quadros. Isso permite acompanhar a trajetória e a posição do objeto ao longo do vídeo. Por fim, periodicamente, o modelo de rastreamento é atualizado com novas características extraídas da região de interesse. Isso permite que o algoritmo se adapte a mudanças na aparência do objeto ou a oclusões parciais que possam ocorrer durante o rastreamento.

O algoritmo *Median Flow* é poderoso para vídeos com movimentos rápidos ou variações bruscas de iluminação. Ele também é capaz de lidar com oclusão e mudanças na variação de luz do objeto. No entanto, o desempenho do algoritmo pode ser afetado por cenas com movimentos de rotação ou escala significativos.

3.1.4 Mosse

O Mosse (BOLME et al., 2010) é um algoritmo baseado na correlação de resposta entre o objeto de interesse e um filtro adaptativo. Ele funciona da seguinte forma: Primeiro, o algoritmo começa com a seleção de uma região de interesse contendo o objeto a ser rastreado em um quadro inicial do vídeo. A região é transformada em um modelo de referência utilizando uma técnica chamada filtro de resposta ao objeto. Depois, o filtro é adaptado para se ajustar às variações de aparência do objeto durante o rastreamento. Isso é feito usando uma técnica onde o filtro é atualizado iterativamente com novas informações a partir dos quadros subsequentes do vídeo. Por fim, o filtro adaptado é aplicado em cada novo quadro do vídeo, procurando por uma região que corresponda ao objeto de interesse. Isso é feito calculando a correlação entre o filtro e as regiões candidatas no quadro atual. A região com a maior correlação é selecionada como a posição estimada do objeto.

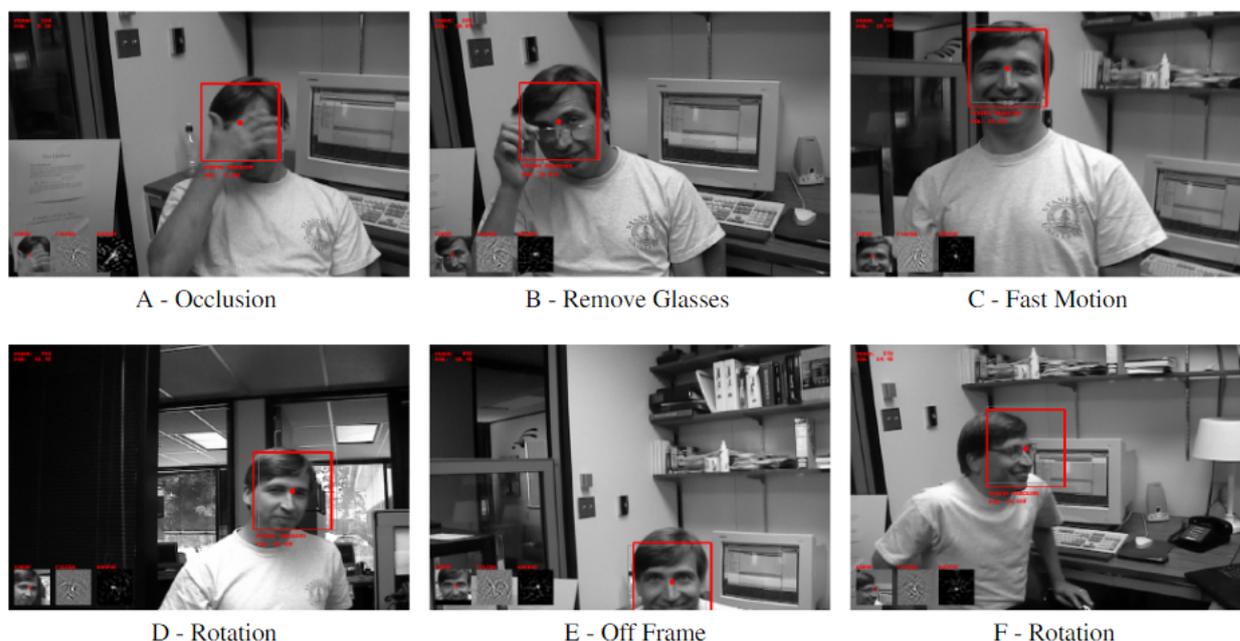


Figura 6 – Exemplo do MOSSE sendo utilizado em diversas situações, como rotação, oclusão e deslocamento

Uma das principais vantagens do algoritmo MOSSE é sua eficiência computacional, permitindo rastreamento em tempo real. Além disso, o uso de técnicas adaptativas e aprendizado online permite ao algoritmo lidar com variações de iluminação, oclusões parciais e outras mudanças na aparência do objeto. Porém, o Mosse não se comporta bem com variações bruscas de escala e rotação, que podem comprometer sua precisão em determinadas situações.

3.1.5 KCF

O KCF (HENRIQUES et al., 2012) é um algoritmo que foi desenvolvido para acompanhar um objeto em movimento ao longo do tempo, com um foco maior na aparência e posição do objeto. Seu funcionamento baseia-se em dois fundamentos principais: as características do objeto e o cálculo da correlação. As características do objeto são obtidas por meio de uma técnica chamada *Feature Pyramid*, que captura informações de várias escalas e níveis de detalhe. Isso permite que o KCF seja capaz de lidar com objetos de diferentes tamanhos e variações na aparência. Assim, se um objeto se dirige para mais perto ou mais distante da câmera, o algoritmo continua capaz de rastrear-lo.

Já o cálculo da correlação, é feito utilizando-se um filtro de correlação, que é construído a partir da representação do objeto. Esse filtro é treinado para maximizar a correlação entre o objeto de referência e as regiões candidatas no próximo quadro do vídeo. Dessa forma, o KCF é capaz de localizar o objeto mesmo quando ocorrem

mudanças na iluminação, oclusões parciais ou desfoque.

Uma vez que o filtro de correlação tenha sido treinado, ele pode ser aplicado para localizar o objeto em quadros subsequentes do vídeo. Isso é feito através de uma janela de busca que percorre a imagem, calculando a correlação entre a representação do objeto e as regiões candidatas. A região com a maior correlação é considerada como a localização atual do objeto.

O KCF também utiliza uma estratégia de aprendizado em tempo real para atualizar o filtro de correlação ao longo do tempo. Isso permite que o algoritmo se adapte a mudanças na aparência do objeto durante o rastreamento, garantindo um desempenho contínuo e preciso.

3.1.6 CPPMT

O CMT (NEBEHAY; PFLUGFELDER, 2015b) foi originalmente desenvolvido em Python pelos seus criadores. Com o objetivo de melhorar o desempenho, ele foi convertido e distribuído em uma nova versão em C++. Essa versão é chamada de CppMT e foi escolhida para este projeto, uma vez que os outros quatro algoritmos selecionados também estão implementados nessa linguagem. Dessa forma, evita-se interferência nos resultados devido à diferença de desempenho entre linguagens distintas.

O CppMT é um algoritmo que utiliza pontos-chave para rastrear objetos. A ideia principal dos pontos-chave é decompor o objeto que se deseja rastrear em partes individuais, ou seja, pontos no espaço, que são mais fáceis e rápidos de serem correspondidos do que utilizar a forma completa do objeto.

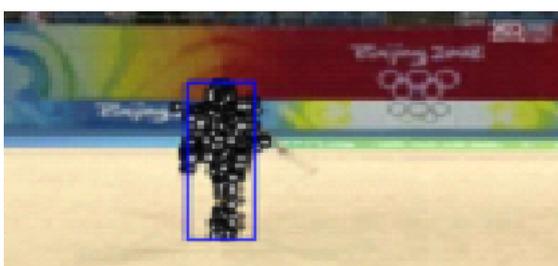


Figura 7 – Exemplo de pontos chave durante a detecção de uma pessoa usando o CppMT

A maioria dos algoritmos encontrados na literatura, que também utilizam pontos-chave para rastreamento, assume um modelo que pressupõe que o corpo de um objeto é rígido e permanecerá sempre assim. No entanto, um dos maiores problemas do rastreamento de objetos é que os corpos se deformam devido a oclusões ou movimentos, o que invalida esse tipo de modelo. Com o CppMT, porém, os pesquisadores descobriram que as informações da aparência dos objetos não estão permanentemente

incorporadas em seu modelo. Além disso, o descritor e o detector dos pontos-chave são independentes do algoritmo, permitindo que o usuário escolha o mais conveniente.

No algoritmo, é utilizado um modelo de aparência estático, baseado na aparência inicial do objeto e em seus descritores circundantes. Primeiro, uma região delimitada contendo o objeto-alvo é usada como entrada. Em seguida, é feita uma varredura global na imagem, procurando correspondências entre os pontos de interesse da entrada e pontos candidatos no quadro atual. Além disso, pontos que correspondem aos descritores do fundo da entrada são descartados. Em seguida, esses pontos são emparelhados com base em suas compatibilidades geométricas, refletindo diretamente deformações no objeto de interesse. Após essa etapa, é realizado um pós-processamento para remover correspondências ambíguas. A saída desse algoritmo é a região delimitada contendo o objeto no quadro atual.

Por fim, deve-se salientar que uma das desvantagens desse algoritmo é a necessidade de vídeos de alta resolução para funcionar corretamente. Caso contrário, o algoritmo não será capaz de detectar pontos-chave nos objetos.

3.1.7 GOTURN

O GOTURN (HELD; THRUN; SAVARESE, 2016) é um rastreador de objetos genéricos, o que significa que ele não é especializado em uma forma específica. Ele pode ser usado para detectar qualquer classe de objeto, independentemente de sua aparência. Para alcançar esse objetivo, o GOTURN utiliza uma rede neural artificial criada com base na arquitetura Caffe.

No entanto, diferentemente da maioria dos rastreadores que utilizam redes neurais, o treinamento do GOTURN é realizado offline, em vez de ser um treinamento online. No treinamento online, os pesos sinápticos são atualizados para cada elemento do conjunto de treinamento durante a fase de teste. No treinamento offline, a rede neural primeiro aprende uma relação genérica entre o objeto e o movimento. Em seguida, durante a fase de treinamento, todos os pesos são congelados e a corretude do algoritmo é testada. Esse procedimento permite que o sistema rastreie objetos de forma robusta, precisa e rápida.

A velocidade é um fator crucial para o GOTURN, uma vez que um de seus objetivos é alcançar uma alta taxa de quadros, próxima a 100 FPS. Isso diferencia o GOTURN de outros algoritmos da mesma classe que utilizam redes neurais, os quais geralmente são mais lentos e não adequados para sistemas em tempo real(HELD; THRUN; SAVARESE, 2016)

Um ponto importante a ser destacado sobre o GOTURN é que existem duas versões disponíveis para o público. Uma versão utiliza o processamento da GPU da



Figura 8 – A arquitetura do treinamento off-line. Na fase de treino, a rede neural aprende a relação objeto e movimento e durante a fase de testes avalia-se o resultado

máquina para executar o rastreamento, enquanto a outra utiliza a CPU. Como a GPU é um hardware especializado em cálculos matemáticos, é fácil concluir que ela proporciona uma taxa de quadros superior em comparação com a versão da CPU.

A figura a seguir resume o funcionamento do GOTURN:

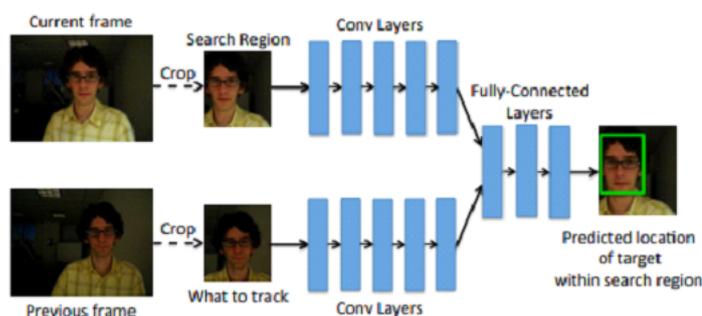


Figura 9 – Arquitetura do GOTURN

Primeiramente, o quadro inicial é recortado e o objeto que se deseja rastrear é centralizado na imagem. Isso permite que a rede neural rastreie qualquer objeto dado como entrada, independentemente de sua forma. Essa imagem é então utilizada como uma das entradas da rede. Para encontrar o objeto no quadro atual, é estimado que, como os objetos tendem a se mover lentamente no espaço, sua posição deve estar próxima da posição no quadro anterior. Assim, é criada uma região de busca com base nessa posição anterior. Para objetos que se movem mais rapidamente nos quadros, é possível aumentar a área de busca, mas isso aumenta a complexidade da rede. Essa imagem também é fornecida como uma das entradas da rede neural. Com base nas entradas obtidas pelos passos descritos anteriormente, a rede neural compara as características do objeto alvo com as da área de busca para determinar para onde o objeto se moveu. Ao realizar esse último passo, a rede fornece as coordenadas da posição superior esquerda e da posição inferior direita da caixa delimitadora que contém o objeto alvo.

Um ponto negativo do GOTURN é o alto consumo de memória durante a execução.

3.1.8 CSRT

O CSRT(LUKEZIC et al., 2018) é um algoritmo eficiente e que foi criado para conseguir ter uma boa acurácia em ambientes complexos e desafiadores, onde ocorram oclusões, mudanças de escala e variações de iluminação. Ele combina a informação de canais de cor e utiliza características espaciais para melhorar a robustez e a precisão do rastreamento.

O algoritmo funciona da seguinte forma: Primeiro o CSRT seleciona uma região de interesse (ROI) que contém o objeto a ser rastreado no primeiro frame do vídeo. Ele extrai características visuais dessa região, como histogramas de cores, texturas ou descritores de forma. Depois, é utilizado um modelo de fluxo óptico para estimar o deslocamento do objeto em cada frame subsequente do vídeo. Ele calcula a semelhança entre as características do objeto na região de interesse atual e as características nas regiões candidatas nos frames seguintes. Periodicamente, o CSRT atualiza o modelo de rastreamento para se adaptar a mudanças no objeto, como variações de iluminação, oclusões parciais ou alterações de escala. Isso é feito por meio de técnicas de aprendizado, como a atualização dos histogramas de cores ou a inclusão de novas informações de textura. Por fim, é realiza verificações de confiabilidade para garantir a precisão do rastreamento. Ele avalia a consistência espacial e temporal do objeto rastreado, bem como a qualidade das correspondências entre as características extraídas.

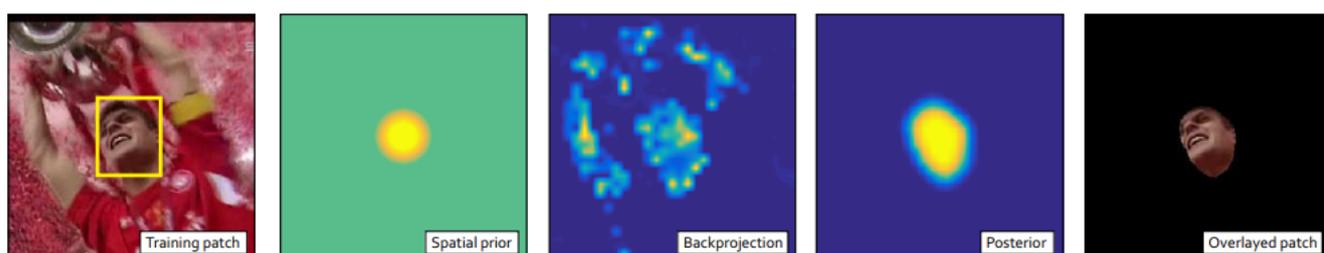


Figura 10 – Aqui, é apresentada a construção da região de interesse. Da esquerda para a direita: O objeto a ser rastreado, a extração de características, como cor e forma, e o filtro que é aplicado para definir a região de busca final.

3.2 Algoritmo Genético

O Algoritmo Genético é uma técnica inspirada na evolução biológica que visa solucionar problemas complexos de otimização. É composto por uma população de

cromossomos que são soluções candidatas, que podem evoluir ao longo de várias gerações.

Nesse processo, cada cromossomo com melhor adaptação são selecionados para reprodução, combinando seus genes através de operadores genéticos que simulam a recombinação genética, chamada de função de crossover e mutações aleatórias, gerando novas soluções. Além disso é possível guardar a melhor solução de cada geração, em um processo chamado de elitismo.

A avaliação da aptidão das soluções é realizada com base em uma função objetivo, chamada de função de *fitness*, que quantifica o desempenho de cada indivíduo. O objetivo é maximizar ou minimizar essa função, e com o passar das gerações atingir uma solução para o problema.

4 Metodologia

4.1 ROTULAÇÃO DA BASE DE DADOS

Não existe na literatura uma base de dados que contenha pessoas na praia com rótulos pré-determinados. Portanto, foi necessário criá-la. Pesquisadores da UFRPE se deslocaram até as praias de Boa Viagem, em Pernambuco, e gravaram vinte e quatro vídeos, capturando-os a partir de vários ângulos diferentes e com uma quantidade variada de pessoas tanto dentro como fora da água.

Após a obtenção dos vídeos, utilizou-se o programa VATIC ([VONDRICK; PATTERSON; RAMANAN](#),) para realizar a rotulação da base de dados. Nesse programa, é possível definir uma região que delimita as pessoas no mar. A cada quadro, o software tenta prever a posição do objeto de interesse. Entretanto, em várias ocasiões, essas previsões estavam incorretas, necessitando de ajustes na posição e no tamanho do polígono. Ao final, o programa gera um arquivo XML contendo as informações de posição e tamanho da região demarcada para cada objeto em cada quadro da imagem.

Contudo, após a rotulação de aproximadamente dezoito vídeos, foi identificado um erro no programa. Esse erro consistia em um número inconsistente de quadros para um mesmo vídeo. Por exemplo, se um vídeo possuía quatrocentos quadros, o VATIC, em alguns momentos, o reconhecia como tendo trezentos quadros, em outros momentos como tendo trezentos e cinquenta, e assim por diante. Como resultado, as anotações geradas apresentavam inconsistências e não puderam ser utilizadas. Foi necessário, então, optar por um novo software para concluir essa tarefa. O escolhido foi o *Dark Label* ([DARKLABEL...](#),), selecionado por sua simplicidade e baixos requisitos. Ele funciona de maneira semelhante ao VATIC, oferecendo duas opções de saída: um arquivo contendo as informações dos polígonos nos quadros, semelhante ao XML do VATIC, ou o próprio vídeo com as anotações visuais. Devido à restrição de tempo, foram selecionados apenas dez vídeos (de um total de vinte e cinco) com aproximadamente trinta segundos de duração cada. Esses vídeos também apresentavam uma quantidade melhor de pessoas, bem como uma variação de posição melhor, com pessoas entrando e saindo do mar, entrando e saindo do ângulo da câmera, formando uma base de dados mais variada.



Figura 11 – Primeiro vídeo



Figura 12 – Segundo vídeo



Figura 13 – Terceiro vídeo



Figura 14 – Quarto vídeo

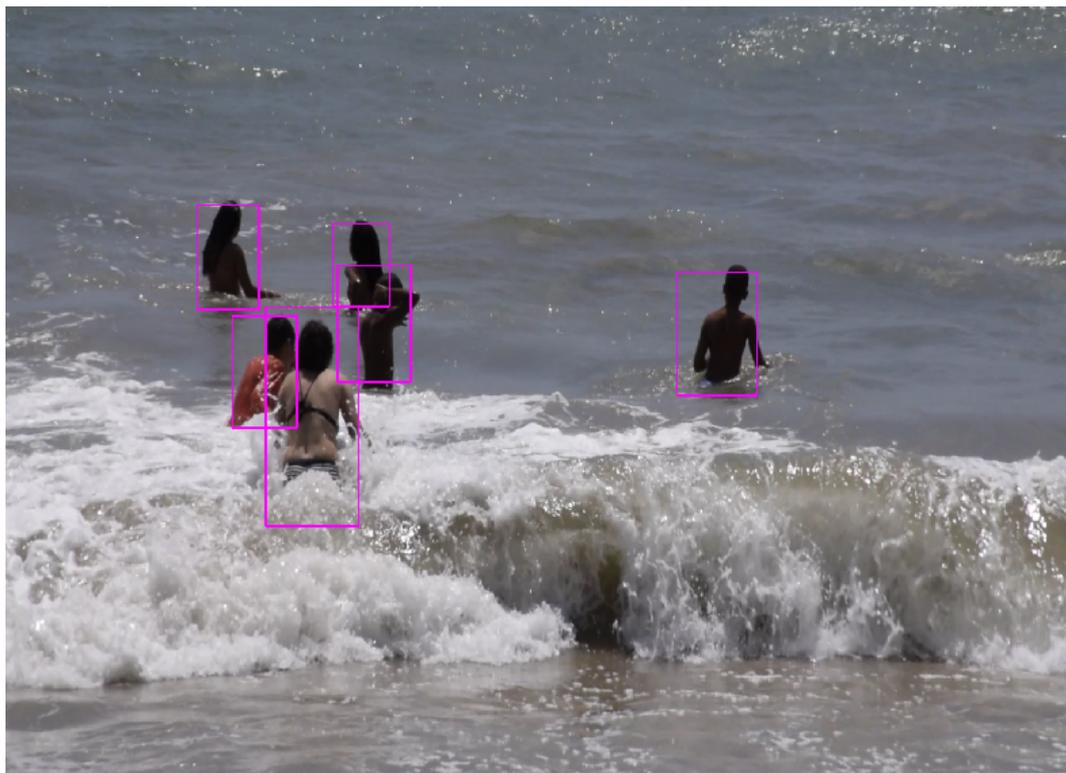


Figura 15 – Quinto vídeo



Figura 16 – Sexto vídeo



Figura 17 – Sétimo vídeo

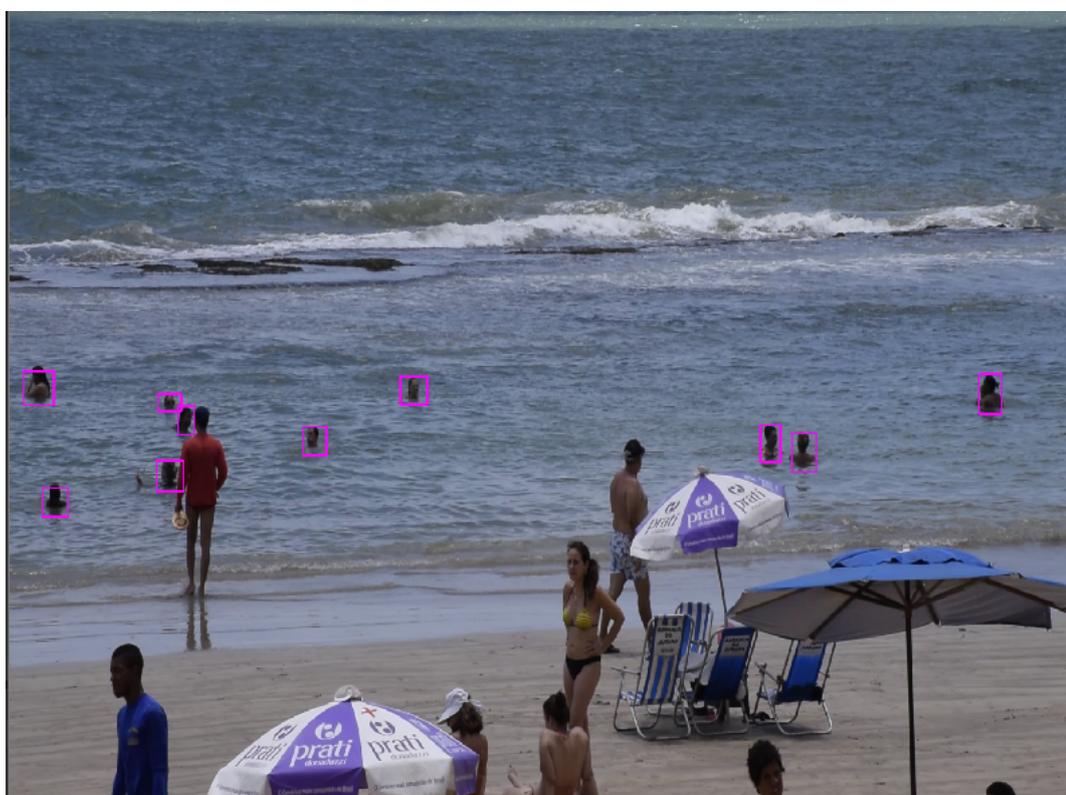


Figura 18 – Oitavo vídeo

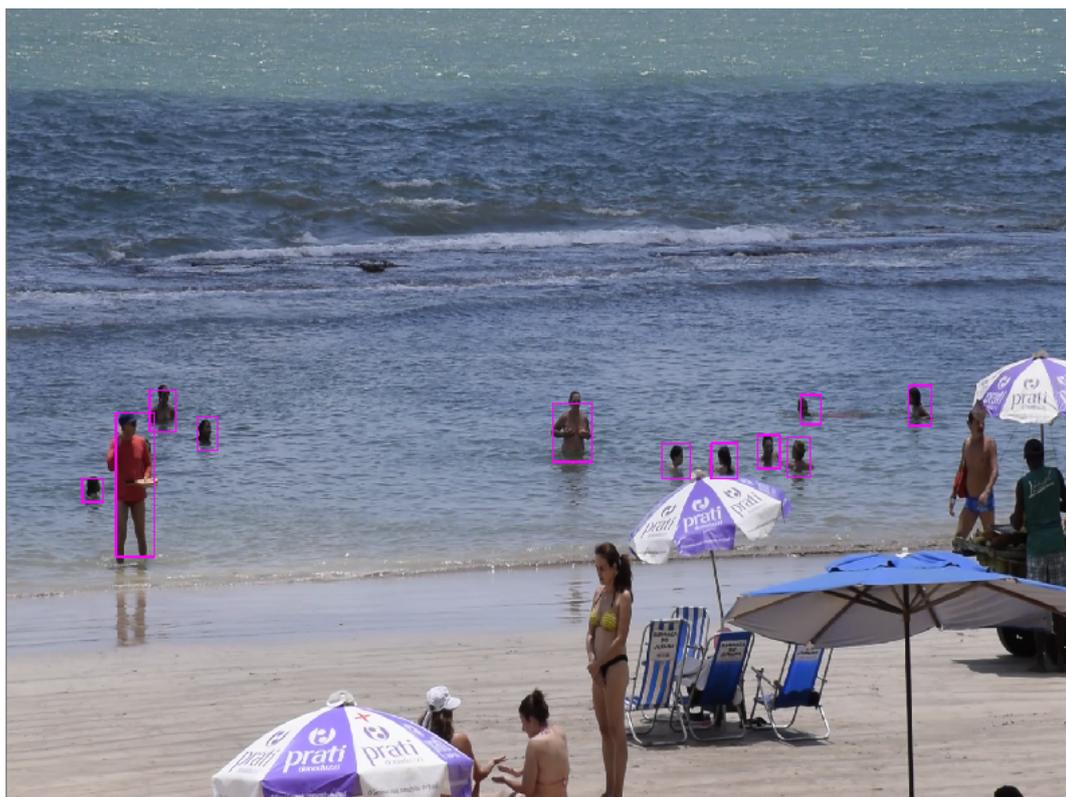


Figura 19 – Nono vídeo



Figura 20 – Décimo vídeo

4.2 ESCOLHA DOS ALGORITMOS

Primeiro, deve-se salientar que o sistema, cuja o qual o projeto desenvolvido nesse artigo refere-se à terceira etapa, tem como alvo final um dispositivo Raspberry, que consiste de um computador embarcado que pode ser acoplado a diversos módulos. Assim, o dispositivo será acoplado a uma câmera digital, tornando-o uma central de monitoramento. Porém este tipo de dispositivo, possui um hardware limitado, se comparado com computadores de mesa. Por isso, o foco desse trabalho foi em algoritmos que possam ser utilizados em dispositivos com poder de processamento reduzido, já que o alvo não são centrais computacionais de alto desempenho que podem executar algoritmos de aprendizagem de máquina robustos.

Diversos algoritmos foram buscados como possíveis candidatos. Porém a maioria ou não havia versão disponível na internet, ou foram desenvolvidas em versões muito ultrapassadas do OPENCV e do Visual Studio, o que causou diversos erros por incompatibilidade com o sistema. Além disso, muitos algoritmos estavam sem nenhuma documentação sobre qual versão do OPENCV foi utilizada para implementá-los, nem qual o ambiente de desenvolvimento integrado foi usado e nesses casos, foi necessário testar diversas versões do OPENCV, tanto antigas, como as mais recentes e em muitos casos, os algoritmos não paravam de apresentar erros, tornando-os inutilizáveis.

Por fim, optou-se por escolher algoritmos que utilizassem abordagens distintas para rastrear as pessoas, evitando que houvesse algoritmos muito semelhantes entre si. Assim, escolheu-se os oito algoritmos apresentados na etapa anterior. Dentre os escolhidos, porém, apenas seis puderam ser testados com sucesso. O GOTURN, além de muito tempo para ser executado, o que o torna inviável para uma aplicação em tempo real, exigia uma quantidade grande de memória, quase 100% da memória disponível na máquina de testes. Já o CppMT executava o código de teste com um erro em tempo de execução que não foi possível corrigir. Sendo assim, foram utilizados apenas o Boosting (GRABNER; GRABNER; BISCHOF, 2006a), o CSRT (LUKEZIC et al., 2018), o KFC (DANELLIAN et al., 2014), o MIL (BABENKO; YANG; BELONGIE, 2009), o Moose (BOLME et al., 2010) e o Median Flow (KALAL; MIKOLAJCZYK; MATAS, 2010)

4.3 AVALIAÇÃO DOS ALGORITMOS

Para realizar a avaliação, foi necessário modificar o código de entrada de todos os algoritmos, pois eles foram originalmente implementados para aceitar apenas um objeto por vez. Portanto, eles foram adaptados para lidar com múltiplos alvos. Isso foi feito reescrevendo a entrada original, para que ao invés de aceitar apenas um, uma lista de objetos fossem passadas.

A métrica escolhida para a avaliação foi a Interseção sobre União (IoU), também conhecida como *Jaccard Index*, é uma métrica comumente usada para avaliar a sobreposição ou a semelhança entre dois conjuntos. Seu resultado é um número que varia entre zero e um. A taxa de acerto do algoritmo está definido abaixo:

Taxa de acerto R:

$$R = \frac{1}{x} \sum_{i=1}^x \frac{A_i \cap B_i}{A_i \cup B_i}$$

Onde:

A : Rotulação da base

B : Resultado do rastreamento do algoritmo

x : Número de pessoas em um vídeo

i : Índice da pessoa dentro do vídeo

j : Índice do vídeo

O resultado é então multiplicado por 100, obtendo a porcentagem de acurácia. Depois, retira-se a média e o desvio padrão dos 10 vídeos.

Além disso, também foi calculado o tempo de execução de cada algoritmo para cada vídeo, e em seguida foi feita a média do tempo. Como os vídeos da base de dados foram gravados a uma resolução de quadro muito alta, os algoritmos demoram muito tempo para executar totalmente, e como o sistema pretende ser uma aplicação em tempo real, foi feito então versões onde eram puladas os seguintes números de quadros: trinta, sessenta e noventa. Assim, foi possível avaliar se ao pular o quadro, ouve uma queda de desempenho.

4.4 OTIMIZAÇÃO COM O ALGORITMO GENÉTICO

Depois que todos os algoritmos forem avaliados, o que obteve o melhor resultado será selecionado para se aplicar um algoritmo genético, visando verificar se existe algum ganho na acurácia. Essa otimização será feita da seguinte forma: Cada algoritmo tem o seguinte conjunto de parâmetros que determinam seu comportamento:

- cheb_attenuation
- filter_lr
- gsl_sigma

- histogram_lr
- hog_clip
- hog_orientations
- kaiser_alpha
- psr_threshold
- scale_lr
- scale_model_max_area
- scale_sigma_factor
- template_size
- weights_lr

Esses parâmetros serão usados como o cromossomo que compõem o gene do indivíduo e esses valores serão inicializados com valores aleatórios inicialmente. Após a inicialização, o algoritmo executa o rastreamento e obtém uma taxa de acurácia para aquele gene.

Então, a função de *fitness* será iniciada e ela foi definida neste projeto da seguinte forma: Foi utilizada a acurácia obtida pelo algoritmo no rastreamento, ou seja quanto maior a porcentagem de acerto, maior é o valor de *fitness*. Com esse valores calculados, o processo de seleção utilizado foi a roda da fortuna, onde as aptidões são somadas e ordenadas do maior para o menor. Um número aleatório é gerado entre zero e a aptidão acumulada, e então o primeiro valor maior que o número escolhido é selecionado como pai para aplicar os operadores de crossover. Isso é feito da seguinte forma: É escolhido aleatoriamente um pai e uma mãe do conjunto de soluções atuais obtidos da etapa anterior e então é feito a média geométrica de cada cromossomo. Essa fórmula está definida abaixo:

$$\sqrt{a_i \cdot c_i}$$

A mutação *Creep* é utilizada para restringir o intervalo de busca dentro das soluções atuais. Essencialmente, para cada parâmetro, a mutação tem uma chance de 5% de ocorrer. Quando um parâmetro é selecionado para mutação, um número aleatório r é gerado no intervalo de 0 a 1. Esse número aleatório é então multiplicado pelo valor atual do parâmetro para determinar o valor mutado. A fórmula para a mutação *Creep* em um parâmetro p é dada por:

$$p \cdot r$$

Isso significa que o novo valor do parâmetro é uma pequena variação (dependente de r) do valor atual do parâmetro, e essa variação é aplicada com uma probabilidade de 5% para cada parâmetro. Essa técnica de mutação ajuda a explorar o espaço de busca de maneira controlada, mantendo-se próximo das soluções atuais.

Por fim, a melhor solução de cada geração foi salva através do elitismo. Esse processo é repetido até que toda a nova população seja gerada. Adotou-se por padrão utilizar apenas vinte gerações, visto que o processo é muito demorado e depois de algumas dezenas de gerações, o ganho é nulo ou negativo.

4.5 MÁQUINA DE TESTES

O computador utilizado para os testes do algoritmo possui a seguinte configuração: oito *gigabytes* de memória RAM DDR4, processador Intel Core i5-7200 e placa de vídeo GeForce 940Mx. No entanto, a placa de vídeo não foi utilizada nos testes, pois todos os algoritmos utilizados requerem apenas processamento da CPU.

5 Resultados

Primeiramente, os video foram rotulados manualmente utilizando a ferramenta VATIC. Cada pessoa no video foi demarcada quadro por quadro, criando assim a base de dados.



Figura 21 – Pessoas Rotuladas

Em seguida, foi desenvolvido um programa que lê as anotações previamente criadas e inicia os algoritmos com a rotulação inicial de cada indivíduo como ponto de partida. Esse programa executa o algoritmo para comparar o resultado obtido e as anotações anteriores. Na implementação desses algoritmos, foi feito o uso do OpenCV, uma poderosa ferramenta de visão computacional amplamente disponível de forma gratuita na internet.

Abaixo, está o resultado obtido para cada algoritmo, pulando *zero, trinta, sessenta e noventa frames*. Os valores entre parênteses representam o desvio padrão.

A Tabela 1 acima se refere a média de acurácia com o pulo de quadros diferentes. A seguir, uma breve explicação sobre o desempenho de cada algoritmo.

O Median Flow, é um algoritmo que é descrito pelos próprios autores como um modelo que se comporta adequadamente com mudanças bruscas de movimento, porém não se adapta a mudança de escala nos objetos. Como nos videos, é muito comum as pessoas se afastarem ou se aproximarem da câmera, isso explica a baixa

Tabela 1 – Acurácia média de rastreamento

	0 Quadros	30 Quadros	60 Quadros	90 Quadros
MIL	42.85% (5.64)	39.23% (12.48)	33.14% (11.43)	31.67% (11.26)
Boosting	46.35% (4.96)	39% (7.24)	33.49% (11.83)	30.25% (11.33)
MedianFlow	33.58% (7.35)	14.5% (10)	13% (9.42)	11.16% (7.85)
KCF	33.62% (12.67)	15.39%(7.9)	13.81% (7.14)	12.65% (6.84)
CSRT	49.97% (7.73)	49.55% (7.5)	47.54% (7.57)	43.17% (7.34)
Mosse	33.41% (10.74)	8.88% (7.27)	5.52% (4.92)	4% (4.12)

acurácia. Além disso, é um modelo baseado exclusivamente nas características do objeto, e como a maioria dos objetos a serem rastreados encontram-se a uma distância considerável, os resultados ficaram abaixo do esperado.

O Mosse, apesar de ter sido o algoritmo que mais rapidamente executou os vídeos, apresentou resultados abaixo do esperado, pois, como ele é um algoritmo baseado apenas em forma e movimento, e as pessoas a serem rastreadas estão sempre a uma distância considerável da câmera, seu modelo não se mostra adequado para essa situação

O KCF se mostrou um modelo com resultados ruins em cenários onde é difícil extrair características do objeto devido a um fundo constante, como imagens no mar, e quando as pessoas estão distantes da câmera. Isso ocorre porque o KCF é altamente dependente da forma e das características físicas e de movimento do objeto que está sendo rastreado. Com o tempo, a falta de características relevantes se acumula, resultando em uma queda na acurácia do algoritmo.

O MIL é um algoritmo que baseia-se apenas nas características físicas do objeto. Ele utiliza um modelo que utiliza um conjunto de características físicas do objeto, porém como a maioria deles está a uma distância grande da câmera, esse conjunto acaba ficando desatualizado devido a prevalência da distância e do fundo da imagem. Além disso, por ser uma abordagem baseada em aprendizagem de máquina, o tempo de processamento acaba tornando-se muito alto, inviabilizando o seu uso em aplicações em tempo real.

Já o Mosse é um algoritmo que além de se basear apenas nas características físicas do objeto, também utiliza muito a imagem de fundo do objeto a ser rastreado como um falso positivo. Como a maioria dos objetos que deseja-se rastrear encontrassem muito longe da câmera, o modelo extrai poucos características e começa a acumular um erro muito grande. Como o fundo também é utilizado, ele começa a ser confundido com o próprio objeto. Com isso, o objeto a ser rastreado começa a ser usado como um falso positivo, o que faz o algoritmo cair na sua taxa de acurácia.

Por outro lado, o Mosse é um algoritmo que pode gerar falsos positivos ao utilizar demais a imagem de fundo como referência. Como o Mosse é baseado em ca-

racterísticas, quando as pessoas a serem rastreadas estão muito distantes da praia, ocorre um acúmulo de erros significativo e o fundo começa a se confundir com o próprio objeto. Isso leva a uma taxa de acurácia reduzida, pois o próprio objeto pode ser interpretado como um falso positivo.

O CSRT, por ser um modelo baseado tanto em características como em cor se mostrou uma combinação poderosa, uma vez que quando há falta de características no objeto a ser rastreado, o filtro de cor ainda pode ser utilizado mantendo consistência no rastreamento quadro a quadro. Isso resulta na taxa de acurácia mais elevadas e consistente nos resultados.

Quanto aos desvios padrões dos algoritmos, é possível identificar uma característica notável: o desvio padrão é baixo. Isso sugere que os algoritmos mantêm uma notável consistência em diferentes cenários de vídeos. Em outras palavras, esses algoritmos demonstram uma habilidade de adaptação a uma variedade de ângulos, horários do dia e distâncias, sem resultar em uma perda significativa de acurácia.

Portanto, a análise dos desvios padrões reforça a capacidade dos algoritmos em se adaptar de forma consistente a diversos desafios apresentados pelos vídeos, contribuindo para a confiabilidade e a qualidade do rastreamento mesmo diante de cenários variados e desafiadores.

Tabela 2 – Tempo de rastreamento em minutos

	0 Quadros	30 Quadros	60 Quadros	90 Quadros
MIL	47	1.8	1.4	0.9
Boosting	18	1.3	0.8	0.7
KCF	12	1.2	0.7	0.5
MedianFlow	18	1.4	0.7	0.5
CSRT	28	1.5	1.3	1.1
Mosse	10	1.3	0.4	0.1

A tabela 2 apresentada acima mostra o tempo de execução dos algoritmos em minutos. Como pode-se observar, ao se reduzir o número de quadros, a velocidade de processamento cai drasticamente em todos os algoritmos.

Dos dados acima, constata-se que o CSRT foi o algoritmo que obteve a melhor taxa de acurácia entre todos os escolhidos. Além disso, observa-se que ao diminuir o número de quadros, ele também foi o algoritmo que conseguiu ter a menor queda no seu desempenho. Sendo assim, ele foi o algoritmo escolhido para ser otimizado com o algoritmo genético. Abaixo, estão os parâmetros que foram otimizados com o algoritmo:

- cheb_attenuation
- filter_lr

- `gsl_sigma`
- `histogram_lr`
- `hog_clip`
- `hog_orientations`
- `kaiser_alpha`
- `psr_threshold`
- `scale_lr`
- `scale_model_max_area`
- `scale_sigma_factor`
- `template_size`
- `weights_lr`

Os resultados obtidos são mostrados abaixo:

Tabela 3 – Média de acurácia do CSRT com otimização

Quadros	Média
0 Quadros	58.12%
30 Quadros	56.03%
60 Quadros	54.89%
90 Quadros	51.64%

A tabela 3 mostra que houve uma melhoria notável na taxa de acurácia, evidenciando um aumento expressivo de quase 20% em comparação com o desempenho obtido nos cenários de zero e trinta quadros. Isso pode ser explicado da seguinte forma: Ao considerar que o CSRT é configurável através dos diversos parâmetros que foram mostrados acima e que esses parâmetros, o limiar do histograma, orientação de *hog*, podem ser ajustados para se adequarem de forma mais precisa para cada cenário. No contexto de um ambiente praiano, por exemplo, onde o mar apresenta uma tonalidade azul diferenciada das pessoas, é crucial empregar um algoritmo que leve em consideração a detecção de cores. Além disso, a capacidade de detectar movimentos de alta intensidade é vital, dada a agitação constante na praia. Por outro lado, a ênfase nas características físicas pode ser reduzida, considerando que muitas pessoas estão distantes da câmera, tornando o excessivo rastreamento de detalhes específicos desnecessários. Como esses parâmetros estão sendo variados, acabam encontrando uma combinação que se adequa melhor a base de dados criada nessa monografia.



Figura 22 – O resultado do rastreamento. Em rosa, a marcação manual da base e em vermelho, o rastreamento do algoritmo. Nos estágios iniciais, o rastreamento é muito preciso, porém vai perdendo essa acurácia com o tempo

Além disso, os resultados sugerem que os parâmetros padrão do algoritmo CSRT no *OpenCV* não estão devidamente otimizados para enfrentar o desafio específico de rastreamento de pessoas em ambientes marítimos., havendo espaço para novas formas de otimização

6 Conclusão

A tarefa de rastrear banhistas em uma praia é complexa, computacionalmente custosa e a maioria dos algoritmos não foram especificamente desenvolvidos com esse objetivo. Com os testes, foi possível inferir como esses algoritmos se comportam nesse ambiente e verificar a viabilidade do uso desses algoritmos em *hardwares de baixo desempenho*

A primeira conclusão é que ao diminuir o número de quadros, há uma perda de eficácia no algoritmo. A segunda é que quanto mais dependente o algoritmo é da forma e da movimentação do objeto, maior é a perda dessa acurácia pois quanto menos quadros, menos precisão ao rastrear o movimento. Como o CSRT também utiliza as cores e a forma, ele não é tão dependente de uma taxa de quadro altas, conseguindo manter resultados acima da média, mesmo com o pulo de quadros. Com esses testes, fica demonstrado que a abordagem mais eficiente para o rastreamento de pessoas no mar é utilizar algoritmos que combinem o uso de modelos que armazenam as características físicas do objeto e seu histograma de cores.

Além disso, foi demonstrado que ao reduzir o número de quadros, todos os algoritmos experimentaram uma queda significativa no tempo de processamento. Isso possibilitou a realização do rastreamento praticamente em tempo real. E o CSRT demonstrou uma perda de eficácia mínima ao se reduzir o número de quadros. Esse comportamento mostra que além de manter a precisão, ele também aproveitava os benefícios da diminuição do tempo de execução. Isso destaca a capacidade do CSRT de otimizar o equilíbrio entre precisão e eficiência, tornando-o uma escolha promissora para cenários em que a velocidade é crucial sem comprometer muito a qualidade do rastreamento.

Obteve-se ainda uma melhoria na acurácia de quase vinte por cento ao se aplicar o algoritmo genético na otimização dos parâmetros do CSRT.

Como foi possível observar nos resultados obtidos nesse trabalho, a otimização dos algoritmos utilizando a abordagem de algoritmo genéticos trouxe um ganho de acurácia considerável de quase vinte por cento. Isso mostra que a aplicação de um algoritmo genético para otimizar os parâmetros do algoritmo CSRT demonstra um avanço significativo na busca por uma configuração mais precisa e adaptável. Diante disso, fica evidente que o espaço para futuras otimizações é vasto, tanto na melhoria da precisão por meio da exploração de diferentes combinações de parâmetros quanto na exploração de novas estratégias utilizando algoritmos genéticos, redes neurais, e etc. Além disso, também é importante que novas bases sejam utilizadas em ambientes

diferentes e que sejam encontrados novos parâmetros para esses ambientes.

Referências

- ABDULGHAFOOR, H. N. A. N. H. A novel real-time multiple objects detection and tracking framework for different challenges. *IEEE Transactions on Pattern Analysis and Machine*, IEEE, 2022. Citado 2 vezes nas páginas 14 e 20.
- AHAMMED, M. T.; GHOSH, S.; ASHIK, M. A. R. Human and object detection using machine learning algorithm. In: IEEE. *2022 Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON)*. [S.l.], 2022. Citado na página 21.
- BABENKO, B.; YANG, M.-H.; BELONGIE, S. Visual tracking with online multiple instance learning. In: IEEE. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. [S.l.], 2009. p. 983–990. Citado 4 vezes nas páginas 15, 25, 26 e 41.
- BJERGE, K.; MANN, H. M. R.; HØYE, T. T. Real-time insect tracking and monitoring with computer vision and deep learning. 2021. Citado na página 20.
- BOLME, D. S. et al. Visual object tracking using adaptive correlation filters. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2010. Citado 3 vezes nas páginas 15, 28 e 41.
- DANELLIAN, M. et al. Adaptive color attributes for real-time visual tracking. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. [S.l.: s.n.], 2014. p. 1090–1097. Citado na página 41.
- DARKLABEL: An Annotation Tool. <<https://github.com/darkpgmr/DarkLabel>>. Acesso em: 21 de agosto de 2023. Citado na página 35.
- GHEISARI, M. et al. A novel enhanced algorithm for efficient human tracking. *International Journal of Informatics and Communication Technology (IJ-ICT)*, v. 11, n. 1, 2022. Citado na página 20.
- GRABNER, H.; GRABNER, M.; BISCHOF, H. Real-time tracking via on-line boosting. In: *BMVC*. [S.l.: s.n.], 2006. v. 1, p. 6. Citado 2 vezes nas páginas 15 e 41.
- GRABNER, H.; GRABNER, M.; BISCHOF, H. Real-time tracking via on-line boosting. In: *BMVC*. [S.l.: s.n.], 2006. v. 1, p. 6. Citado na página 25.
- HELD, D.; THRUN, S.; SAVARESE, S. Learning to track at 100 fps with deep regression networks. *European Conference Computer Vision (ECCV)*, 2016. Citado na página 31.
- HENRIQUES, J. F. et al. Exploiting the circulant structure of tracking-by-detection with kernels. In: *Proceedings of the European Conference on Computer Vision*. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 15 e 29.
- HOSANG, J.; BENENSON, R.; SCHIELE, B. How good are detection proposals, really? *MPI Informatics*, Saarbrücken, Germany, 2013. Citado na página 19.

- HSIANG, W. Y. et al. Smiletrack: Similarity learning for occlusion-aware multiple object tracking. *College of AI and Green Energy and Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan*, 2022. Citado na página 22.
- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Forward-backward error: Automatic detection of tracking failures. In: IEEE. *Pattern Recognition (ICPR), 2010 20th International Conference on*. [S.l.], 2010. p. 2756–2759. Citado 3 vezes nas páginas 15, 27 e 41.
- LIU, Z. et al. Sparsetrack: Multi-object tracking by performing scene decomposition based on pseudo-depth. 2023. Citado na página 23.
- LUKEZIC, A. et al. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, Springer, v. 126, n. 7, p. 671–688, 2018. Citado 3 vezes nas páginas 15, 33 e 41.
- NEBEHAY, G.; PFLUGFELDER, R. Clustering of static-adaptive correspondences for deformable object tracking. *IEEE*, 2015. Citado na página 25.
- NEBEHAY, G.; PFLUGFELDER, R. Clustering of Static-Adaptive correspondences for deformable object tracking. In: *Computer Vision and Pattern Recognition*. [S.l.]: IEEE, 2015. Citado na página 30.
- NINOS, A. et al. Radar-based robust people tracking and consumer applications. *IEEE Sensors Journal*, IEEE, 2022. Citado na página 21.
- SPAMPINATO, C.; PALAZZO, S.; GIORDANO, D. Evaluation of tracking algorithm performance without ground-truth data. *Department of Electrical, Electronics and Computer Engineering, University of Catania Italy*, 2011. Citado na página 19.
- VONDRICK, C.; PATTERSON, D.; RAMANAN, D. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, Springer Netherlands, p. 1–21. ISSN 0920-5691. 10.1007/s11263-012-0564-1. Disponível em: <<http://dx.doi.org/10.1007/s11263-012-0564-1>>. Citado na página 35.
- WANG, Y.; YANG, H. Multi-target pedestrian tracking based on yolov5 and deepsort. In: IEEE. *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*. [S.l.], 2022. Citado na página 21.
- WEN, L. et al. Multiple target tracking based on undirected hierarchical relation hypergraph. *Center for Biometrics and Security Research & National Laboratory of Pattern Recognition Institute of Automation, Chinese Academy of Sciences, China*, 2014. Citado na página 19.
- WU, B.; NEVATIA, R. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *University of Southern California, Institute for Robotics and Intelligent Systems*, Los Angeles, USA, 2006. Citado na página 18.
- 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 (CVPR)*. [S.l.], 2013. p. 2411–2418. Citado na página 19.

WU, Y.; LIM, J.; YANG, M.-H. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 37, n. 9, p. 1834–1848, 2015. Citado na página 20.

YANG, T. et al. Real-time multiple objects tracking with occlusion handling in dynamic scenes. *College of Automatic Control, Northwestern Polytechnical University, Xi'an, China and National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China*, 2004. Citado na página 18.

ZHANG, F.-Q.; MA, L.-J. Multi-object crowd real-time tracking in dynamic environment based on neural network. *Journal of Network Intelligence*, Taiwan Ubiquitous Information, v. 7, n. 2, 2022. Citado na página 22.

ZHANG, K.; ZHANG, L.; YANG, M. Real-time compressive tracking. In: *Proc. ECCV'12*. [S.l.: s.n.], 2012. Citado na página 25.

ZHOU, Q.; AGGARWAL, J. Object tracking in an outdoor environment using fusion of features and cameras. In: IEEE. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2006. p. 1074–1081. Citado na página 18.