UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Rastreamento em Tempo Real de Múltiplos Objetos por Associação de Detecções

Dissertação de Mestrado

Michel Conrado Cardoso Meneses

São Cristóvão – Sergipe

2019

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Michel Conrado Cardoso Meneses

Rastreamento em Tempo Real de Múltiplos Objetos por Associação de Detecções

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Prof. Dr. Leonardo Nogueira Matos
Coorientador(a): Prof. Dr. Bruno Otavio Piedade Prado

São Cristóvão – Sergipe

2019

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do Curso de Mestrado em Ciência da Computação-UFS. Candidato: Michel Conrado Cardoso Meneses.

Em 25 dias do mês de outubro do ano de dois mil e dezenove, com início às 09h00min, realizou-se na sala de seminário do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato Michel Conrado Cardoso Meneses, que desenvolveu o trabalho intitulado: *"Rastreamento em Tempo Real de Múltiplos Objetos por Associação de Detecções"*, sob a orientação do Prof. Dr. Leonardo Nogueira Matos. A Sessão foi presidida pelo Prof. Dr. Leonardo Nogueira Matos (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. André Britto de Carvalho (PROCC/UFS) e, em seguida, ao Prof. Paulo Salgado Gomes de Mattos Neto (UFPE). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO *"(aprovado/reprovado)"*. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária *"Prof. José Aloísio de Campos"*, 25 de outubro de 2019.

Prof. Dr. Leonardo Nogueira Matos
(PROCC/UFS)
Presidente

Prof. Dr. André Britto de Carvalho
(PROCC/UFS)
Examinador Interno

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
(UFPE)
Examinador Externo

Michel Conrado Cardoso Meneses
Candidato

Este trabalho é dedicado à minha família,
que sempre acreditou e investiu em mim.

Agradecimentos

Ao Criador, por me conceder todas as condições para triunfar; ao meu pai Messias Meneses, por sempre acreditar em mim; à minha mãe Carla Meneses, por sempre me lembrar de que a vida é feita para se divertir; à minha irmã Giovanna Meneses, pelo companheirismo, pelo carinho e pela admiração incondicionais que sempre demonstrou; à minha amiga Natasha Carmo, pela atenção e pelo apoio ao longo de todas as etapas desta jornada; à minha amiga Mislene Nunes, pelos momentos de aprendizado e descontração no melhor laboratório do PROCC; aos meus amigos Luiz Henrique da Costa e Reneílson Santos, pelas inúmeras, infindáveis e sempre bem-humoradas conversas sobre empreendedorismo e política; aos meus orientadores Leonardo Matos e Bruno Prado, por insistirem na minha permanência no programa e por investirem seu tempo na minha formação; ao professor André Carvalho, por abrir as portas e me acolher no ICMC durante passagem acadêmica por São Carlos; ao empresário Claifton do Carmo, por acreditar em mim e apresentar o problema que deu origem a este trabalho; ao gerente operacional da empresa Auto Viação Modelo, Hector Coronado, por me conceder acesso à empresa e fornecer as imagens de monitoramento utilizadas durante esta pesquisa; aos pesquisadores voluntários Gabriel Fonseca, Gustavo Paixão e Igor Lopes, pela prontidão, pelo empenho, pela paciência e pelo excelente trabalho realizado ao longo da preparação das bases de dados construídas nesta pesquisa; à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro no desenvolvimento deste trabalho; ao Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI), pelo fornecimento de recursos computacionais, financiados pela FAPESP (proc. 2013/07375-0);

A todos os citados, meus sinceros agradecimentos.

O único homem que não comete erros é aquele
Que nunca faz coisa alguma.
Não tenha medo de errar,
Pois aprenderá a não cometer
Duas vezes o mesmo erro.
(Theodore Roosevelt)

Resumo

Devido ao recente avanço na área de detecção de objetos, o rastreamento por detecção (no inglês, *tracking-by-detection*) tornou-se o principal paradigma adotado por rastreadores de múltiplos objetos. Com base na extração de diferentes características dos objetos detectados, tais algoritmos são capazes de estimar a similaridade e o padrão de associação dos objetos ao longo de sucessivas imagens. No entanto, uma vez que as funções de similaridade aplicadas por algoritmos de rastreamento são construídas manualmente, sua adaptação para novos cenários é dificultada. Este trabalho investigou o uso de técnicas de aprendizado de máquina baseadas em redes neurais artificiais para a indução automática de funções de similaridade entre objetos. Durante seu treinamento, tais redes foram apresentadas a padrões de associações corretas e incorretas entre detecções de objetos amostradas de bases reais de rastreamento. Para tanto, diferentes características relacionadas à aparência e à movimentação de objetos foram consideradas. Uma rede treinada foi inserida num *framework* de rastreamento de múltiplos objetos, o qual foi avaliado em três diferentes cenários de experimentação: o rastreamento de pedestres, o rastreamento de passageiros de ônibus e a contagem automática de passageiros de ônibus. No primeiro experimento, realizado com base no *benchmark* MOT Challenge, o método proposto obteve acurácia similar à apresentada por algoritmos considerados estado da arte, porém a um custo computacional até 16 vezes menor. Já o segundo experimento foi realizado a partir de uma base de dados construída localmente, sobre a qual o método proposto igualou a acurácia de sua principal *baseline*, o método DeepSORT, porém com ganho de $42,8\%$ em velocidade. O terceiro experimento correspondeu a um estudo de caso no qual a contagem obtida através do método proposto apresentou um erro médio absoluto $40,7\%$ menor que sua *baseline*. Ao fim deste trabalho foi possível verificar que o método proposto pode ser adaptado automaticamente a diferentes cenários e apresenta competitiva relação acurácia *vs* velocidade de execução, quando comparado aos algoritmos considerados nos experimentos.

Palavras-chave: Rastreamento de múltiplos objetos, Visão computacional, Aprendizado de máquina.

Abstract

With the recent advances in the object detection research field, tracking-by-detection has become the leading paradigm adopted by multi-object tracking algorithms. By extracting different features from detected objects, those algorithms can estimate the objects' similarities and association patterns along successive frames. However, since similarity functions applied by tracking algorithms are handcrafted, it is difficult to employ them in new contexts. In this study, it is investigated the use of artificial neural networks to automatically learning a similarity function that can be used among detections. During training, the networks were introduced to correct and incorrect association patterns, sampled from real tracking datasets. For such, different motion and appearance features have been considered. A trained network has been inserted into a multiple-object tracking framework, which has been assessed on three different experiment scenarios: the tracking of pedestrians, the tracking of bus passengers and the automatic counting of bus passengers. During the first, conducted on the MOT Challenge benchmark, the proposed method scored an accuracy similar to state-of-the-art trackers, but at a computational cost at least 16 times lower. In the second experiment, which was based on a local dataset, the proposed tracker matched the results obtained by its direct baseline, the DeepSORT tracker, but with a speed gain of 42.8%. Finally, the third experiment was a study case, where the median absolute error scored by the proposed method was 40.7% inferior to its baseline. In the end, this study could demonstrate that the proposal method can be automatically adapted to different tracking scenarios while presenting highly competitive cost-effectiveness when compared to those algorithms considered in the experiments.

Keywords: Multiple-object tracking, Computer vision, Machine learning.

Lista de ilustrações

Lista de tabelas

Lista de algoritmos

Lista de abreviaturas e siglas

API	*Application Programming Interface* (do inglês, Interface de Programação de Aplicações)
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNN	*Convolutional Neural Network* (do inglês, Rede Neural Convolucional)
CNT	Confederação Nacional do Transporte
FM	*Fragmentation* (do inglês, Fragmentação)
FPS	*Frames Per Second* (do inglês, Quadros Por Segundo)
GMM	*Gaussian Mixture Model* (do inglês, Modelo de Mistura de Gaussianas)
HOG	*Histogram of Oriented Gradient* (do inglês, Histograma de Gradientes Orientados)
IDS	*Identity Switches* (do inglês, Trocas de Identidades)
KCF	*Kernelized Correlation Filter*
mAP	*Mean Average Precision*
ML	*Mostly Lost* (do inglês, Majoritariamente Perdidos)
MLP	*Multilayer Perceptron* (do inglês, *Perceptron* Multicamadas)
MOT	*Multiple Object Tracking* (do inglês, Rastreamento de Múltiplos Objetos)
MOTA	*Multiple Object Tracking Accuracy* (do inglês, Acurácia de Rastreamento de Múltiplos Objetos)
MOTP	*Multiple Object Tracking Precision* (do inglês, Precisão de Rastreamento de Múltiplos Objetos)
MT	*Mostly Tracked* (do inglês, Majoritariamente Rastreados)
NTU	Associação Nacional das Empresas de Transportes Urbanos
PDF	*Probability Density Function* (do inglês, Função de Densidade de Probabilidade)
R-CNN	*Regions with* CNN (do inglês, Regiões com Rede Neural Convolucional)

R-FCN	*Region-based Fully Convolutional Networks*
RGB	Modelo de cores *Red, Green and Blue* (do inglês, Vermelho, Verde e Azul)
ReLU	*Rectified Linear Unit* (do inglês, Unidade Linear Retificada)
RPN	*Region Proposal Network*
SIFT	*Scale-Invariant Feature Transform*
SSD	*Single Shot MultiBox Detector*
SURF	*Speeded-Up Robust Features*
SVM	*Support Vector Machine* (do inglês, Máquina de Vetores de Suporte)
YOLO	*You Only Look Once*

Lista de símbolos

α	Letra grega minúscula alfa
β	Letra grega minúscula beta
δ	Letra grega minúscula delta
τ	Letra grega minúscula tau
$\forall$	Para todos
$\in$	Pertence
$\ni$	Possui ao menos um
$\mathbb{R}$	Conjunto dos números Reais

Sumário

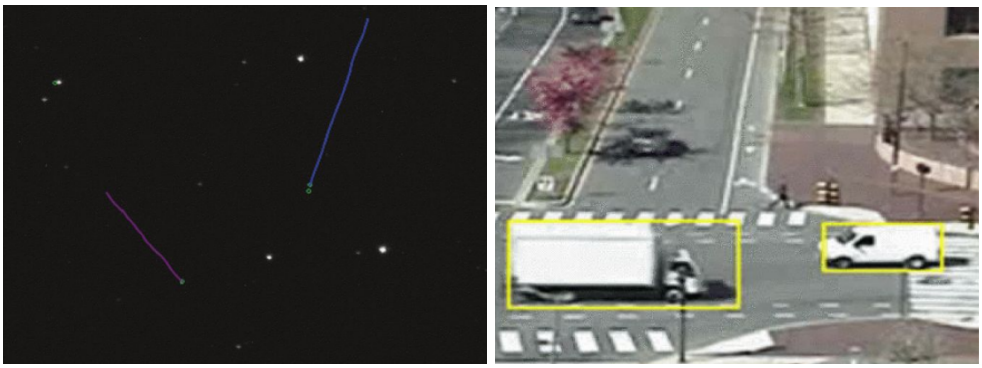
1 Introdução

O rastreamento de múltiplos objetos (do inglês, *Multiple Object Tracking* ou MOT) corresponde a um dos principais problemas investigados na área de Visão Computacional, haja vista a quantidade de pesquisas com carácter multidisciplinar que se baseiam na resolução de tal tarefa. Na área médica, vasos sanguíneos são rastreados com o propósito de mapeamento (IBRAHIM et al., 2017); na área biológica, rastreiam-se células (MENG; SHEN, 2016)(Figura 1a), peixes (FEIJÓ et al., 2018), morcegos (RODRIGUES et al., 2016) e moscas (DEQIN et al., 2016) de modo a estudar seus comportamentos; na agricultura, o MOT está presente na detecção e identificação de espécies de plantações (HAMUDA et al., 2018)(Figura 1b); a astronomia baseia-se no rastreamento para estudar corpos celestes (LEI et al., 2016)(Figura 1c); na área química, o rastreamento de múltiplos objetos é usado para caracterizar as micro-estruturas de materiais compósitos (ZHOU et al., 2016) e interpretar o comportamento coletivo de coloides ativos (WANG et al., 2016). O MOT também está presente na análise de tráfego em cidades inteligentes (MRITHU; FRANCIS, 2016)(Figura 1d), no desenvolvimento de veículos autônomos (CHEN et al., 2017b) e na análise de eventos esportivos (BAYSAL; DUYGULU, 2016). Finalmente, além de contribuir diretamente para o desenvolvimento de diferentes aplicações, o rastreamento de múltiplos objetos corresponde também a uma etapa fundamental para a resolução de diversos outros problemas de Visão Computacional, tais como estimação de pose (ANDRILUKA et al., 2018), reconhecimento de atividades (LIN et al., 2016) e análise comportamental (CHONG et al., 2017).

Figura 1 – Ilustração de aplicações de algoritmos MOT.

(a) Rastreamento de células sanguíneas. (b) Rastreamento de plantações.

(c) Rastreamento de corpos celestes. (d) Rastreamento de veículos.



Fonte: Meng e Shen (2016), Hamuda et al. (2018), Lei et al. (2016), Mrithu e Francis (2016), com alterações do próprio autor.

Além de relevante para diversas áreas, o problema de rastreamento de múltiplos objetos em vídeo também é desafiador. Tal problema envolve identificar e localizar os alvos de interesse, estimar suas trajetórias e manter um registro de suas identidades ao longo de toda a sequência de imagens. Assim como no problema para um único objeto, o MOT apresenta como desafios a detecção de seus alvos e a estimação das suas trajetórias dadas eventuais alterações nas suas escalas, na iluminação do cenário e na movimentação da câmera. No entanto, por lidar com mais de um alvo ao mesmo tempo, o MOT ainda possui desafios adicionais, tais como a oclusão parcial ou completa dos objetos, o gerenciamento do início e fim das trajetórias, o desaparecimento momentâneo de objetos e a troca de identidade entre estes, a qual pode ocorrer tanto pela aparência semelhante dos alvos quanto pela interação entre os mesmos (LUO et al., 2014).

Dessa forma, diferentes técnicas para a resolução desse problema são propostas na literatura (LEAL-TAIXÉ et al., 2017). Devido aos avanços na área de detecção de objetos (REN et al., 2015), as técnicas de rastreamento mais recentes adotam o paradigma conhecido como rastreamento por detecção (no inglês, *tracking-by-detection*). Tal paradigma modela o problema de rastreamento de múltiplos objetos como uma tarefa de associação de detecções contidas em diferentes quadros de um vídeo. Em outras palavras, dada uma sequência de imagens, as técnicas

de MOT baseadas no rastreamento por detecção aplicam um detector sobre cada imagem, de modo a localizar os objetos alvos do rastreamento. Uma vez localizados, busca-se associar as detecções referentes a um mesmo objeto, as quais estão presentes em diferentes imagens. Dessa forma, encontram-se as trajetórias de cada alvo.

A partir de tal paradigma, as técnicas de rastreamento geralmente formulam o problema de associação de detecções como uma tarefa de otimização relacionada a grafos. De acordo com tal formulação, cada vértice do grafo representa uma detecção e cada uma de suas arestas armazena o valor do custo da associação entre detecções. Assim, é comum a aplicação de algoritmos de otimização em redes para associações globais (Li Zhang; Yuan Li; NEVATIA, 2008) e o uso de técnicas de programação linear, como o Método Húngaro, para associações locais (GEIGER et al., 2014). No primeiro caso, o grafo é formado a partir das detecções obtidas ao longo de toda a sequência de imagens processada. Assim, ao aplicar o algoritmo de otimização, encontram-se as trajetórias globais de cada alvo. Já no segundo caso, em geral tem-se um grafo bipartido formado pelas detecções presentes na última imagem visualizada e na imagem atualmente processada. Este último cenário é comum entre os rastreadores *online*, os quais têm como vantagem a possibilidade de serem usados em aplicações de tempo real, como a vigilância e a navegação autônoma, por exemplo (FAN et al., 2016).

De modo a obter o valor do custo das associações, os algoritmos de rastreamento por detecção constroem modelos que descrevem cada um dos objetos detectados. A partir de tais modelos, busca-se extrair alguma medida que aponte a similaridade entre as detecções. Dentre os modelos mais utilizados encontram-se aqueles baseados na aparência dos objetos. Os modelos de aparência mais tradicionais baseiam-se em *templates* de *pixels* puros (ORON; BAR-HILLEL; AVIDAN, 2015) e em histograma de cores (TANG et al., 2016). Já modelos mais recentes contam com características visuais extraídas automaticamente através de arquiteturas de aprendizado profundo baseadas em redes neurais convolucionais (LEAL-TAIXÉ; FERRER; SCHINDLER, 2016). No entanto, devido à possibilidade de alvos com aparência semelhante (como pedestres ou veículos), algumas técnicas de rastreamento também constroem modelos que incorporam outras características além da aparência, como padrões de movimentação (ORON; BAR-HILLE; AVIDAN, 2014) e de interação social (RISTANI; TOMASI, 2015). Finalmente, há técnicas que utilizam simultaneamente mais de um modelo para estimação da similaridade (ALAHI et al., 2016).

Apesar de diferentes características serem utilizadas para elaboração da medida de similaridade entre detecções, percebe-se que grande parte das técnicas descritas na literatura realiza tal cálculo com base em funções construídas manualmente (YOON et al., 2016; BEWLEY et al., 2016; WOJKE; BEWLEY; PAULUS, 2017; BOCHINSKI; EISELEIN; SIKORA, 2017). Muito embora tais funções possam ser ajustadas com base no refinamento de seus parâmetros durante fases de experimentação, pode-se supor que a utilização de métodos automáticos para a elaboração da função de custo permite que as características consideradas sejam exploradas de

maneira mais eficiente no que se refere à discriminação das detecções. De fato, aplicações de técnicas de Aprendizado Profundo visando-se o aprendizado da função de similaridade entre detecções no contexto do rastreamento de múltiplos objetos já foram propostas por diferentes trabalhos (SON et al., 2017; SADEGHIAN; ALAHI; SAVARESE, 2017). No entanto, devido à complexidade dos modelos gerados, a performance em termos de velocidade de rastreamento relatada por seus autores não ultrapassa a taxa de 4 imagens por segundo, de modo a desencorajar a utilização de tais modelos no contexto do rastreamento em tempo real.

Desse modo, percebe-se a oportunidade de exploração de modelos de Aprendizado de Máquina mais simples que aqueles baseados em redes neurais profundas para o cálculo da similaridade entre objetos detectados durante a tarefa de MOT em tempo real. Assim como justificado por trabalhos que exploraram o uso de modelos de Aprendizado Profundo (SON et al., 2017; SADEGHIAN; ALAHI; SAVARESE, 2017), a partir do treinamento com base em associações previamente rotuladas, modelos de Aprendizado de Máquina seriam capazes de aprender a discriminar detecções com fundamento em relações mais complexas e mais robustas do que aquelas expressadas por meio de funções construídas manualmente. Além disso, devido à simplicidade dos modelos pretendidos quando comparados àqueles de arquitetura profunda, seria possível utilizá-los durante o rastreamento *online* de múltiplos objetos em tempo real.

1.1 Hipótese

Acredita-se que, no contexto do rastreamento de múltiplos objetos em tempo real, seja possível induzir um modelo de Aprendizado de Máquina para, a partir de características visuais previamente extraídas, estimar o custo da associação entre a trajetória de um objeto já identificado e uma nova detecção, de maneira a obter resultados de rastreamento melhores que aqueles alcançados com base em funções de custo construídas manualmente e também que aqueles obtidos a partir de modelos de Aprendizado Profundo.

1.2 Objetivos

Este trabalho tem como objetivo geral explorar o uso de modelos de aprendizado de máquina para a análise em tempo real da similaridade entre objetos detectados durante o rastreamento de múltiplos objetos. Já seus objetivos específicos são:

- Induzir um modelo de aprendizado de máquina capaz de estimar o custo da associação entre a trajetória de um objeto já identificado e uma nova detecção;
- Desenvolver um algoritmo de rastreamento de múltiplos objetos em tempo real baseado na utilização do modelo induzido;

- Avaliar o método proposto em três cenários de experimentação, os quais são o rastreamento de pedestres, o rastreamento de passageiros de ônibus e a contagem automática de passageiros de ônibus.

1.3 Estudo de Caso

Este trabalho tem como estudo de caso o rastreamento de pessoas visando-se a contagem automática de passageiros de ônibus. Tal aplicação foi escolhida com base na demanda da empresa de transporte urbano Auto Viação Modelo[1], a qual está sediada na cidade de Aracaju, estado de Sergipe. O principal interesse da empresa em relação ao desenvolvimento deste trabalho está na possibilidade de estimar a incidência de fraudes referentes ao uso do transporte sem o devido pagamento por parte dos passageiros. Desse modo, ao longo deste trabalho utilizaram-se imagens do circuito interno de segurança dos ônibus da empresa.

1.4 Estrutura do trabalho

O restante deste trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta os trabalhos relacionados; o Capítulo 3 discute a fundamentação teórica relacionada ao rastreamento por detecção; o Capítulo 4 apresenta o algoritmo de rastreamento investigado neste trabalho; o Capítulo 5 descreve os experimentos realizados e discute os resultados alcançados; Capítulo 6 descreve o estudo de caso conduzido; o Capítulo 7 apresenta as conclusões deste trabalho.

[1] Mais detalhes acerca da empresa podem ser encontrados através do seu endereço eletrônico: <http://www.viacaomodelo.com.br/institucional.php>

2

Trabalhos Relacionados

De modo a obter os trabalhos relacionados para o trabalho proposto, inicialmente realizou-se uma revisão sistemática da literatura acerca dos algoritmos de rastreamento de múltiplos objetos num vídeo. Em seguida, com base nas respostas às questões de pesquisa, parte dos resultados encontrados foi selecionada para compor a lista de trabalhos relacionados. As seções a seguir descrevem a condução da revisão e da seleção dos trabalhos.

2.1 Revisão Sistemática da Literatura

A revisão conduzida neste trabalho inspirou-se no método apresentado por Kitchenham (2012), o qual tem como principal objetivo possibilitar a obtenção de evidências a respeito de um determinado tópico, de maneira formal, sistemática e reproduzível. As seções a seguir descrevem a metodologia utilizada neste trabalho para a revisão sistemática a respeito de algoritmos de MOT e os resultados obtidos.

2.1.1 Metodologia

Para dar início à aplicação do método de revisão sistemática, inicialmente foram elaboradas as questões de pesquisa e foram selecionados os estudos primários. A seguir será discutido o processo de elaboração das questões, a estratégia de busca utilizada e os critérios de seleção e exclusão considerados.

2.1.1.1 Questões de pesquisa

Com base no objetivo estabelecido para a revisão realizada neste trabalho, foram elaboradas as seguintes questões de pesquisa:

1. Quais são os algoritmos utilizados para rastrear múltiplos objetos presentes num vídeo?

2. Quais as bases usadas para avaliar os algoritmos de rastreamento de múltiplos objetos em vídeo?

3. Quais são as métricas adotadas para avaliar o desempenho dos algoritmos de rastreamento de múltiplos objetos em vídeo?

2.1.1.2 Seleção de fontes

Para responder às questões de pesquisa, foram estabelecidos os seguintes critérios de seleção das bases de consulta:

1. Disponibilidade de consulta de artigos através da *Web*;

2. Presença de mecanismo de busca através de palavras-chave;

3. Garantia de resultados únicos através da busca de um mesmo conjunto de palavras-chave;

4. Possibilidade de *download* dos resultados em formato *.bib*.

A partir dos critérios de seleção de fontes de consulta, foram selecionadas para este trabalho as seguintes bases:

- ACM *Digital Library*;
- IEEE *Xplore*;
- *Science Direct*.

Além do fato de atenderem aos critérios de seleção estabelecidos, tais bases também foram selecionadas por serem consideradas as principais fontes de pesquisa na área de Computação. Aqui vale ressaltar que a base *Springer Link*, apesar de também ser uma fonte relevante para a Computação, não foi selecionada para este trabalho pois a mesma não permite a exportação de todos os seus resultados em formato *.bib*, apenas em *.csv*.

2.1.1.3 Termos de busca

Uma vez eleitas as bases de dados, definiu-se que as consultas seriam realizadas via *web*, com base na presença literal dos termos de busca no resumo de cada resultado. Tal consideração foi motivada por dois fatores: 1) ao restringir a consulta ao resumo de cada resultado impede-se que os mecanismos de busca das diferentes fontes vasculhem, cada um à sua maneira, outros trechos dos resultados (como documento completo, palavras-chave dos autores, palavras-chave da fonte, etc); 2) ao utilizar os termos de busca em seu modo literal impede-se que os mecanismos de busca apliquem variações aos termos de acordo com seus próprios critérios. Dessa forma,

percebe-se que a escolha deste método teve como principal objetivo unificar o processo de busca em todas as fontes consideradas.

Com base no método adotado, foram definidos os seguintes termos de busca e suas variantes:

- Multiple object tracking, multi object tracking, multi-object tracking, MOT;
- Multiple person tracking, multi person tracking, multi-person tracking;
- Multiple people tracking, multi people tracking, multi-people tracking;
- Multiple human tracking, multi human tracking, multi-human tracking;
- Multiple target tracking, multi target tracking, multi-target tracking, multitarget tracking;

Grande parte dos termos selecionados, assim como suas variações, foi adicionada com base na execução de consultas piloto, através das quais foi possível aferir a qualidade dos resultados obtidos utilizando-se tais termos. Como pode-se notar, muitos desses termos referem-se ao rastreamento de pessoas. Tal ocorrência se dá pelo fato de que a maioria dos trabalhos encontrados durante as consultas piloto tem como estudos de caso o rastreamento de pessoas.

Finalmente, a Tabela 1 apresenta as *strings* de busca utilizadas em cada fonte. Vale observar que a *string* utilizada na fonte IEEE *Xplore* possui menos termos que as demais em virtude da própria base limitar o uso de até 15 termos.

Tabela 1 – *Strings* de busca utilizadas durante a revisão sistemática.

Fonte	***String* de Busca**
ACM *Digital Library*	recordAbstract:(+(“multiple object tracking” “multi object tracking” “MOT” “Multi-Object tracking” “multiple person tracking” “multi person tracking” “Multi-Person tracking” “multiple people tracking” “multi people tracking” “Multi-People tracking” “multiple human tracking” “multi human tracking” “Multi-Human tracking” “multiple target tracking” “multi target tracking” “Multi-Target tracking” “multitarget tracking”))
IEEE *Explore*	("Abstract":"multiple object tracking" OR "Abstract":"Multi-Object tracking" OR "Abstract":"MOT" OR "Abstract":"multiple person tracking" OR "Abstract":"Multi-Person tracking" OR "Abstract":"multiple people tracking" OR "Abstract":"Multi-People tracking" OR "Abstract":"multiple human tracking" OR "Abstract":"Multi-Human tracking" OR "Abstract":"multiple target tracking" OR "Abstract":"Multi-Target tracking" OR "Abstract":"multitarget tracking")
Science Direct	abs("multiple object tracking" OR "multi object tracking" OR "MOT" OR "Multi-Object tracking" OR "multiple person tracking" OR "multi person tracking" OR "Multi-Person tracking" OR "multiple people tracking" OR "multi people tracking" OR "Multi-People tracking" OR "multiple human tracking" OR "multi human tracking" OR "Multi-Human tracking" OR "multiple target tracking" OR "multi target tracking" OR "Multi-Target tracking" OR "multitarget tracking")

2.1.1.4 Seleção dos estudos

Para esta revisão, definiu-se que os estudos buscados corresponderiam a artigos resumidos (*short-papers*) e completos (*full-papers*). Para filtrar tais resultados foram adotados os seguintes critérios de inclusão e exclusão:

1. Os artigos devem datar a partir de 2016;
2. Os artigos não devem estar duplicados (*i.e.*, apenas uma única versão deve ser mantida);
3. Os arquivos *.bib* dos artigos devem conter seus respectivos títulos e nomes de autores;
4. Os artigos devem estar escritos em inglês;
5. Os artigos devem estar publicados em sua versão final (*i.e.*, apenas pesquisas já finalizadas devem ser consideradas);
6. Os artigos não devem corresponder a capítulos de livros;
7. Os artigos não devem corresponder a estudos secundários (*i.e., surveys*);
8. Os artigos devem estar relacionados diretamente ao tema *Multiple Object Tracking*;
9. Os artigos devem estar disponíveis na íntegra via portal de periódicos da CAPES;
10. Os artigos devem apresentar algoritmos de rastreamento totalmente automáticos;
11. Os artigos devem contemplar a descrição de algoritmos de rastreamento baseados exclusivamente em visão monocular RGB;
12. Os artigos devem descrever algoritmos que baseiam-se em imagens provindas de uma única câmera;
13. Os artigos devem avaliar os algoritmos propostos por meio de bases de dados públicas;
14. Os artigos devem apresentar resultados quantitativos obtidos após a avaliação do algoritmo proposto;
15. Os artigos devem estar publicados em veículos com conceito CAPES *qualis* A.

A partir dos critérios estabelecidos, definiu-se que a seleção dos estudos ao longo desta revisão ocorreria de acordo com o seguinte protocolo: um único pesquisador aplica a estratégia de busca para a identificação de potenciais estudos primários; uma vez identificados, tais estudos são selecionados pelo mesmo pesquisador através da verificação dos critérios de inclusão e exclusão estabelecidos.

2.1.1.5 Extração de informações

Após a seleção dos estudos primários, definiu-se que seriam extraídas de cada artigo as seguintes informações:

- Solução *online* ou *offline*;
- Modelo de aparência utilizado;
- Modelo de movimentação utilizado;
- Bases de dados utilizadas para experimentação;
- Medidas de qualidade consideradas durante experimentação.

Como já discutido brevemente no Capítulo 1, soluções *online* referem-se a rastreadores baseados na captura de apenas uma nova imagem por iteração, enquanto que soluções *offline* correspondem a rastreadores com acesso imediato a todas as imagens do vídeo processado. Por sua vez, modelos de aparência e de movimentação, tais como serão apresentados com mais detalhes no Capítulo 3, referem-se a ferramentas matemáticas empregadas por algoritmos MOT para descrever o estado de cada objeto rastreado a partir de suas características de aparência e de movimentação, respectivamente. É com base nestas descrições que aqueles algoritmos calculam a similaridade entre objetos e, por consequência, o custo da associação entre suas respectivas detecções. Dessa forma, entendeu-se que a partir das informações listadas anteriormente seria possível responder às questões de pesquisa definidas no início desta revisão.

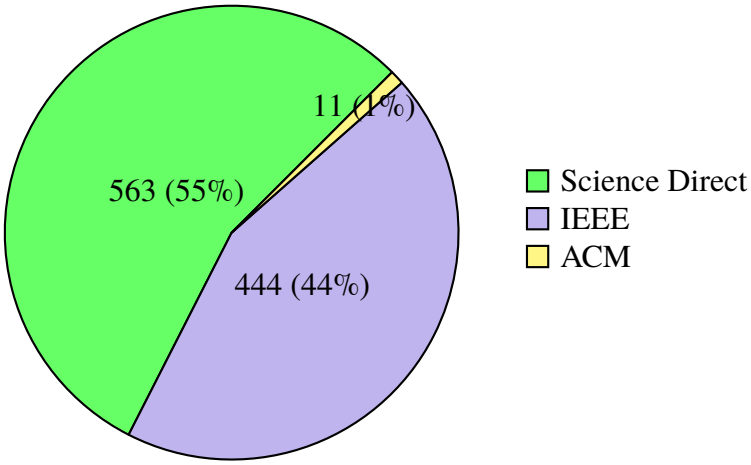
2.1.2 Resultados

Nesta seção serão discutidos os resultados obtidos durante cada etapa da revisão sistemática. Em seguida, as informações extraídas dos artigos selecionados serão utilizadas para a formulação das respostas às questões de pesquisa.

2.1.2.1 Seleção dos Estudos Primários

As consultas sobre as bases de pesquisa foram realizadas no dia 24 de abril de 2018. Vale ressaltar que tais consultas referem-se especificamente à aplicação do protocolo de revisão sistemática utilizado para o levantamento dos principais trabalhos relacionados, de modo que consultas independentes foram realizadas continuamente após tal revisão. Em todas as bases consultadas durante esta revisão utilizou-se o filtro relacionado ao ano mínimo dos resultados. Além disso, na base *Science Direct* utilizou-se o filtro relacionado à remoção de resultados referentes a capítulos de livros. A Figura 2 apresenta a quantidade de resultados obtidos por fonte de pesquisa. Como pode-se notar, a fonte que menos contribuiu para a obtenção de resultados foi a base *ACM*, a qual forneceu apenas 11 artigos.

Figura 2 – Distribuição dos resultados iniciais obtidos após a consulta em cada fonte de pesquisa.

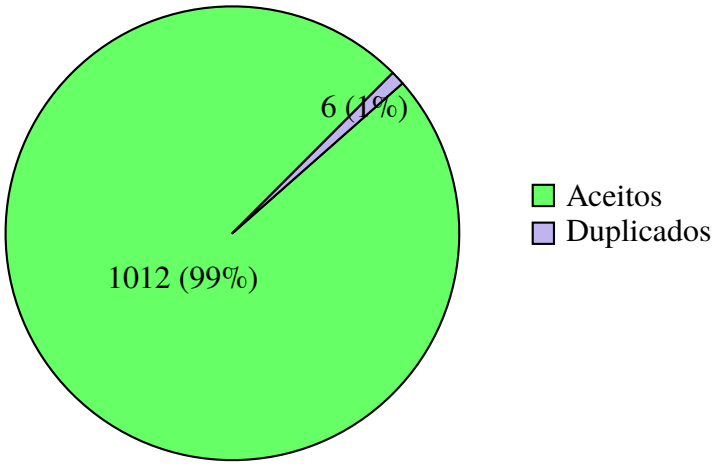


.

Fonte: o próprio autor.

Os artigos obtidos a partir de cada base foram então analisados de acordo com os critérios de inclusão e exclusão definidos para esta revisão. Inicialmente considerou-se apenas o critério de duplicidade. A Figura 3 ilustra a distribuição dos trabalhos analisados nesta etapa.

Figura 3 – Distribuição dos estudos primários aceitos e duplicados.



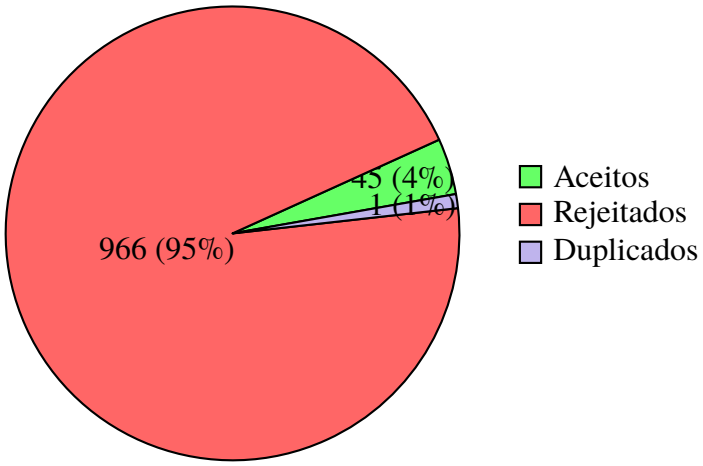
.

Fonte: o próprio autor.

2.1.2.2 Seleção dos Estudos Secundários

Uma vez selecionados, os estudos primários foram filtrados com base nos demais critérios de seleção. Para isso, seu conteúdo foi analisado na íntegra. A Figura 4 ilustra a distribuição dos estudos aceitos e rejeitados após tal análise. Ao fim do processo foram aceitos 45 trabalhos, os quais correspondem a cerca de 4% do total obtido após a seleção dos estudos primários.

Figura 4 – Distribuição dos estudos finais aceitos e rejeitados em relação aos estudos primários.

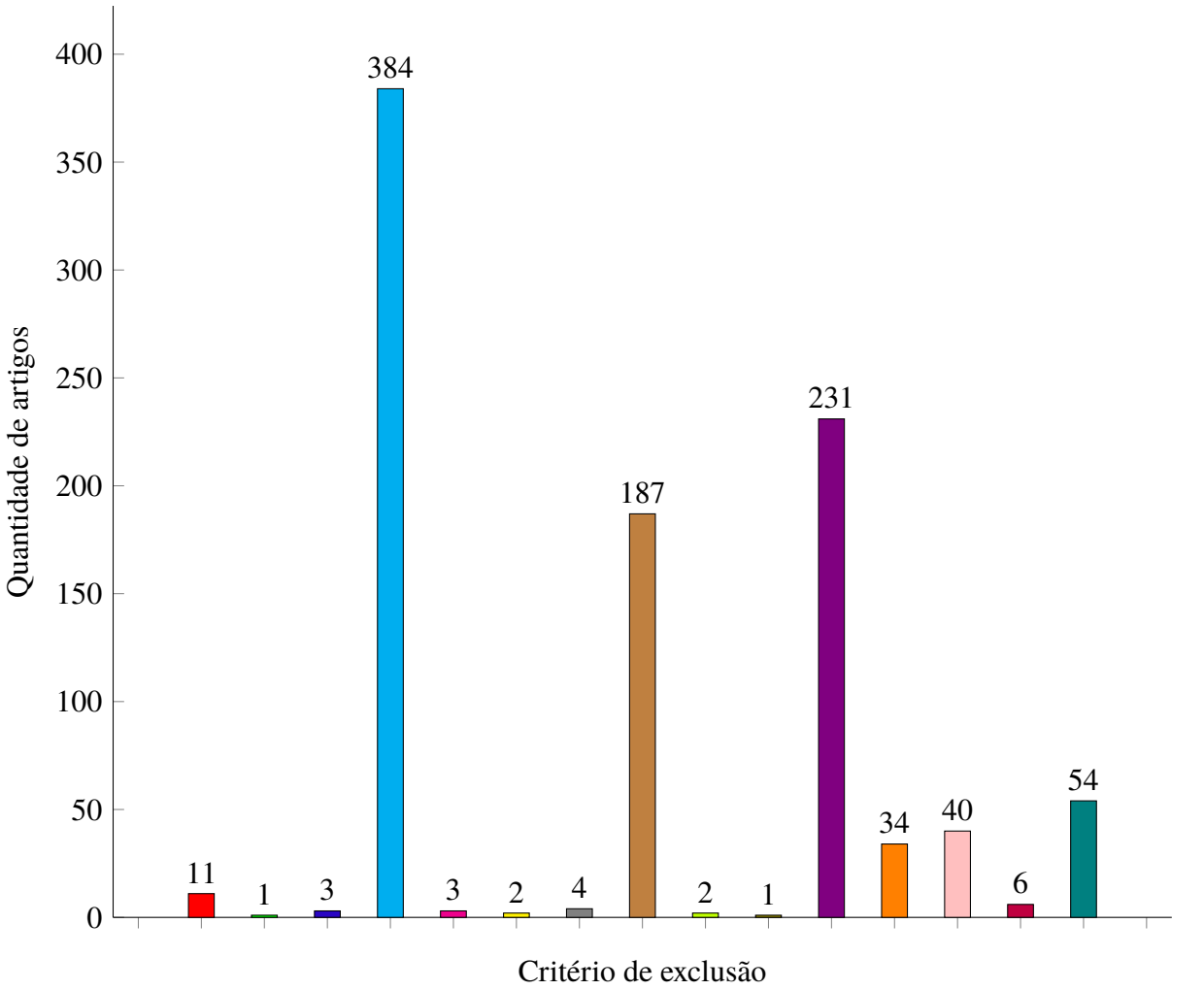


.

Fonte: o próprio autor.

A Figura 5 ilustra a distribuição dos estudos primários rejeitados em relação aos respectivos critérios de exclusão aplicados. Nota-se que a maioria dos estudos foi rejeitada por não estar escrita no idioma inglês (n=384), seguida pela não descrição de algoritmos de rastreamento baseados exclusivamente em visão monocular RGB (n=231), pela não relação direta ao tema MOT (n=187), por não possuírem *qualis* CAPES A (n=54) e por não avaliarem os algoritmos de rastreamento propostos através de bases públicas (n=40).

Figura 5 – Distribuição dos estudos primários rejeitados em relação aos critérios de exclusão aplicados.

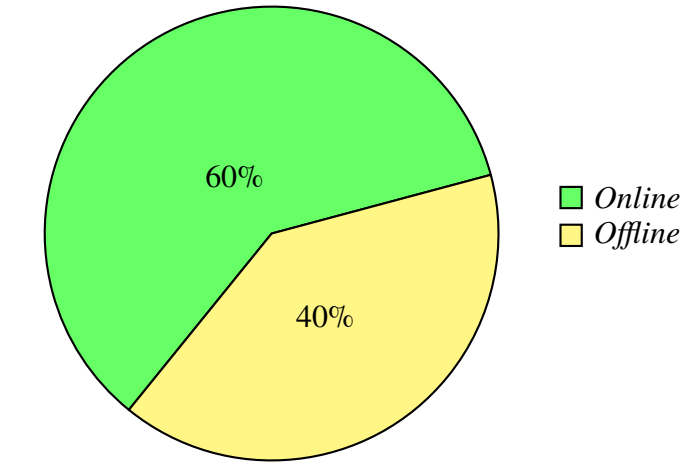


Fonte: o próprio autor.

2.1.2.3 Extração de Dados

Após selecionados, os estudos secundários foram submetidos ao protocolo de extração de informações, de modo a permitir a formulação das respostas às questões de pesquisa estipuladas neste trabalho. A Figura 6 apresenta a distribuição dos algoritmos de rastreamento descritos naqueles estudos com base em seu modo de operação. Percebe-se que 60% dos algoritmos foram projetados para operar de maneira *online*, ou seja, realizam o rastreamento à medida em que novos quadros são capturados. Este resultado já era esperado, haja vista a quantidade significativa de aplicações em tempo real na qual algoritmos de rastreamento *online* podem ser utilizados.

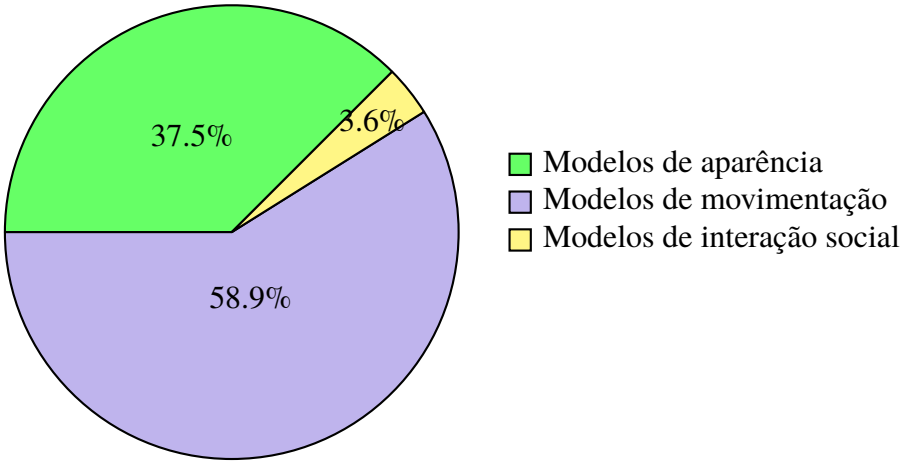
Figura 6 – Distribuição dos algoritmos de rastreamento descritos nos artigos selecionados com base em seu modo de operação.



Fonte: o próprio autor.

Já a Figura 7 apresenta a distribuição dos modelos empregados pelos estudos secundários para descrever objetos durante o rastreamento.

Figura 7 – Distribuição dos modelos empregados pelos estudos secundários.

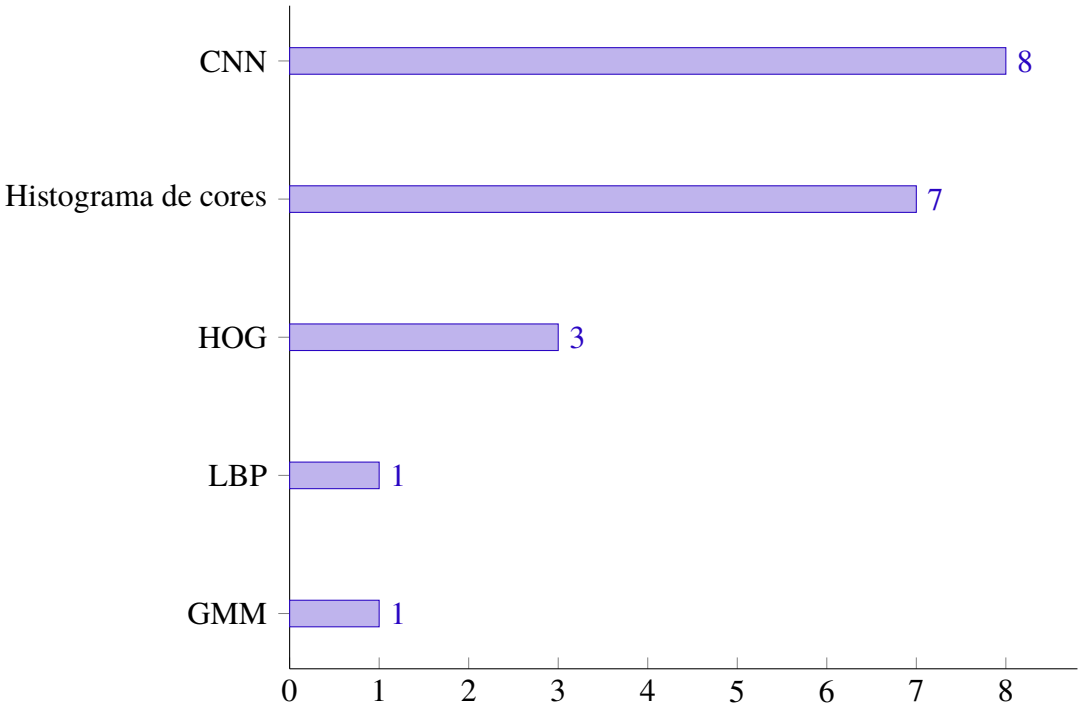


Fonte: o próprio autor.

Com base na Figura 7, percebe-se que 58.9% dos modelos utilizados descreveram a movimentação dos alvos rastreados. Tal constatação também já era esperada, uma vez que no problema de MOT é comum a ocorrência de objetos com aparências semelhantes, de modo que os seus padrões de movimentação tornam-se pistas valiosas para sua discriminação. Já em relação aos modelos de interação social, pode-se supor, com base no conteúdo dos artigos avaliados, que sua utilização foi mínima (n=3.6%) pelo fato da mesma estar relacionada ao rastreamento de multidões (FENG et al., 2017). Finalmente, vale ressaltar que o uso dos diferentes modelos pelos artigos analisados não foi mutualmente exclusivo.

A Figura 8 descreve com mais detalhes os modelos de aparência utilizados pelos artigos selecionados. Esta descrição é realizada em termos dos descritores visuais considerados pelos modelos, quando existentes. Nos casos em que descritores não são considerados, os modelos são referenciados a partir de suas ferramentas estatísticas. É possível perceber que 8 dos modelos de aparência basearam-se em características extraídas por meio de redes neurais convolucionais (no inglês, *Convolutional Neural Networks* ou CNN) (YANG; WU; JIA, 2017), seguidas pelo uso de histogramas de cores (BA et al., 2015), com 7 ocorrências, e por histogramas de gradientes orientados (no inglês, *Histogram of Oriented Gradient* ou HOG) (ULLAH; CHEIKH; IMRAN, 2016), com 3 ocorrências.

Figura 8 – Distribuição dos modelos de aparência utilizados pelos artigos finais em termos de descritores visuais considerados. Modelos que não empregam descritores são referenciados com base em suas ferramentas estatísticas.

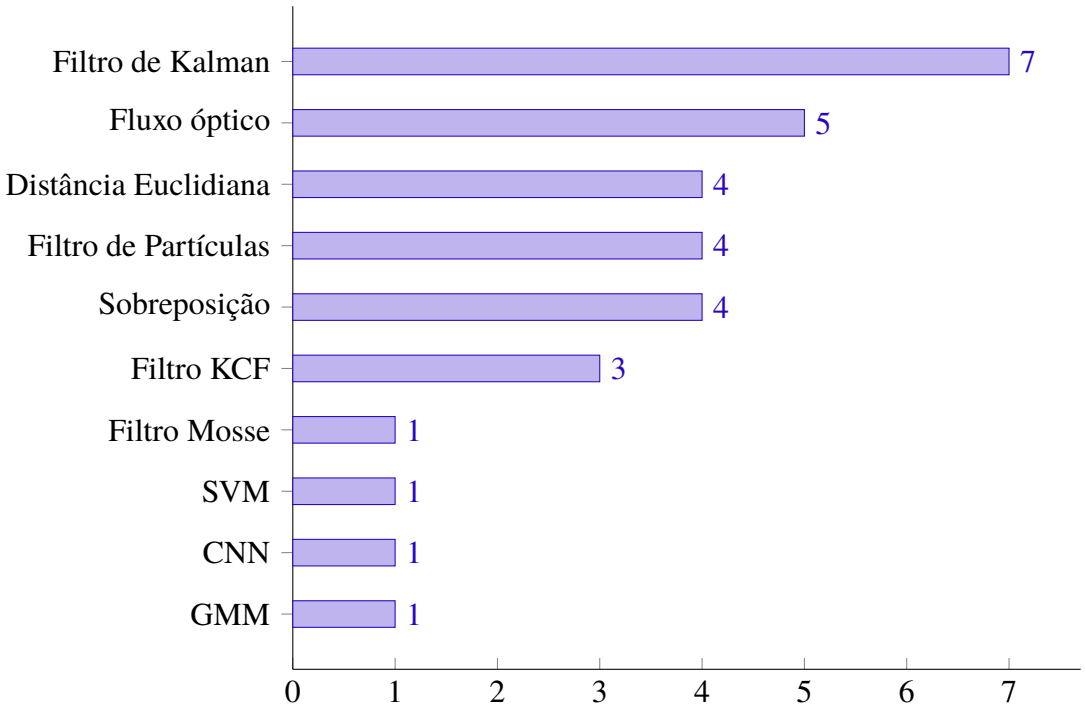


Fonte: o próprio autor.

Já a Figura 9 descreve os modelos de movimentação utilizados pelos artigos selecionados. Neste caso, a descrição é realizada em termos das ferramentas estatísticas aplicadas pelos modelos,

quando existentes. Os modelos que não empregam tais ferramentas são referenciados com base na forma como mensuram movimentação. Pode-se perceber que modelos baseados no Filtro de Kalman (HILKE et al., 2016) e em fluxo óptico (BULLINGER; BODENSTEINER; ARENS, 2017) contabilizaram 7 e 5 ocorrências cada, respectivamente. Já modelos baseados somente na distância euclidiana entre objetos (AL-SHAKARJI et al., 2017) obtiveram 4 ocorrências, juntamente com aqueles baseados em Filtros de Partículas (FU et al., 2017) e na taxa de sobreposição entre objetos (BOCHINSKI; EISELEIN; SIKORA, 2017).

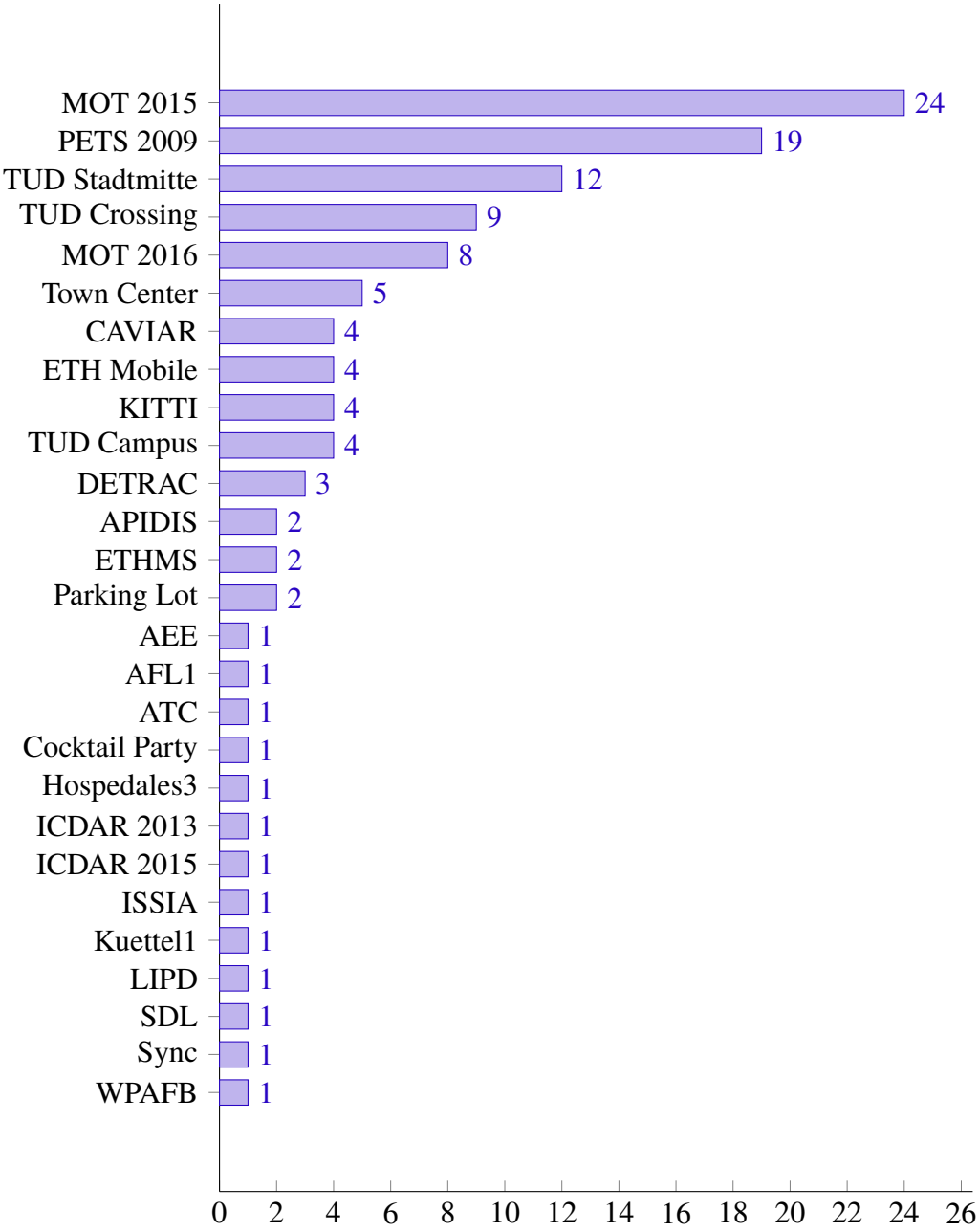
Figura 9 – Distribuição dos modelos de movimentação utilizados pelos artigos finais em termos de ferramentas estatísticas aplicadas. Modelos que não empregam tais ferramentas são referenciados com base na forma como mensuram movimentação.



Fonte: o próprio autor.

Já a Figura 10 apresenta a distribuição das bases de dados utilizadas pelos artigos selecionados para avaliar seus algoritmos de rastreamento. Como pode-se perceber, a base MOT 2015 (LEAL-TAIXÉ et al., 2015) foi a mais utilizada pelos artigos, totalizando 24 ocorrências. Em seguida, foram mais empregadas as bases PETS 2009 (n=19) (ZHANG et al., 2009), TUD Stadtmitte (n=12), TDU Crossing (n=9) (ANDRILUKA; ROTH; SCHIELE, 2008), MOT 2016 (MILAN et al., 2016) (n=8) e Town Center (BENFOLD; REID, 2011) (n=5). Vale notar que estas últimas também compõem as bases MOT 2015 e MOT 2016, as quais referem-se a *benchmarks* formados por bases concedidas publicamente (num esquema de *crowdsourcing*). Além disso, é importante ressaltar que a utilização das bases registradas ao longo dos artigos avaliados não foi mutualmente exclusiva.

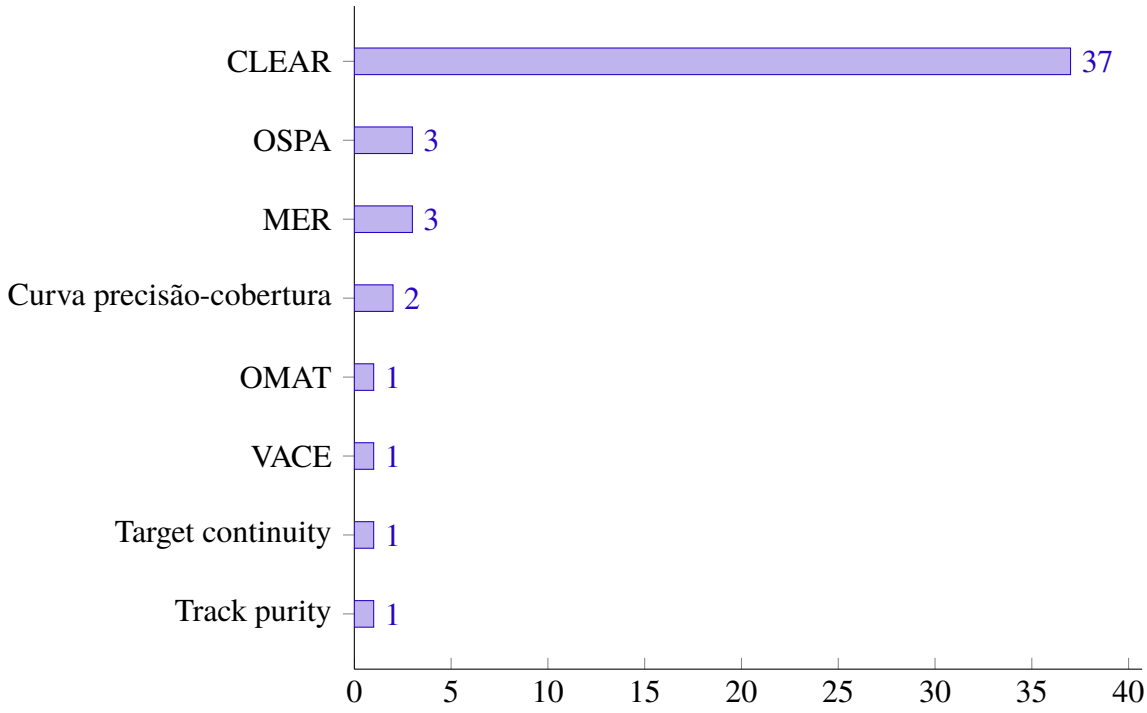
Figura 10 – Distribuição das bases de dados utilizadas pelos artigos selecionados para avaliar seus algoritmos de rastreamento.



Fonte: o próprio autor.

Finalmente, a Figura 11 apresenta a distribuição das medidas de qualidade utilizadas pelos artigos selecionados durante o processo de avaliação de seus rastreadores. O conjunto de medidas CLEAR (BERNARDIN; STIEFELHAGEN, 2008) foi o mais utilizado, totalizando 37 ocorrências. Este conjunto engloba a medida de acurácia total de rastreamento MOTA (no inglês, *Multiple Object Tracking Accuracy*) e a medida de precisão MOTP (no inglês, *Multiple Object Tracking Precision*). Enquanto a primeira avalia o rastreador com base na ocorrência de falsos positivos e falsos negativos, a segunda considera a qualidade do posicionamento e das dimensões estimados pelo rastreador para cada objeto durante cada instante do processo.

Figura 11 – Distribuição das medidas de qualidade utilizadas pelos artigos selecionados para avaliar seus algoritmos de rastreamento.



Fonte: o próprio autor.

2.1.3 Análise

Com base nos resultados apresentados na subseção 2.1.2, é possível responder as questões de pesquisa levantadas no início da revisão sistemática. Com relação à primeira questão, a qual questiona as características dos algoritmos de rastreamento de múltiplos objetos em vídeo, pode-se afirmar que a maior parte dos algoritmos encontrados foi projetada para operar de maneira *online*. Ou seja, de modo a associar as trajetórias de objetos já conhecidas com novas detecções obtidas no instante atual, estes algoritmos utilizam apenas informações referentes a imagens capturadas em instantes passados. Além disso, considerando os trabalhos levantados, pode-se afirmar que a maior parte dos algoritmos propostos associam trajetórias e detecções com base em medidas de similaridade obtidas a partir de características extraídas por meio de redes neurais convolucionais e de modelos de movimentação baseados no Filtro de Kalman.

Já em relação à questão sobre as bases de dados utilizadas para avaliar algoritmos de rastreamento, pode-se afirmar que a principal delas é a base de rastreamento de pedestres MOT 2015. Esta base é composta por 11 sequências de treinamento e 11 sequências de teste, sendo que cada par de sequências treinamento-teste corresponde a um cenário com características diferentes (iluminação, ângulo, movimentação da câmera, resolução, dentre outras). Ao todo, a base conta com 5500 quadros de treinamento e 5783 quadros de teste, totalizando 389 e 607 segundos de vídeo, respectivamente. Para todas as sequências são disponibilizadas as localizações dos objetos a serem rastreados, sendo que os rótulos do rastreamento apenas estão disponíveis

para as sequências de treinamento. Por fim, os gestores da base também disponibilizam uma ferramenta para a avaliação dos algoritmos de rastreamento, a qual calcula, dentre outras medidas, as métricas CLEAR, o número de identidades trocadas e a taxa de processamento.

Finalmente, quanto à última questão de pesquisa, a qual refere-se às métricas de avaliação dos algoritmos de rastreamento, pode-se afirmar que a medida mais utilizada corresponde ao conjunto de métricas CLEAR.

2.2 Seleção dos Trabalhos Relacionados

A partir dos resultados finais obtidos por meio da revisão sistemática realizada, foram selecionados 10 trabalhos relacionados, cujos algoritmos de rastreamento operam de modo *online* e os quais apresentaram performance superior aos demais algoritmos. Para isso, considerou-se o desempenho obtido sobre as bases MOT 2015 e MOT 2016, em termos da medida MOTA e da taxa de processamento por imagem (medida em *hertz*). A Tabela 2 e a Tabela 3 descrevem a performance dos trabalhos relacionados sobre aquelas bases, respectivamente. Além da medida MOTA e da taxa de processamento, estas tabelas também apresentam o número de identidades trocadas (no inglês, *ID switches* ou IDS) pelos algoritmos.

Tabela 2 – Desempenho dos trabalhos relacionados no *benchmark* MOT Challenge 2015. O símbolo * indica algoritmos que utilizaram detectores privados, não fazendo uso das detecções públicas disponibilizadas pelo *benchmark*. Já o símbolo † indica algoritmos cujas execuções basearam-se no uso de placas gráficas.

Referência	Algoritmo	MOTA	IDS	Frequência (Hz)
(BEWLEY et al., 2016)	SORT*	33,4%	1001	**260,5**
(LIN et al., 2017)	HASR	30,5%	602	34,3
(CHEN et al., 2017a)	AP_HWDPL†	**38,5%**	586	6,7
(YANG; JIA, 2015)	TDAM	33,0%	464	5,9
(PARK; LEE; YOON, 2016)	RCMOT_COR	31,1%	-	3,8
(YOON et al., 2016)	SCEA	27,1%	604	6,8
(SHEN et al., 2018)	ATH-MOSSE	23,4%	3728	15
(BORAGULE; JEON, 2017a)	TFMOT	23,8%	**404**	11,3

Tabela 3 – Desempenho dos trabalhos relacionados no *benchmark* MOT Challenge 2016. O símbolo * indica algoritmos que utilizaram detectores privados, não fazendo uso das detecções públicas disponibilizadas pelo *benchmark*. Já o símbolo † indica algoritmos cujas execuções basearam-se no uso de placas gráficas.

Referência	Algoritmo	MOTA	IDS	Frequência (Hz)
(BOCHINSKI; EISELEIN; SIKORA, 2017)	IOU*	57,1%	2167	**3004,6**
(WOJKE; BEWLEY; PAULUS, 2017)	DeepSORT*†	**61,4%**	781	17,4
(BORAGULE; JEON, 2017a)	TFMOT	36,7%	**667**	14,8

Como observado na Tabela 2, o algoritmo SORT (BEWLEY et al., 2016) é o que apresenta a maior taxa de processamento dentre os métodos avaliados sobre a base MOT 2015. A cada iteração, este algoritmo associa as novas detecções realizadas sobre o quadro adquirido no instante atual às trajetórias mantidas até o instante imediatamente anterior. Para isso, utiliza-se o Método Húngaro (KUHN, 2005). Os objetos rastreados são representados a partir de um modelo de movimentação linear baseado no Filtro de Kalman (KALMAN, 1960), cujo estado estimado é composto pela posição e pela dimensão dos objetos relacionados a cada trajetória monitorada. O custo de associação considerado pelo SORT corresponde à taxa de sobreposição entre o estado previsto pelo filtro e cada nova detecção.

O HASR (LIN et al., 2017) se baseia num processo hierárquico de associações em dois níveis. No primeiro nível, o custo da associação entre as trajetórias de objetos já identificados e novas detecções corresponde à taxa de sobreposição entre estas últimas e o futuro estado daqueles objetos, o qual é estimado através do Filtro de Kalman, assim como feito pelo algoritmo SORT. No entanto, durante esta associação apenas são consideradas detecções que satisfazem determinadas restrições de posicionamento e geometria. Tais restrições são descritas por funções construídas manualmente. As trajetórias não associadas são tratadas numa segunda etapa, juntamente com as detecções remanescentes. Nesta nova etapa, o custo da associação entre trajetórias e detecções é calculado a partir de modelos de aparência baseados em representações esparsas.

Já o algoritmo AP_HWDPL (CHEN et al., 2017a) faz uso de um Filtro de Partículas (ARULAMPALAM et al., 2002) para modelar a movimentação de cada objeto rastreado. Através deste filtro, o algoritmo estima as múltiplas regiões do quadro capturado no instante atual nas quais há maior probabilidade de se encontrar os objetos rastreados até o instante anterior. As detecções realizadas sobre tais regiões são então selecionadas como candidatas para possíveis associações com as trajetórias daqueles objetos. Em seguida, o custo destas possíveis associações é calculado pelo algoritmo AP_HWDPL com base num modelo de aparência, o qual é baseado em características visuais extraídas de diferentes camadas de uma CNN.

O método TDAM (YANG; JIA, 2015) realiza a associação entre trajetórias de objetos já identificados e novas detecções com base em três diferentes custos: um referente à distância entre o centro geométrico dos objetos e das detecções, um correspondente à diferença entre suas dimensões e um relacionada às características visuais dos objetos e das regiões da imagem delimitadas pelas detecções. Este último custo é computado por meio de um modelo baseado na Cadeia Oculta de Markov (RABINER; JUANG, 1986), o qual leva em consideração a variação do histograma de gradientes orientados de cada objeto rastreado ao longo do tempo. Ao fim de cada associação, utilizam-se as características visuais relacionadas à cada detecção para atualizar os parâmetros do modelo de aparência.

Por sua vez, o algoritmo RCMOT_COR (PARK; LEE; YOON, 2016) considera dois custos de associação entre novas detecções e trajetórias de objetos já identificados. O primeiro é calculado com base na velocidade relativa dos objetos, enquanto que o segundo é computado a

partir de um modelo de aparência baseado num filtro de correlação. Após associar detecções a trajetórias ao longo de uma etapa local, este algoritmo também realiza a associação entre múltiplas trajetórias já identificadas, por meio de uma etapa global. Para isso, considera-se uma medida de confiabilidade das trajetórias, a qual é influenciada por seus comprimentos e pelo número de detecções já associadas às mesmas.

Em contrapartida, o método SCEA (YOON et al., 2016) associa trajetórias de objetos rastreados e novas detecções com base em dois custos: o primeiro está relacionado à distância entre as dimensões geométricas dos objetos e das detecções. Já o segundo é computado a partir de seus respectivos histogramas de cores. Além disso, o algoritmo também considera para o cálculo do custo a utilização de restrições referentes à movimentação relativa dos objetos ao longo do vídeo.

Já o algoritmo ATH-MOSSE (SHEN et al., 2018) considera como custo de associação entre trajetórias e novas detecções a similaridade entre a aparência dos objetos relacionados àquelas trajetórias e as características visuais das regiões da imagem delimitadas por aquelas detecções. Este custo é computado com base na aplicação do Filtro MOSSE (HENRIQUES et al., 2014). Além disso, este algoritmo também considera como custo de associação a taxa de sobreposição entre objetos rastreados e novas detecções.

De maneira similar, o método TFMOT (BORAGULE; JEON, 2017b) associa novas detecções a trajetórias de objetos rastreados com base em mais de um custo: o primeiro refere-se à distância euclidiana entre os centros geométricos de detecções e objetos, o segundo corresponde à distância entre suas dimensões, já o terceiro está relacionado à diferença entre seus histogramas de cores. Além disso, o algoritmo também aplica restrições quanto à distância euclidiana entre objetos e detecções, de modo a impedir a ocorrência de associações impraticáveis.

Por sua vez, o algoritmo IOU (BOCHINSKI; EISELEIN; SIKORA, 2017) considera apenas um custo de associação, o qual corresponde à taxa de sobreposição entre objetos já rastreados e novas detecções. Além disso, por também não basear-se na aplicação de filtros, este rastreador apresenta a maior taxa de processamento dentre os algoritmos listados na Tabela 2 e na Tabela 3.

Por fim, o algoritmo DeepSORT (WOJKE; BEWLEY; PAULUS, 2017) corresponde a uma atualização do método SORT, na qual foram adicionadas duas novas funções de custo de associação. A primeira delas corresponde à distância de Mahalanobis (JOSEPH; GALEANO; LILLO, 2013) entre as características geométricas de cada nova detecção e aquelas projetadas pelo Filtro de Kalman com base nos objetos já rastreados. Por sua vez, a segunda função refere-se à distância do cosseno entre os descritores referentes à aparência dos objetos detectados e aqueles já monitorados. Estes descritores são obtidos por meio de uma CNN.

Dado o custo-benefício do método DeepSORT, observado a partir da Tabela 2 e da Tabela 3, tal método foi definido como a principal referência deste trabalho. Assim, o mesmo foi

utilizado como *baseline* durante todos os experimentos realizados ao longo deste trabalho.

2.2.1 Discussão

Ao analisar os trabalhos relacionados é possível perceber que muito embora os algoritmos propostos utilizem diferentes características para descrever tanto objetos detectados quanto aqueles já rastreados, o custo final das associações é computado a partir de funções construídas manualmente. Assim, pode-se questionar a fidelidade das relações expressas por tais funções em relação ao comportamento observado dos objetos num cenário real de rastreamento.

Por exemplo, a função de custo do algoritmo DeepSORT corresponde a uma soma ponderada entre a distância de Mahalanobis referente à geometria dos objetos detectados e daqueles já rastreados, e a distância do cosseno entre os vetores que descrevem suas aparências. Com base em tal função surgem as seguintes questões concernentes à associação entre objetos: até que ponto a diferença entre a aparência de dois objetos é mais relevante que a distância entre seus centros geométricos, suas escalas ou suas proporções? A relevância de cada característica não apresenta nenhuma correlação com o histórico de associações de um determinado objeto? As características de um determinado objeto não contribuem de maneiras diferentes a depender do tempo em que o mesmo encontra-se oculto?

Com base em tais questões, percebe-se que os algoritmos de rastreamento apresentados nos trabalhos relacionados dão margem para a exploração de funções de custo de associação mais complexas, as quais expressem relações mais robustas no que tange o comportamento real apresentado por objetos num cenário de rastreamento. Mais ainda, nota-se a possibilidade da utilização de técnicas de Aprendizado de Máquina para a construção de tal função de custo diretamente a partir de dados de trajetórias extraídos de cenários reais de rastreamento. Assim, não apenas vislumbra-se a possibilidade de aumentar a qualidade das associações realizadas, como também de adequar o mesmo algoritmo de rastreamento para operar em diferentes cenários.

3

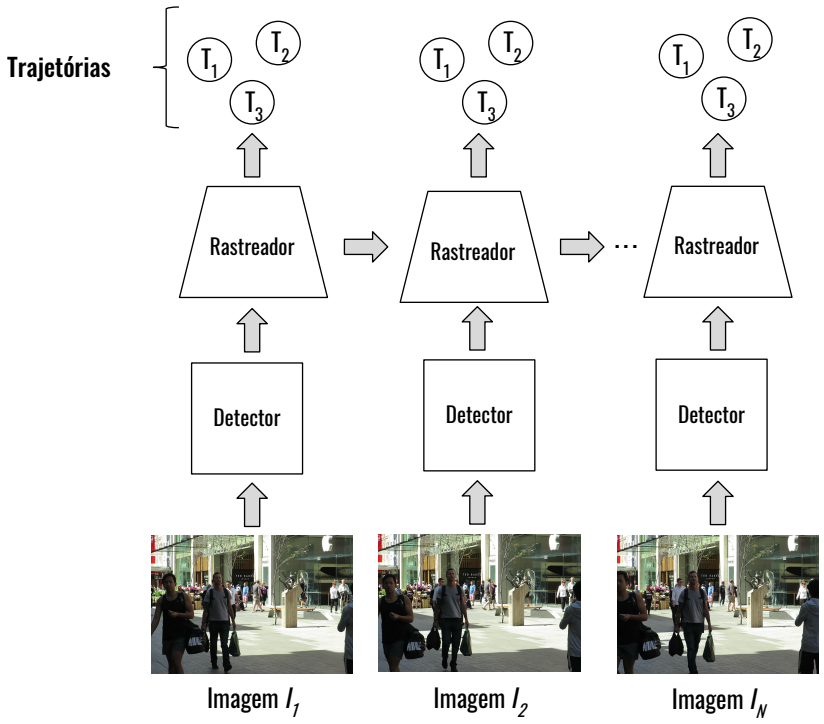
Rastreamento de Múltiplos Objetos por Detecção

Os trabalhos levantados ao final da revisão sistemática discutida no Capítulo 2 ressaltam uma tendência comum aos rastreadores de múltiplos objetos recentemente propostos na literatura: trata-se da aplicação do paradigma *tracking-by-detection* (do inglês, rastreamento por detecção) (LEAL-TAIXÉ et al., 2017). Como seu próprio nome sugere, este paradigma propõe o rastreamento de objetos a partir da aplicação contínua de detectores. Mais precisamente, dada uma sequência de imagens $\{I_1, ..., I_N\}$ e um conjunto de detecções $\{D_t \mid (\forall t)(t \in \{1, ..., N\} \text{ e } D_t = \{d_1^t, d_2^t, ..., d_i^t, ...\})\}$ respectivamente realizadas sobre cada imagem $I_t \in \{I_1, ..., I_N\}$, o rastreador deve gerar um conjunto de trajetórias $\{T_1, T_2, ..., T_i, ...\}$, onde $T_i = \{d_i^1, ..., d_i^N\}$ corresponde a uma sequência ordenada de detecções relacionadas ao objeto o_i. Cada detecção $d_i^t = [u_i^t, v_i^t, w_i^t, h_i^t, c_i^t]$ é definida como uma delimitação retangular sobre a imagem I_t com centro geométrico (u_i^t, v_i^t), largura e altura (w_i^t, h_i^t) e confiança c_i^t. Assim, por meio da trajetória T_i é possível estimar o estado do objeto o_i ao longo de toda a sequência $\{I_1, ..., I_N\}$ (SINGH; RAJAN; S., 2017).

A Figura 12 ilustra o funcionamento do paradigma *tracking-by-detection* num cenário de rastreamento *online*, ou seja, no qual o rastreador apenas tem acesso a imagens capturadas até o instante da iteração atual. Inicialmente um detector de objetos é aplicado sobre a imagem I_t obtida no instante t. O rastreador utiliza as detecções retornadas $\{d_1^t, d_2^t, ..., d_i^t, ...\}$ para modelar os objetos rastreados $\{o_1, o_2, ..., o_i, ...\}$ e inicializar suas respectivas trajetórias $\{T_1, T_2, ..., T_i, ...\}$. No instante subsequente $t + 1$, o mesmo detector é aplicado sobre a imagem I_{t+1}. As novas detecções realizadas $\{d_1^{t+1}, d_2^{t+1}, ..., d_i^{t+1}, ...\}$, juntamente com os modelos previamente obtidos, são utilizados pelo rastreador para atualizar o seu conjunto de trajetórias $\{T_1, T_2, ..., T_i, ...\}$. Todo esse procedimento é repetido até que as N imagens da sequência sejam processadas. Nota-se que através desde paradigma o rastreamento de múltiplos objetos pode ser modelado como a associação de detecções obtidas em instantes subsequentes: para cada nova detecção d_j^{k+1} obtida no instante $k + 1$ busca-se indicar o conjunto $T_i = \{d_i^1, ..., d_i^k\}$ cuja união $T_i \cup \{d_j^{k+1}\}$ estime da

melhor forma possível a trajetória do objeto o_i ao longo da sequência $\{I_1, ..., I_{k+1}\}$.

Figura 12 – Ilustração do paradigma de rastreamento *tracking-by-detection*.



Fonte: o próprio autor.

Com base na descrição realizada até aqui, percebe-se que a detecção de objetos corresponde a uma tarefa fundamental para a execução do paradigma *tracking-by-detection*. Ainda assim, outras etapas tão importantes quanto devem ser performadas, de maneira que ao final de cada iteração as trajetórias estimadas para cada objeto rastreado sejam atualizadas apropriadamente. Ao todo, as tarefas executadas durante cada iteração deste paradigma num instante $t + 1$ podem ser enumeradas como:

1. Detecção de objetos: obtêm-se um novo conjunto de detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$ referentes aos objetos $\{o_1, o_2, ..., o_j, ...\}$ localizados na imagem I_{t+1} atualmente processada;

2. Modelagem de objetos: atualizam-se os modelos dos objetos $\{o_1, o_2, ..., o_i, ...\}$ a partir de suas respectivas detecções $\{d_1^t, d_2^t, ..., d_i^t, ...\}$ realizadas no instante anterior t;

3. Discriminação de objetos: comparam-se os estados dos objetos $\{o_1, o_2, ..., o_i, ...\}$ estimados através de seus respectivos modelos às detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$ referentes aos objetos $\{o_1, o_2, ..., o_j, ...\}$;

4. Associação de detecções: atualizam-se as trajetórias $\{T_1, T_2, ..., T_i, ...\}$ a partir das detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1} ...\}$ mais similares aos estados dos objetos $\{o_1, o_2, ..., o_i, ...\}$ estimados por meio de seus respectivos modelos.

Como observado no Capítulo 2, a literatura especializada propõe uma extensa e diversificada coleção de estratégias e algoritmos destinados especificamente à resolução de cada uma das etapas envolvidas na execução do paradigma *tracking-by-detection*. Assim, de modo a proporcionar um maior entendimento acerca de tal paradigma, as subseções que se seguem discorrem sobre as principais abordagens utilizadas para a resolução de cada uma de suas etapas.

3.1 Detecção de Objetos

A detecção de objetos corresponde a um dos principais e mais clássicos problemas investigados dentro da área de Visão Computacional, não apenas pela sua complexidade mas também por sua relevância para a realização de demais tarefas relacionadas à visão: desde a segmentação, a estimação de pose e inclusive o rastreamento, diversos algoritmos baseiam-se na detecção de objetos como etapa preliminar essencial para sua execução (LIU et al., 2018).

De modo geral, pode-se afirmar que, dada uma imagem $I(x, y)$, a tarefa de detectar objetos consiste em apontar, com confiança $c \in [0, 1]$, a localização (u, v) e as dimensões espaciais (w, h) de instâncias de objetos pertencentes a um conjunto $L = \{l_1, ..., l_N\}$ de N categorias previamente conhecidas (*e.g.* ser humano, carro, mesa) ou a inexistência de tais instâncias. A saída de um detector corresponde, portanto, ao conjunto $D = \{d_1, d_2, ..., d_i, ...\}$, sendo $d_i = [u, v, w, h, c, l_j]$ a detecção referente ao i-ésimo objeto localizado na imagem $I(x, y)$ com categoria l_j. Nota-se que tal tarefa diferencia-se da mera classificação, a qual limita-se a atribuir uma única categoria à imagem por inteiro com base em seu conteúdo principal. A Figura 13 ilustra a diferença entre tais tarefas.

Figura 13 – Comparação entre as tarefas de classificação de imagem e detecção de objetos. Enquanto a primeira consiste em atribuir uma única categoria à imagem com base em seu conteúdo principal, a segunda visa apontar a localização, as dimensões espaciais e a categoria de diferentes objetos contidos na imagem.

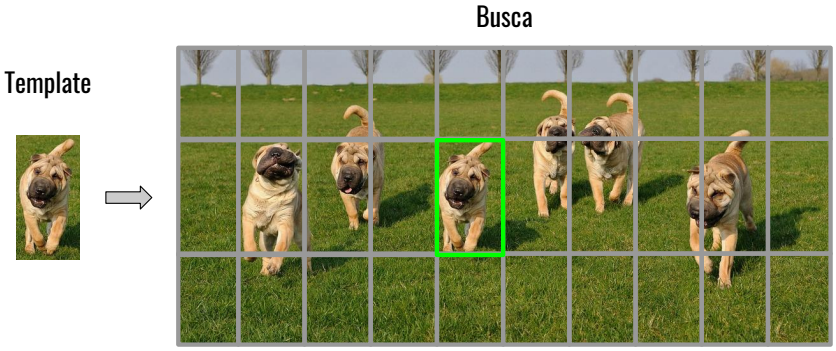
(a) Classificação de imagem.

(b) Detecção de objetos.

Fonte: o próprio autor.

Apesar de distintas, pode-se entender a tarefa de detecção como sendo uma extensão do problema de classificação, na medida em que é possível aplicar um mesmo classificador sobre diferentes regiões de uma imagem, de modo a identificar a localização dos objetos de interesse. De fato, é dessa maneira que grande parte dos algoritmos de detecção de objetos propostos na literatura solucionam tal problema. Ainda em seus primórdios, tais algoritmos baseavam-se na sobreposição e no alinhamento de características simples previamente extraídas dos objetos a serem detectados (Figura 14). Tais características, as quais eram determinadas manualmente, envolviam bordas, *key-points* e *templates* (AGARWAL; TERRAIL; JURIE, 2018). Já a partir da década de 1990, modelos de classificação baseados em técnicas de Aprendizado de Máquina passaram a integrar os detectores, de modo que tornou-se possível utilizar características mais elaborados. É o caso dos métodos baseados em redes neurais (ROWLEY; BALUJA; KANADE, 1996), Adaboost (VIOLA; JONES, 2001) e SVM (DALAL; TRIGGS, 2005).

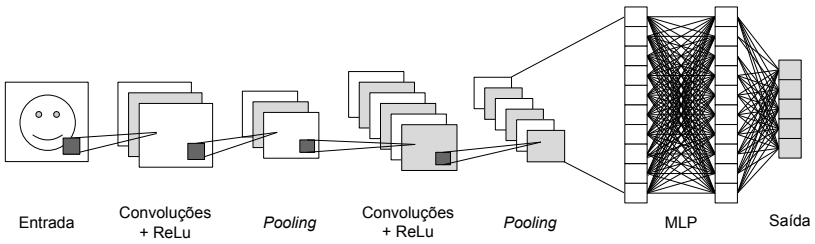
Figura 14 – Ilustração de algoritmo de detecção de objetos baseado em sobreposição de *template*.



Fonte: o próprio autor.

No entanto, foi a partir de 2012 que iniciou-se uma significativa transformação na área de Visão Computacional com a publicação do trabalho realizado por Krizhevsky, Sutskever e Hinton (2012). Este trabalho demonstrou com resultados práticos o potencial das arquiteturas profundas baseadas em redes neurais convolucionais (do inglês, *convolutional neural networks* ou CNN) para a resolução de problemas relacionados à visão. Mais especificamente, aquele trabalho propôs um classificador de imagens baseado numa CNN, o qual pôde ser aplicado diretamente nas imagens de entrada sem a necessidade da aplicação de extratores de características externos. A performance deste classificador sobre o *benchmark* ImageNet superou a dos demais métodos propostos até então, o que destacou a capacidade das CNN em modelar características visuais de modo fim-a-fim com base no aprendizado supervisionado. A partir de então, o uso de tais arquiteturas profundas tornou-se tendência não só para a classificação de imagens (Figura 15), mas para toda a área de Visão Computacional.

Figura 15 – Ilustração da arquitetura de um classificador genérico baseado numa CNN.



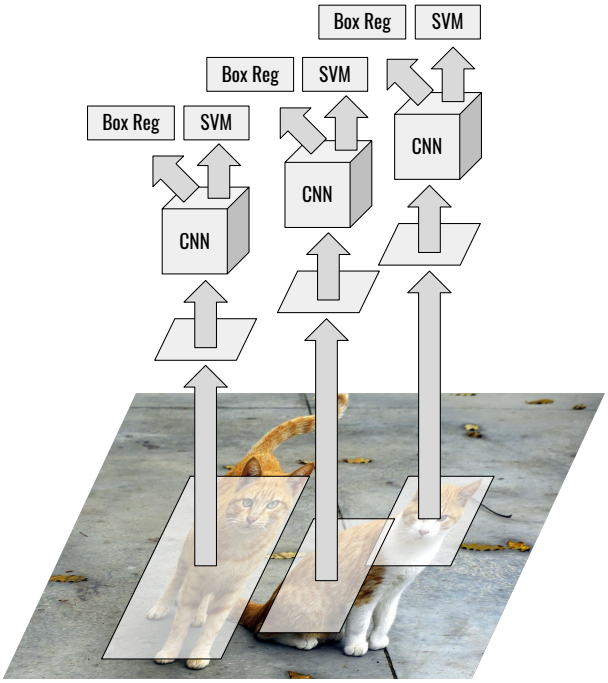
Fonte: o próprio autor.

Os resultados promissores alcançados na área de classificação de imagens inspiraram o desenvolvimento de algoritmos de detecção baseados em CNN. Muito embora uma discussão mais aprofundada a respeito de tais arquiteturas esteja além do escopo deste trabalho, uma explicação sobre os principais aspectos destes detectores será abordada ao longo da subseção a seguir.

3.1.1 Detectores baseados em CNN

Dentre os primeiros detectores baseados em CNN encontram-se os do tipo *region based* (do inglês, baseado em regiões). Estes tiveram origem com o detector R-CNN desenvolvido por Girshick et al. (2013), o qual é ilustrado na Figura 16. Assim como métodos de detecção mais tradicionais, o R-CNN realiza a detecção de objetos com base na classificação de diferentes regiões da imagem. De modo a selecionar tais regiões, é aplicado sobre a imagem o método de busca seletiva (UIJLINGS et al., 2013). Este método baseia-se no agrupamento hierárquico de regiões visualmente similares em termos de cores, texturas, formas e tamanhos, sendo tal agrupamento realizado através de um algoritmo de super-segmentação (FELZENSZWALB; HUTTENLOCHER, 2004). As sub-imagens referentes a cada região obtida pelo método de busca seletiva são então reescaladas e introduzidas numa CNN. Esta, por sua vez, é responsável por extrair características visuais relevantes, as quais são finalmente apresentadas a um classificador SVM e a um regressor. O primeiro tem como função apontar a classe do objeto supostamente contido na região analisada. Já o segundo exerce o papel de ajuste das dimensões da região proposta, visando-se corrigir possíveis falhas do algoritmo de busca seletiva.

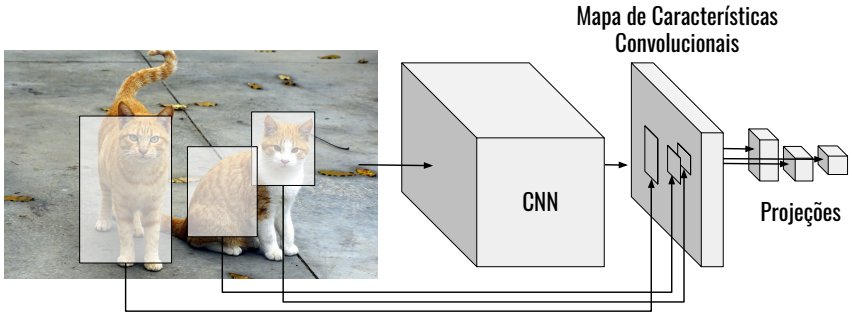
Figura 16 – Ilustração do detector R-CNN (GIRSHICK et al., 2013). Após a aplicação do método de busca seletiva (UIJLINGS et al., 2013), as regiões encontradas são reescaladas e apresentadas a uma CNN. Esta extrai de cada sub-imagem características relevantes que são apresentadas a um classificador SVM e a um regressor. O primeiro é responsável por indicar a classe do possível objeto contido na região em questão, enquanto que o segundo corrige as dimensões da região proposta pelo algoritmo de busca seletiva.



Fonte: o próprio autor.

Como pode-se notar, o método R-CNN apresenta como principais gargalos a aplicação de uma mesma CNN sobre diferentes regiões da imagem analisada, o que o impede de ser utilizado em aplicações de tempo real. Além disso, o algoritmo de busca seletiva empregado pelo R-CNN não apresenta nenhuma capacidade de aprendizado, o que limita a qualidade das regiões propostas. Com o objetivo de contornar tais gargalos, outros métodos de detecção do tipo *region based* foram desenvolvidos. Inspirado diretamente no R-CNN, o método *Fast* R-CNN (GIRSHICK, 2015) propõe a aplicação de toda a imagem numa mesma CNN uma única vez, de modo a gerar um mapa de características convolucionais. Este mapa corresponde a uma imagem com múltiplos canais obtida após a convolução de diversos filtros espaciais sobre a imagem de entrada da rede. Sobre este mapa projetam-se as regiões propostas pelo método de busca seletiva. Com base em tais regiões, extraem-se sub-mapas, os quais são reescalados e apresentados a um classificador e a um regressor. Estes são utilizados de maneira semelhante ao processo realizado pelo método R-CNN. A Figura 17 ilustra o comportamento de tal detector.

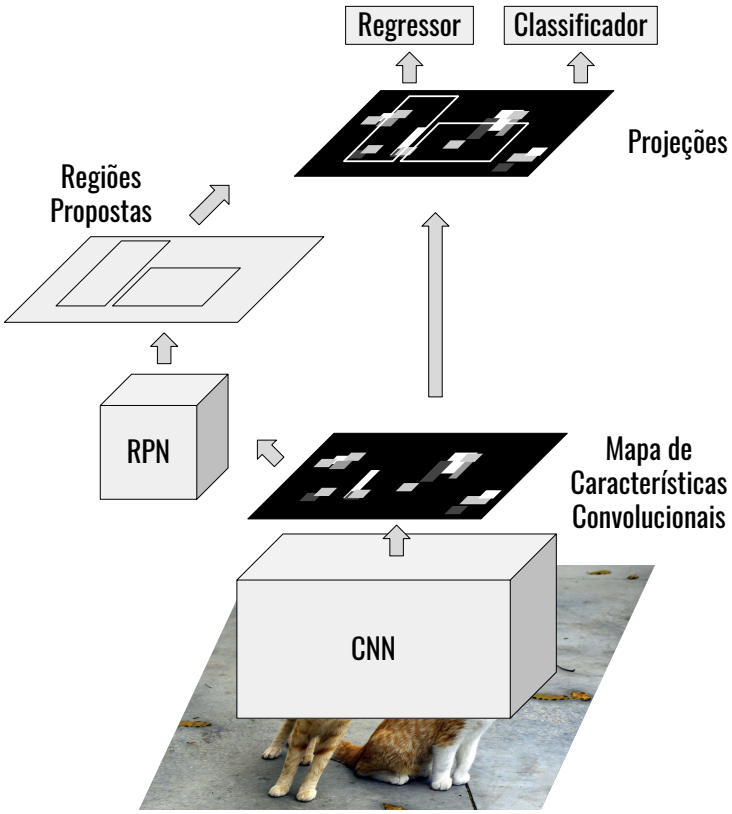
Figura 17 – Ilustração do detector Fast R-CNN (GIRSHICK, 2015). Toda a imagem é apresentada a uma única CNN, de modo a extrair-se um mapa de características convolucionais. Em seguida, as regiões de interesse obtidas por meio da execução do método de busca seletiva sobre a imagem original são projetadas sobre tal mapa. De maneira semelhante ao procedimento seguido pelo R-CNN, cada projeção é apresentada a um classificador e a um regressor, os quais são responsáveis, respectivamente, por apontar a classe do possível objeto contido na região em questão e ajustar as dimensões da região proposta pelo algoritmo de busca seletiva.



Fonte: o próprio autor.

Tanto o detector R-CNN quanto seu sucessor Fast R-CNN baseiam-se na aplicação do método de busca seletiva para a obtenção das regiões de interesse a serem analisadas. Como afirmado anteriormente, tal método não possui capacidade de aprendizado, o que limita a qualidade das regiões propostas e, consequentemente, a das detecções finais. De modo a superar tal gargalo, uma segunda extensão do R-CNN, o algoritmo Faster R-CNN (REN et al., 2015), substitui o método de busca seletiva pela aplicação de uma CNN projetada exclusivamente para a proposição das regiões de interesse. A Figura 18 ilustra o comportamento de tal detector. Assim como feito pelo Fast R-CNN, inicialmente aplica-se uma única CNN sobre toda a imagem de entrada, de modo a obter um mapa de características convolucionais. Em seguida, tal mapa é apresentado a uma CNN denominada RPN (do inglês, *Region Proposal Network*), a qual tem como objetivo propor as regiões nas quais há maior probabilidade de encontrar objetos de interesse. Uma vez propostas, tais regiões são projetadas sobre o mapa de características convolucionais. Finalmente, assim como nos métodos do tipo *region based* discutidos anteriormente, as projeções obtidas são apresentadas a um classificador e a um regressor.

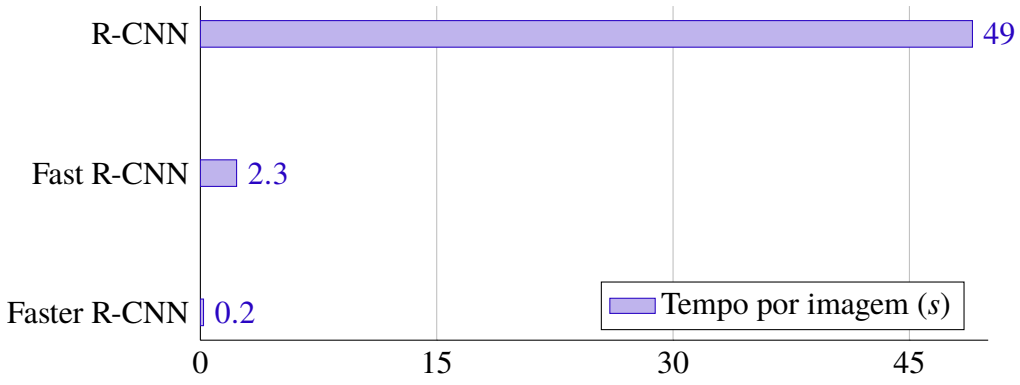
Figura 18 – Ilustração do detector Faster R-CNN (REN et al., 2015). Inicialmente, um mapa de características convolucionais é extraído da imagem de entrada por meio de uma CNN. Em seguida, tal mapa é apresentado à rede RPN, a qual propõe as regiões da imagem onde há maior chance de encontrarem-se objetos. As regiões propostas são projetadas sobre o mapa de características. Finalmente, tais projeções são apresentadas a um classificador e a um regressor, de modo semelhante ao realizado pelos métodos R-CNN e Fast R-CNN.



Fonte: o próprio autor.

A evolução dos métodos de detecção do tipo *region based* discutidos até aqui pode ser analisada com base em seus respectivos tempos de execução, os quais são apresentados na Figura 19. Percebe-se um ganho significativo em performance à medida em que a maior parte da complexidade dos detectores é repassada para as redes convolucionais, como no caso da extração e compartilhamento de características profundas (Fast R-CNN) e a proposição de regiões de interesse (Faster R-CNN). Tal ganho evidencia novamente a capacidade daquelas redes em extrair e modelar características visuais a partir de aprendizado supervisionado, além de demonstrar que tal capacidade pode ser estendida para tarefas que vão além da classificação.

Figura 19 – Tempo de inferência por imagem dos detectores R-CNN, Fast R-CNN e Faster R-CNN sobre a base de teste do *benchmark* VOC2007.



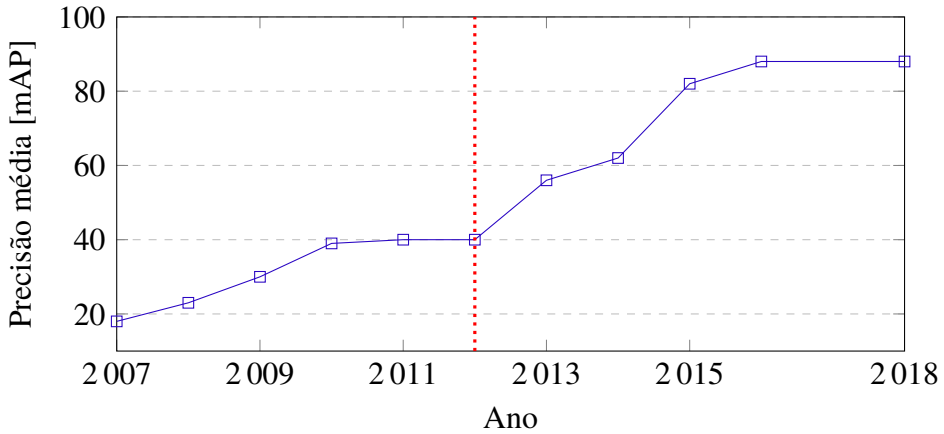
Fonte: Girshick (2015), Ren et al. (2015).

Devido a seus resultados promissores (Figura 20), tais detectores influenciaram o desenvolvimento de novos métodos também baseados em redes convolucionais profundas. É o caso dos detectores R-FCN (DAI et al., 2016) e Mask R-CNN (HE et al., 2017), também do tipo *region based*, e dos chamados métodos *single shot*, como as famílias de detectores SSD (LIU et al., 2015) e YOLO (REDMON; FARHADI, 2018a), capazes de operar em tempo real. Sendo assim, a partir do uso das redes convolucionais, deu-se início a uma nova era no campo da pesquisa referente à detecção de objetos, como aponta a Figura 21. Nela é possível notar o salto de qualidade realizado pelos novos detectores submetidos ao *benchmark* PASCAL VOC após o ano de 2012, a partir do qual o uso de CNN tornou-se uma tendência.

Figura 20 – Exemplo de aplicação do detector Faster R-CNN sobre imagem contendo objetos de diferentes categorias. Cada detecção corresponde a uma marcação retangular sobre a imagem, de maneira que suas cores e rótulos indicam a categoria do objeto detectado.

Fonte: o próprio autor.

Figura 21 – Evolução dos algoritmos submetidos ao *benchmark* de detecção de objetos PASCAL VOC ao longo dos últimos anos. A linha tracejada vermelha destaca o ano de 2012, o qual marca o início da submissão de detectores baseados em redes convolucionais profundas. Nota-se que a partir de tal ano houve uma acentuada aceleração na qualidade dos detectores submetidos.



Fonte: Liu et al. (2018).

3.2 Modelagem

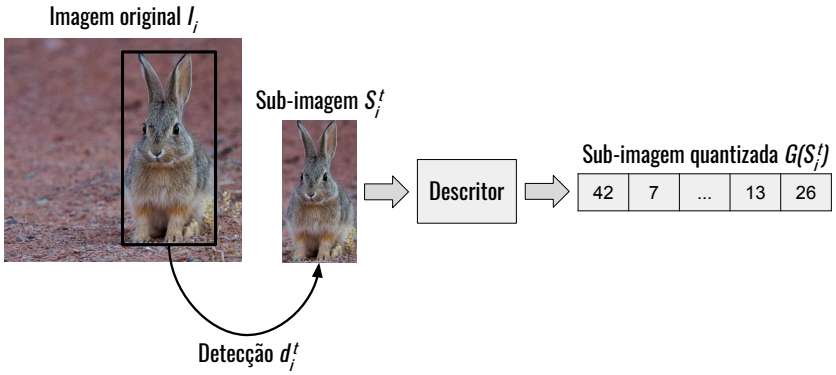
O rastreamento com base em detecções requer, além do detector propriamente, a concepção de modelos complementares, a partir dos quais seja possível descrever os objetos

rastreados $\{o_1, o_2, ..., o_i, ...\}$ por meio das detecções que compõem suas respectivas trajetórias. Dado um objeto o_i, rastreado até o instante t e com trajetória T_i, seu modelo pode ser utilizado para estimar suas características no instante futuro $t + 1$. As características estimadas podem ser comparadas às das novas detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$, também realizadas no instante $t + 1$. Por meio destas comparações, o rastreador é capaz de determinar qual destas detecções deve ser associada à trajetória T_i. Assim, a modelagem adequada de objetos é fundamental para sua re-identificação e, consequentemente, para a correta atualização de suas trajetórias (LEAL-TAIXÉ et al., 2017). Como observado no Capítulo 2, os principais modelos utilizados pelos algoritmos de rastreamento propostos recentemente na literatura podem ser divididos em duas categorias principais: os modelos de *aparência* e os modelos de *movimentação*. Ambas as categorias são discutidas com maiores detalhes nas subseções que se seguem.

3.2.1 Modelos de aparência

A aparência dos objetos corresponde a um conjunto de características visuais importantes para sua re-identificação ao longo do rastreamento. O conhecimento a respeito das suas cores, formas e texturas particulares permite que um determinado objeto seja reconhecido em outras imagens com maior facilidade. Nesse sentido, rastreadores baseados em detecção podem fazer uso de modelos de aparência para representar esse tipo de conhecimento. Mais precisamente, dado um objeto o_i com trajetória $T_i = \{d_i^1, ..., d_i^t\}$ ao longo da sequência de imagens $\{I_1, ..., I_t\}$, seu modelo de aparência deve ser capaz de computar um descritor A_i^{t+1}, o qual represente numericamente as características visuais estimadas para o objeto o_i no instante $t + 1$ (YU et al., 2016). Esta estimativa, por sua vez, é obtida a partir da aplicação de algoritmos descritores sobre as sub-imagens $\{S_i^1, ..., S_i^k\}$ referentes às regiões das imagens $\{I_1, ..., I_t\}$ delimitadas pelas detecções $\{d_i^1, ..., d_i^t\}$, respectivamente. A Figura 22 ilustra a aplicação de um descritor sobre a sub-imagem S_i^t, a qual refere-se à região da imagem I_t delimitada pela detecção d_i^t.

Figura 22 – Ilustração de um descritor genérico aplicado sobre a sub-imagem S^j delimitada pela detecção d_j de um determinado objeto o_j. Ao fim da aplicação, obtêm-se uma representação numérica $G(S^j)$ da sub-imagem no formato de um vetor.



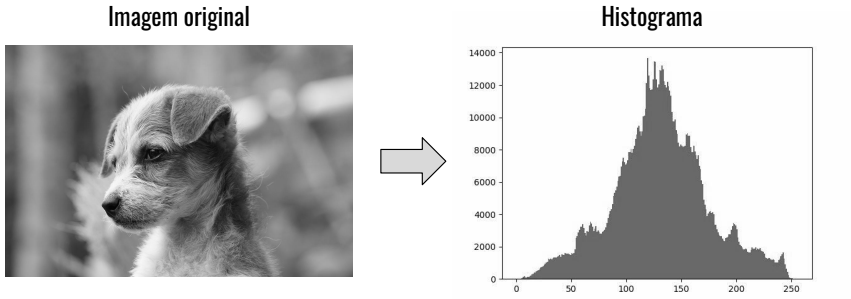
Fonte: o próprio autor.

O interesse em quantizar imagens vai muito além das aplicações relacionadas ao rastreamento: a re-identificação de pessoas (ZHENG et al., 2016), a busca por imagens (UCHIDA, 2016) e até mesmo a detecção de objetos (AGARWAL; TERRAIL; JURIE, 2018) são algumas das áreas de pesquisa que utilizam algoritmos para representar características visuais de maneira numérica. Desse modo, há na literatura especializada uma extensa coleção de algoritmos projetados para esta finalidade. Dentre os mais conhecidos encontram-se aqueles baseados em histogramas (LI et al., 2013). Dada uma imagem I com dimensão MxN e cujos níveis de cinza encontram-se no intervalo $[0, L-1]$, seu histograma normalizado é definido como a função discreta:

$$p(r_k) = \frac{n_k}{MN}\ ,\ 0 \leq k \leq L-1, \tag{3.1}$$

onde r_k corresponde ao valor da intensidade do *pixel* k e n_k é a quantidade de *pixels* em I com intensidade r_k (GONZALEZ, 2008). Descritores baseados em histogramas, portanto, quantizam a distribuição da intensidade de brilho ao longo de toda a imagem. No entanto, por não capturarem informações espaciais, os mesmos são incapazes de descrever a geometria e o posicionamento dos elementos que compõem a imagem. Logo, a utilização de histogramas é característica dos modelos de aparência baseados em cores. A Figura 23 ilustra o histograma obtido a partir de uma determinada imagem em escala de cinzas.

Figura 23 – Ilustração do histograma $p(r_k)$ referente a uma determinada imagem I em escala de cinzas. $p(r_k)$ indica a quantidade de *pixels* n_k em I (eixo das ordenadas) que apresentam determinado valor de intensidade de brilho r_k (eixo das abcissas).

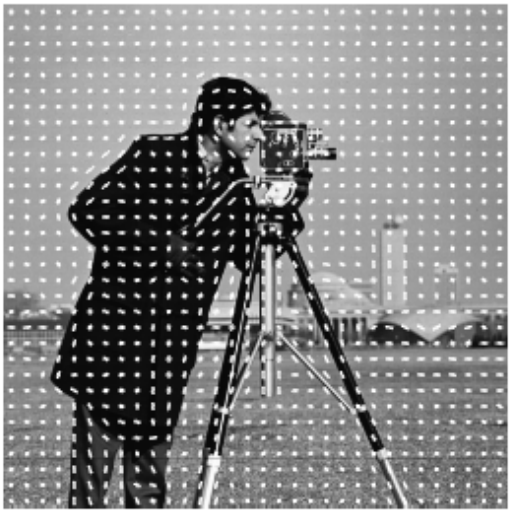


Fonte: o próprio autor.

Apesar de apresentarem baixo custo computacional e serem invariantes às operações de translação e rotação, os histogramas são incapazes de expressar informações espaciais da imagem, como dito anteriormente. Assim, os mesmos tornam-se ineficazes, por exemplo, para modelar objetos cujas distribuições de cores assemelham-se à do plano de fundo (LI et al., 2013). Desse modo, a literatura também propõe a utilização dos chamados descritores locais, os quais baseiam-se na detecção de arestas, saliências e pontos de interseção na imagem. Dentre os principais métodos destacam-se aqueles baseados na orientação de gradientes, como HOG, SIFT e SURF (MOHAN, 2014). Como ilustrado na Figura 24, estes descritores calculam a intensidade

e a direção da variação de brilho em diferentes regiões da imagem, o que permite descrevê-la com base nas silhuetas de objetos.

Figura 24 – Ilustração de descritores HOG plotados sobre uma imagem. Os gradientes apresentados indicam a intensidade e a direção da normal à variação de brilho em diferentes regiões da imagem. Percebe-se que tais gradientes permitem descrever a imagem a partir da silhueta de objetos.



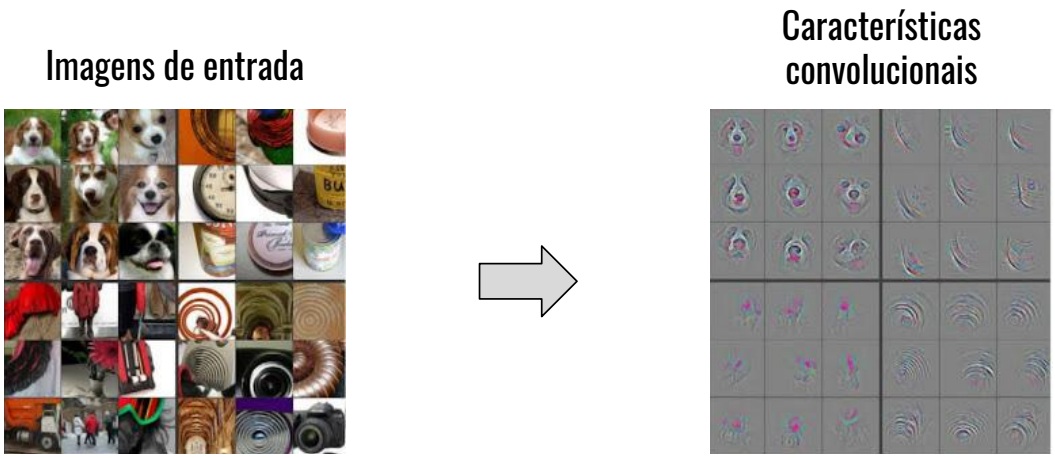
Fonte: <https://www.mathworks.com/help/vision/ref/extracthogfeatures.html>

Por fim, em contraste aos descritores manuais discutidos até aqui, métodos automáticos de extração de características e quantização de imagens baseados em Aprendizado de Máquina passaram a ser explorados com maior frequência devido à sua capacidade de descobrir e otimizar a representação de características visuais para aplicações específicas (TAIGMAN et al., 2014). Dentre estes métodos destacam-se aqueles baseados em redes neurais convolucionais profundas (CNN) (LI et al., 2018). Assim como observado no caso dos detectores de objetos, as CNN são capazes de extrair características visuais com diferentes níveis semânticos, as quais englobam cores, texturas, formas e demais propriedades da imagem a ser quantizada (HORN et al., 2017). Assim, uma das maneiras de se aplicar uma CNN para descrever uma imagem I é treinar um classificador baseado nesta rede e remover suas últimas camadas (geralmente as do tipo densa), as quais estão relacionadas à indicação da categoria à qual pertence I. A nova saída da rede passa a ser uma representação numérica $G(I)$, que era utilizada como entrada pelas camadas removidas para realizar a classificação de I.

A Figura 25 ilustra as características visuais de determinadas imagens, as quais foram extraídas a partir de uma CNN (ZEILER; FERGUS, 2014) treinada sobre a base ImageNet (DENG et al., 2009). Tais características são apresentadas através de mapas de ativação, os quais são gerados pela quarta camada convolucional daquela rede. É possível notar a capacidade da

rede em gerar mapas semelhantes a partir de imagens correlacionadas, de maneira que suas características em comum são realçadas. É o caso da face dos cachorros presentes no grupo superior esquerdo, da borda arredondada dos objetos contidos no canto superior direito, das regiões avermelhadas nas imagens presentes no canto inferior esquerdo e dos contornos circulares dos objetos contidos no grupo inferior esquerdo. Em todos os casos percebe-se determinado grau de invariância das características realçadas pela rede, o que a torna adequada para descrever a aparência de objetos no contexto do rastreamento.

Figura 25 – Ilustração das características visuais de determinadas imagens extraídas a partir de uma CNN com arquitetura AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), a qual contém 5 camadas convolucionais seguidas por 3 camadas densas. As características ilustradas correspondem a mapas de ativação obtidos na quarta camada convolucional da rede. É possível notar certa invariância dos mapas em relação a determinados grupos de imagens apresentadas como entrada à rede.



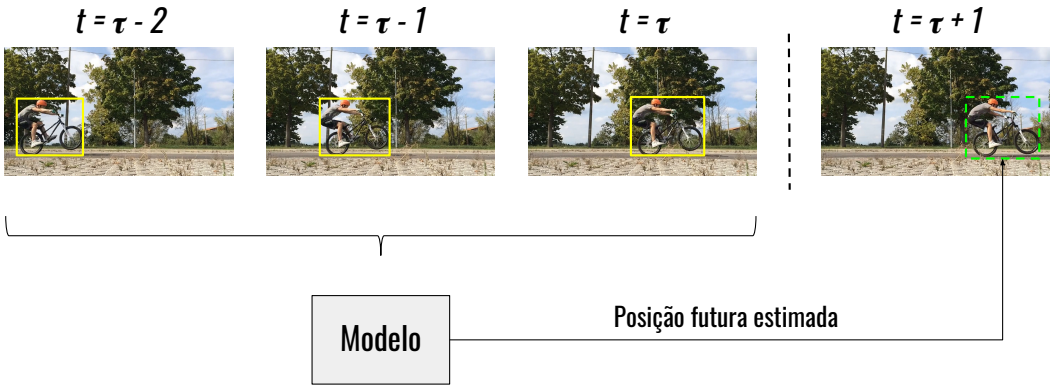
Fonte: Zeiler e Fergus (2014) com modificações do próprio autor.

3.2.2 Modelos de movimentação

Uma vez que o rastreamento visual abrange, dentre outras questões, a localização de objetos ao longo de um vídeo, tem-se que o conhecimento a respeito da maneira como os mesmos se locomovem pode ser útil para descrevê-los e posteriormente reconhecê-los. É com esse objeto que modelos de movimentação são aplicados durante o rastreamento. Tais modelos correspondem a ferramentas matemáticas capazes de descrever a dinâmica com que objetos se locomovem ao longo de uma sequência de imagens (FAN et al., 2016). A Figura 26 ilustra o uso de um destes modelos para estimar a futura posição de um determinado objeto a partir do conhecimento de suas posições passadas.

De acordo com a literatura, uma das principais formas de se modelar a movimentação de objetos no contexto do rastreamento corresponde à aplicação de filtros Bayesianos (FAN et

Figura 26 – Ilustração de um modelo de movimentação aplicado para estimar a posição de um determinado objeto no instante futuro $t = \tau + 1$ com base em sua posição nos instantes anteriores $t \leq \tau$.



Fonte: o próprio autor.

al., 2016). Para este caso, considera-se que cada objeto $\{o_i\}$ pode ser descrito por um vetor de estado X_i^t, o qual é formado, dentre outras possíveis variáveis, por suas coordenadas (x_i, y_i) sobre a imagem I_t no instante t (LI et al., 2010). Além disso, o estado de cada objeto $\{o_i\}$ evolui no decorrer do tempo de acordo com a equação diferencial:

$$X_i^t = f(X_i^{t-1}) + V_i^t, \tag{3.2}$$

na qual f corresponde à função de transição de estado do modelo e V_t^i ao seu ruído no instante t. Desse modo, os filtros Bayesianos buscam determinar o estado futuro X_i^{t+1} de um dado objeto o_i através da probabilidade *a posteriori* $P(X_i^{t+1}|Z_i)$, sendo $Z_i = \{Z_i^1, Z_i^2, ..., Z_i^t\}$ o conjunto de observações (ou medições) do estado do objeto o_i no instante t, as quais são definidas como:

$$Z_i^t = g(X_i^t) + W_i^t, \tag{3.3}$$

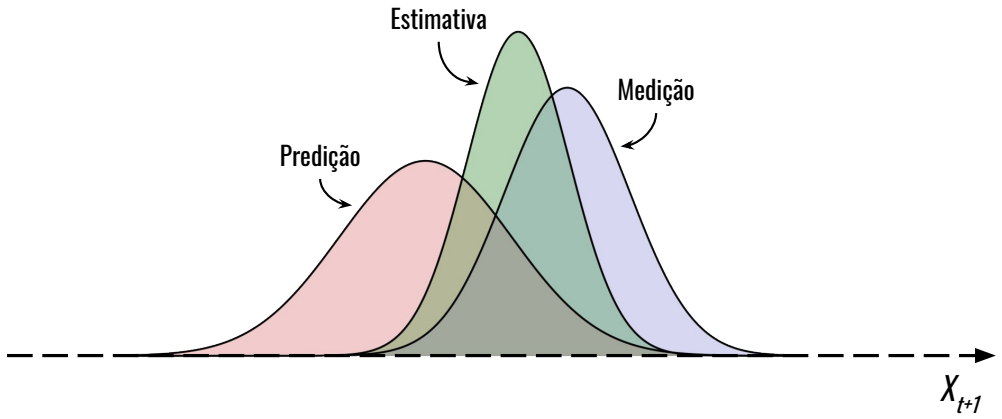
onde g corresponde à função de medição e W_i^t é o seu ruído. Vale notar que durante o rastreamento tais medições são comumente realizadas por meio de algoritmos de detecção de objetos (CZYZEWSKI; DALKA, 2008).

Tendo em vista os ruídos inerentes ao seu modelo de transição e às suas observações, o estado X_i^t não pode ser aferido diretamente. Dessa forma, o papel dos filtros Bayesianos é gerar uma estimativa $\hat{X}_i^t$, a qual maximize a verossimilhança $P(X_i^t|\hat{X}_i^t)$. Em outras palavras, seu objetivo é filtrar observações ruidosas, de modo a obter-se a melhor estimativa possível para um determinado sinal, neste caso o estado X_i^t de um objeto o_i rastreado.

Dentre os principais filtros Bayesianos descritos na literatura, destaca-se o Filtro de

Kalman (KALMAN, 1960). Considerando que as equações 3.2 e 3.3 descrevem sistemas lineares, o Filtro de Kalman computa o estado estimado $\hat{X}_i^{t+1}$ a partir da combinação das funções de densidade de probabilidade (no inglês, *probability density function* ou PDF) do modelo de transição de estado e de suas observações. Para tanto, o filtro executa recursivamente duas etapas: a de predição e a de medição. Na primeira, utiliza-se o modelo de transição para predizer o futuro estado $\hat{X}_i^{t+1}$ com base em seu valor atual $\hat{X}_i^t$; já na segunda, realiza-se uma medição indireta do estado X_i^t e ajusta-se a predição realizada na etapa anterior. Tal processo é repetido durante todo o rastreamento. A Figura 27 ilustra a manipulação realizada pelo Filtro de Kalman para gerar a PDF do estado futuro estimado $\hat{X}_i^{t+1}$.

Figura 27 – Ilustração da combinação de funções de densidade de probabilidade (PDF) realizada pelo Filtro de Kalman com base num sistema discreto unidimensional. As PDF do estado estimado no instante $t+1$ a partir do modelo de transição e da medição Z_i^{t+1} são mesclados pelo filtro, de modo a obter-se a PDF do estado estimado $\hat{X}_i^{t+1}$, o qual maximiza a verosimilhança $P(X_i^{t+1}|\hat{X}_i^{t+1})$ (FARAGHER, 2012).



Fonte: o próprio autor.

Muito embora seja considerado um estimador ótimo, o Filtro de Kalman assume que tanto a incerteza do modelo de transição de estado quanto as de suas observações comportam-se segundo uma distribuição gaussiana. Além disso, também assume-se que os processos que definem tais variáveis são de natureza linear (LI et al., 2016). Tais restrições nem sempre refletem o cenário do rastreamento. Desse modo, um segundo tipo de filtro Bayesiano surge como alternativa na literatura: trata-se dos Filtros de Partículas, os quais são capazes de estimar o estado de sistemas dinâmicos não-lineares (GIRON-SIERRA, 2016). A intuição por trás destes filtros está em gerar uma aproximação do estado futuro estimado $\hat{X}_i^{t+1}$ através de um conjunto finito de N amostras aleatórias ponderadas $\{\hat{X}_{i,p}^{t+1}, w_{i,p}^{t+1}\}_{p=1}^{N}$, denominadas partículas. Tal aproximação é

realizada como:

$$p(X_i^t|Z_i^{1:t}) \approx \sum_{p=1}^{N} w_{i,p}^t \delta(X^t - \hat{X}_{i,p}^t), \tag{3.4}$$

sendo $\hat{X}_{i,p}^t$ e $w_{i,p}^t$ o estado estimado e o peso atribuídos à partícula de índice p (KüNSCH, 2013), respectivamente. De modo semelhante ao Filtro de Kalman, o algoritmo que embasa os Filtros de Partículas fundamenta-se na execução recursiva das etapas de predição e medição. Na primeira utiliza-se o modelo de transição para computar a nova amostra do estado $\hat{X}_i^{t+1}$ predita por cada partícula. Já na segunda etapa, seus respectivos pesos são atualizados com base na nova medição Z_i^{t+1}, de modo a ajustar-se o estado final estimado pelo filtro. Todo esse processo é repetido ao longo da aplicação do filtro (ABDELALI; ESSANNOUNI; ABOUTAJDINE, 2016).

Além dos filtros Bayesianos, vale ressaltar também a utilização de técnicas de Aprendizado de Máquina para a construção de modelos de movimentação aplicáveis ao contexto do rastreamento. Tais técnicas também baseiam-se na estimativa do estado futuro X_i^{t+1} de um determinado objeto o_i a partir de observações das suas posições passadas. No entanto, por seguirem um paradigma de aprendizado baseado em exemplos, estas técnicas tendem a ser mais agnósticas do que aqueles filtros no que se refere a suas suposições em relação à dinâmica do movimento tratado (MARTINEZ; BLACK; ROMERO, 2017). Além disso, em contraste à natureza degenerativa dos filtros Bayesianos, os modelos de Aprendizado de Máquina têm caráter discriminativo, ou seja, suas saídas são valores determinísticos ao invés de distribuições (ITER; KUCK; ZHUANG, 2016).

Dentre as técnicas de aprendizado de máquina mais exploradas recentemente para modelar a movimentação de objetos no contexto do rastreamento, encontram-se aquelas baseadas no paradigma de aprendizado supervisionado (WANG; ZHANG; YI, 2017; CHENG et al., 2018; TANG et al., 2018). Em termos formais, estes algoritmos podem ser aplicados da seguinte forma: dado um conjunto de N observações $Z_i = \{Z_i^t, Z_i^{t-1}, ..., Z_i^{t-N}\}$ respectivamente relacionadas ao estado X_i^k do objeto o_i nos instantes $k \in [t - N,\ t]$, sendo t o instante atual, constrói-se por meio de exemplos rotulados uma função de regressão $f(Z_i)$, a qual gera como saída o valor $\hat{X}_i^{t+1}$. Este valor deve minimizar o erro $E = \delta(\hat{X}_i^{t+1} - X_i^{t+1})$, sendo X_i^{t+1} o estado do objeto o_i no instante futuro $t + 1$ (ALTCHé; FORTELLE, 2017). Vale ressaltar novamente que tanto as medições contidas no conjunto Z_i quanto o estado estimado $\hat{X}_i$ do objeto o_i são tratados pelo algoritmo de aprendizado como valores determinísticos, não como distribuições de probabilidade.

3.3 Discriminação

Como discutido nas subseções anteriores, durante a execução do paradigma *tracking-by-detection* busca-se construir modelos que descrevam os N múltiplos objetos $\{o_1, ..., o_N\}$ rastreados ao longo de uma sequência de imagens até o instante atual t. Independentemente das características

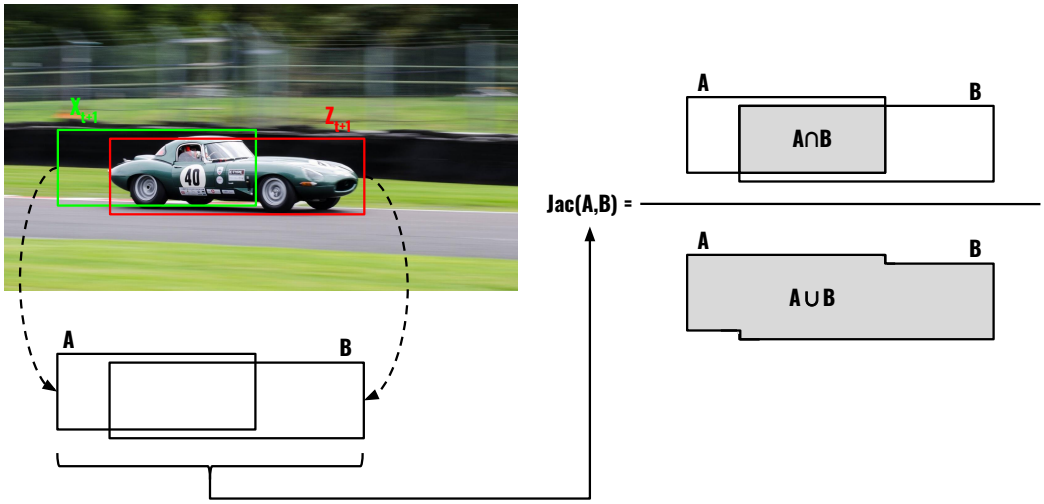
descritas, estes modelos são atualizados a partir de M novas detecções $\{d_1^{t+1}, ..., d_M^{t+1}\}$ obtidas no instante $t+1$. No entanto, pelo fato dos rastreadores gerenciarem a trajetória de múltiplos objetos simultaneamente, surge a necessidade de determinar quais detecções referem-se a quais objetos. Em outras palavras, para cada par $(o_i\ ,\ d_j^{t+1})$, onde $o_i \in \{o_1, ..., o_N\}$ e $d_j^{t+1} \in \{d_1^{t+1}, ..., d_M^{t+1}\}$, os rastreadores devem computar uma medida $s_{i,j}$ que indique a similaridade entre o estado do objeto o_i estimado por seu modelo para o instante $t+1$ e a detecção d_j^{t+1} obtida também no instante $t+1$.

Nesse sentido, diferentes métricas e estratégias de discriminação podem ser aplicadas a depender do tipo de modelo empregado para descrever os objetos rastreados. No caso dos modelos de movimentação, busca-se comparar o estado futuro $\hat{X}_i^{t+1}$ estimado pelo modelo do objeto o_i no instante $t+1$ com a detecção d_j^{t+1}. Considerando-se que tanto o estado estimado $\hat{X}_i^{t+1}$ quanto a detecção d_j^{t+1} tenham como atributos as coordenadas (x, y) e as dimensões (w, h) de seus respectivos objetos em relação à imagem I_{t+1} obtida no instante $t+1$, é possível compará-los por meio da taxa de sobreposição entre suas regiões (BEWLEY et al., 2016; LIN et al., 2017; BOCHINSKI; EISELEIN; SIKORA, 2017). Tal medida, conhecida como Índice de Jaccard (BOUCHARD; JOUSSELME; DORé, 2013), é definida como:

$$Jac(A, B) = \frac{C(A \cap B)}{C(A \cup B)}, \tag{3.5}$$

sendo A e B, respectivamente, conjuntos não-nulos de *pixels* contidos nas regiões da imagem ocupadas pelo estado estimado $\hat{X}_i^{t+1}$ e pela detecção d_j^{t+1}, enquanto $C(.)$ corresponde ao operador de contagem de amostras (SHI; NGAN; LI, 2014). A Figura 28 ilustra a computação de tal medida. Nota-se que o numerador do Índice de Jaccard corresponde à área de interseção entre A e B, enquanto que seu denominador equivale à área de união entre tais conjuntos. Assim, $Jac(A, B) \in \mathbb{R} \mid 0 \leq Jac(A, B) \leq 1$, de modo que $Jac(A, B) = 1$ sempre que $A = B$.

Figura 28 – Ilustração do cálculo da taxa de sobreposição (Índice de Jaccard) entre o estado estimado $\hat{X}_i^{t+1}$ de um objeto o_i no instante $t+1$ e a detecção d_j^{t+1} realizada no instante $t+1$ e referente ao objeto o_j. Dadas as sub-imagens A e B, respectivamente formadas pelos *pixels* sobrepostos por $\hat{X}_i^{t+1}$ e d_j^{t+1}, o numerador do Índice de Jaccard corresponde à área de interseção entre A e B, enquanto seu denominador equivale à área de união entre estas sub-imagens.



Fonte: o próprio autor.

Já nos casos em que são considerados modelos de aparência, busca-se comparar o descritor A_i^{t+1} computado pelo modelo, o qual refere-se à aparência estimada do objeto o_i no instante $t+1$, e o descritor B_j, o qual quantifica a sub-imagem S_j^{t+1} relacionada à região da imagem I_{t+1} delimitada pela detecção d_j^{i+1}. Para tanto, o mesmo algoritmo descritor utilizado para a construção do modelo deve ser aplicado sobre S_j^{t+1} de modo a se obter B_j. Assumindo que tanto A_i^{i+1} quanto B_j correspondam a vetores de características visuais com dimensão N, o complemento de sua distância pode ser utilizado como medida de similaridade $s_{i,j}$. Dentre as distâncias já utilizadas com essa finalidade encontram-se a distância Euclidiana (SCHROFF; KALENICHENKO; PHILBIN, 2015), a distância do Cosseno (NGUYEN; BAI, 2011) e a distância χ^2 ponderada (TAIGMAN et al., 2014). Esta última é definida como:

$$\chi^2(V_1, V_2) = \sum_{k=0}^{N} w_k \frac{(V_1[k] - V_2[k])^2}{(V_1[k] + V_2[k])}, \tag{3.6}$$

onde V_1 e V_2 representam, respectivamente, os descritores A_i^{t+1} e B_j, N corresponde à dimensão dos vetores V_1 e V_2, e w_k refere-se ao peso dado para o termo $\frac{(V_1[k]-V_2[k])^2}{(V_1[k]+V_2[k])}$.

Finalmente, é possível também que mais de um tipo de modelo seja considerado para descrever os objetos rastreados. Por exemplo, pode-se utilizar simultaneamente modelos de movimentação e modelos de aparência, de modo a tornar mais robusta a re-identificação de objetos. (BORAGULE; JEON, 2017b). De fato, durante curtos intervalos de tempo nos quais a aparência

de um determinado objeto é significativamente alterada devido a variações de luminosidade, seu modelo de movimentação pode ser mais útil para re-identificá-lo. Já durante longos períodos de oclusão, sua aparência pode ser mais discriminatória. Nesse sentido, uma das formas de se obter a medida de similaridade final $s_{i,j}$ é combinando-se a medida $s^a_{i,j}$, computada individualmente através do modelo de aparência, e a medida $s^m_{i,j}$, também calculada individualmente por meio do modelo de movimentação. Dentre os possíveis tipos de combinação encontra-se a seguinte soma ponderada:

$$s_{i,j} = \alpha \cdot s^a_{i,j} + \beta \cdot s^m_{i,j}, \tag{3.7}$$

sendo α e β os pesos dados às medidas de similaridade baseadas exclusivamente no modelo de aparência e no modelo de movimentação, respectivamente (YOON et al., 2016; YU et al., 2016; WOJKE; BEWLEY; PAULUS, 2017).

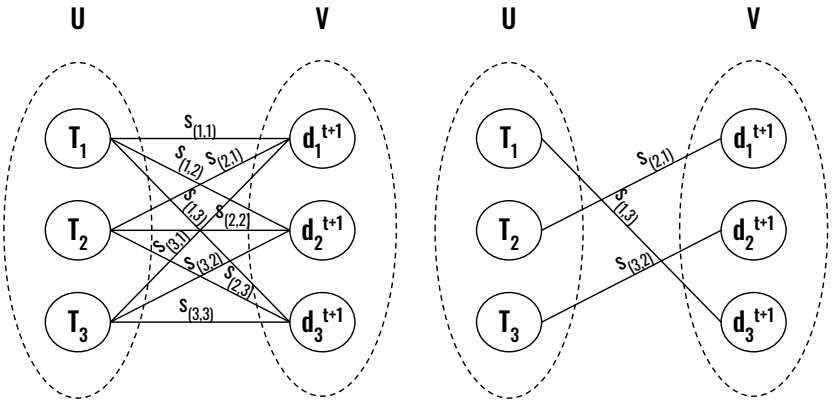
3.4 Associação

A última etapa realizada durante cada iteração do rastreamento por detecção corresponde à associação entre as trajetórias $\{T_1, T_2, ..., T_i, ...\}$, relacionadas aos objetos rastreados $\{o_1, o_2, ..., o_i, ...\}$, e as novas detecções $\{d^{t+1}_1, d^{t+1}_2, ..., d^{t+1}_j, ...\}$ realizadas no instante $t + 1$. Para tanto, os rastreadores baseados no paradigma *tracking-by-detection* formulam esta associação como um problema de otimização em grafos (FAN et al., 2016). No caso mais simples, constrói-se um grafo bipartido $G(U, V, E)$, cujos vértices $U = \{u_1, u_2, ..., u_i, ...\}$ representam as trajetórias $\{T_1, T_2, ..., T_i, ...\}$, os vértices $V = \{v_1, v_2, ..., v_j, ...\}$ referem-se às detecções $\{d^{t+1}_1, d^{t+1}_2, ..., d^{t+1}_j, ...\}$ e as arestas $E = \{e_{1,1}, e_{1,2}, ..., e_{i,j}, ...\}$ possuem como peso as medidas de similaridade $\{s_{1,1}, s_{1,2}, ..., s_{i,j}, ...\}$, as quais estão relacionadas aos respectivos pares $\{(o_1\ ,\ d^{t+1}_1), (o_1\ ,\ d^{t+1}_2), ..., (o_i\ ,\ d^{t+1}_j), ...\}$. Assim, deve-se determinar o sub-conjunto de arestas que maximize a soma de pesos, com a restrição de que cada vértice $u_i \in U$ e $v_j \in V$ deve estar conectado a no máximo uma única aresta. A Figura 29 ilustra a construção de tal grafo juntamente com a resolução do problema de associação. Esta pode ser alcançada, por exemplo, por meio do Método Húngaro (KUHN, 2005).

A estratégia de associação apresentada anteriormente considera que as trajetórias $\{T_1, T_2, ..., T_i, ...\}$ só podem ser atualizadas com base em detecções $\{d^{t+1}_1, d^{t+1}_2, ..., d^{t+1}_j, ...\}$ extraídas de uma única imagem I_{t+1} por iteração. Ou seja, uma nova imagem I_{t+2} somente pode ser processada após finalizada a associação entre $\{T_i\}$ e $\{d^{t+1}_j\}$ com base em I_{t+1}. Devido ao número limitado de imagens consideradas durante cada iteração, tal estratégia é classificada como local. Estratégias de associação deste tipo são geralmente empregadas por algoritmos de rastreamento projetados para aplicações *online*, nas quais não se tem acesso *a priori* a toda a sequência de N imagens $\{I_1, ..., I_N\}$. No entanto, nos casos em que tal sequência é conhecida, é possível estender a estratégia local discutida anteriormente, de modo a atualizar as trajetórias $\{T_1, T_2, ..., T_i, ...\}$

Figura 29 – Ilustração da associação de trajetórias $\{T_1, T_2, ..., T_i, ...\}$ e detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$ através de otimização de grafo. Inicialmente constrói-se um grafo bipartido $G(U, V, E)$, onde $U = \{T_1, T_2, ..., T_i, ...\}$, $V = \{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$ e o peso das arestas E corresponde às medidas de similaridades $\{s_{1,1}, s_{1,2}, ..., s_{i,j}, ...\}$, as quais relacionam-se aos respectivos pares $\{(o_1\ ,\ d_1^{t+1}), (o_1\ ,\ d_2^{t+1}), ..., (o_i\ ,\ d_j^{t+1}), ...\}$. Em seguida, determina-se o subconjunto de arestas que maximize a soma de pesos, de modo que cada vértice esteja conectado a no máximo uma única aresta.

(a) Construção do grafo $G(U, V, E)$. (b) Resolução da associação.

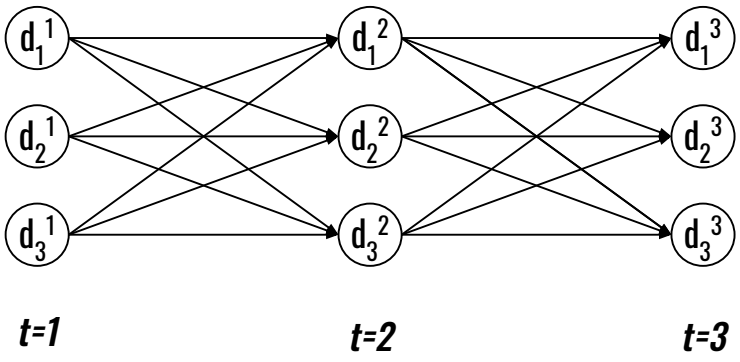


Fonte: o próprio autor.

a partir das detecções $\{D_t \mid (\forall t)(t \in \{1, ..., N\}$ e $D_t = \{d_1^t, d_2^t, ..., d_i^t, ...\})\}$ obtidas ao longo de $\{I_1, ..., I_N\}$ numa única etapa de associação. Esse tipo de abordagem, considerada global, pode ser implementada a partir da construção de um grafo orientado $G(V, E)$ cujos vértices $v_j \in V$ representam todas as detecções realizadas ao longo das N imagens da sequência $\{I_1, ..., I_N\}$. Além disso, o conjunto de arestas E é formado por elementos definidos como $e = (d_i^t, d_j^{t+1})$, os quais conectam detecções realizadas nas imagens I_t e I_{t+1}, respectivamente. Assim como para associações locais, os pesos das arestas deste grafo equivalem à medida de similaridade $s_{i,j}$ referente às suas respectivas detecções. A Figura 30 ilustra a construção de um grafo orientado aplicado à associação global de detecções.

Por fim, vale ressaltar que, assim como no caso local, ao final da associação global são obtidas trajetórias $\{T_1, T_2, ..., T_i, ...\}$, onde $T_i = \{d_i^1, d_i^2, ...\}$, que maximizem a soma dos pesos das arestas de suas respectivas detecções. Tal resultado pode ser alcançado, por exemplo, através de técnicas de programação linear e programação dinâmica (FAN et al., 2016).

Figura 30 – Ilustração de um grafo direcional $G(V, E)$ aplicado à associação global de detecções. Seus vértices $v_j \in V$ representam todas as detecções $\{D_t \mid (\forall t)(t \in \{1, ..., N\}$ e $D_t = \{d_1^t, d_2^t, ..., d_i^t, ...\})\}$ realizadas ao longo das N imagens da sequência $\{I_1, ..., I_N\}$. Já o seu conjunto de arestas E é formado por elementos definidos como $e = (d_i^t, d_j^{t+1})$, os quais conectam detecções realizadas nas imagens I_t e I_{t+1}, respectivamente. O peso de suas arestas equivale à medida de similaridade $s_{i,j}$ referente às suas respectivas detecções.



Fonte: o próprio autor.

4

Método Proposto para Rastreamento

O método proposto baseia-se num modelo de regressão que recebe como entrada características relacionadas à aparência e à movimentação de objetos $\{o_1, o_2, ..., o_i, ...\}$, rastreados até o instante t e com trajetórias $\{T_1, T_2, ..., T_i, ...\}$, sendo $T_i = \{d_i^{k_1}, ..., d_i^{k_Z} \mid k_Z \leq t\}$ uma trajetória de comprimento Z, e das novas detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$, realizadas no instante atual $t+1$, e gera como saída o custo $c_{i,j}$ da associação entre cada par (T_i, d_j^{t+1}). As seções apresentadas a seguir discutem o método de rastreamento proposto, denominado SmartSORT, e a metodologia desenvolvida para a indução do seu modelo de regressão.

4.1 Visão Geral

O algoritmo 1 descreve o método SmartSORT. Este método considera um cenário de rastreamento no qual cada objeto o_i rastreado até o instante atual t possui uma trajetória $T_i = \{d_i^{k_1}, ..., d_i^{k_Z} \mid k_Z \leq t\}$ de comprimento Z, sendo seu estado s_i modelado como:

$$s_i = [u_i, v_i, h_i, r_i, \boldsymbol{a}_i], \tag{4.1}$$

em que u_i e v_i representam, respectivamente, as posições verticais e horizontais em *pixels* do centro do objeto o_i sobre a imagem I_{k_Z} obtida em k_Z; h e r correspondem, respectivamente, à altura e à razão de aspecto da detecção $d_i^{k_Z}$; e $\boldsymbol{a}$ denota o descritor de aparência obtido com base em $d_i^{k_Z}$. O estado s_i do objeto o_i é atualizado sempre que sua trajetória T_i no instante t é associada a uma nova detecção d_j^{t+1} (linha 8 do algoritmo 1). Nesse caso, s_i incorpora as dimensões de d_j^{t+1}, juntamente com seu descritor de aparência (Figura 31). Caso nenhuma associação aconteça, preserva-se o estado s_i.

De modo a gerenciar as trajetórias $\{T_1, T_2, ..., T_i, ...\}$, o método SmartSORT considera que para cada objeto o_i rastreado até o instante t há um contador de perdas L_i, o qual é incrementado somente quando nenhuma detecção é associada a T_i durante uma iteração do rastreamento. Se

Algoritmo 1: Método de rastreamento SmartSORT.

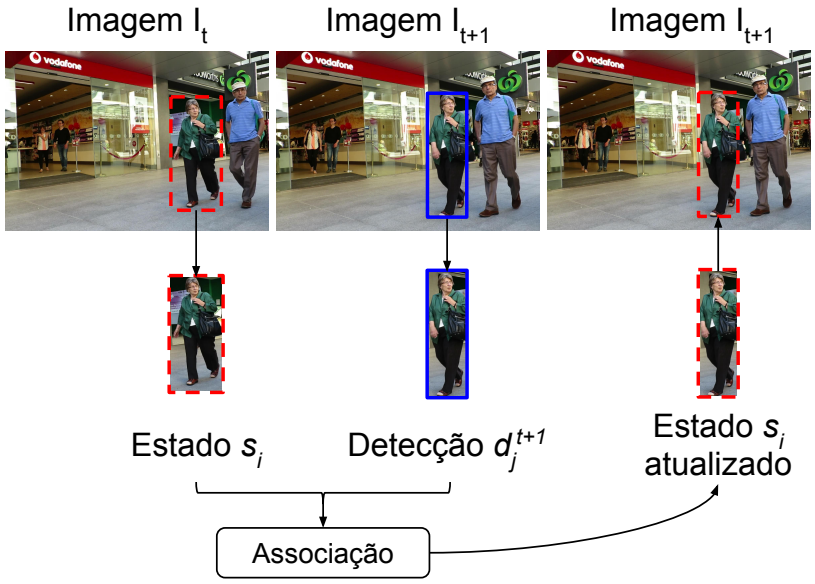
Dados: Conjunto de novas detecções D realizadas em $t+1$, conjunto de trajetórias $\{T_1, T_2, ..., T_i, ...\}$ existentes em t, limiar de custo C_{max}, limiar de perda L_{max}
Resultado: Conjunto de trajetórias atualizadas U em $t+1$

```
1  para cada d ∈ D faça
2  |  d ← computaDescritor(d);
3  fim
4  U ← {T_1, T_2, ..., T_i, ...};
5  C ← custoAssociacao({T_1, T_2, ..., T_i, ...}, D, C_max);
6  A ← metodoHungaro(C);
7  para cada (T, d) ∈ A faça
8  |  T ← atualizaTrajetoria(T, d);
9  fim
10 para cada T ∈ {T_1, T_2, ..., T_i, ...} faça
11 |  se T ∉ A então
12 |  |  incrementaContadorPerdas(T);
13 |  |  se não Confirmado(T) então
14 |  |  |  U ← U − {T};
15 |  |  fim
16 |  fim
17 |  se contadorPerdas(T) > L_max então
18 |  |  U ← U − {T};
19 |  fim
20 fim
21 para cada d ∈ D faça
22 |  se d ∉ A então
23 |  |  T ← {d};
24 |  |  U ← U ∪ {T};
25 |  fim
26 fim
27 retorna U;
```

o valor de L_i excede um determinado limiar L_{max}, a trajetória T_i é descartada, uma vez que assume-se que o objeto o_i deixou a cena permanentemente (linha 18 do algoritmo 1). Uma nova trajetória T_j é criada para cada detecção d_j^{t+1} não associada a nenhuma trajetória já existente (linha 23 do algoritmo 1). Durante as próximas três iterações a partir de sua criação, a nova trajetória T_j é considerada não-confirmada. O SmartSORT descarta trajetórias não-confirmadas cujos respectivos contadores de perda são incrementados (linha 14 do algoritmo 1).

O método SmartSORT trata a associação entre trajetórias existentes e novas detecções como um problema de otimização em grafos. De modo a computar o seu custo de associação (linha 5 do algoritmo 1), consideram-se informações de movimentação e aparência (*i.e.*, suas dimensões e seus descritores). No entanto, diferentemente de trabalhos relacionados (YOON et al., 2016; YU et al., 2016; WOJKE; BEWLEY; PAULUS, 2017), o SmatSORT computa tal custo por meio de um modelo de regressão induzido através de técnicas de Aprendizado de Máquina

Figura 31 – Ilustração da atualização do estado s_i de um objeto o_i, o qual foi associado a uma nova detecção d_j^{t+1}.



Fonte: o próprio autor.

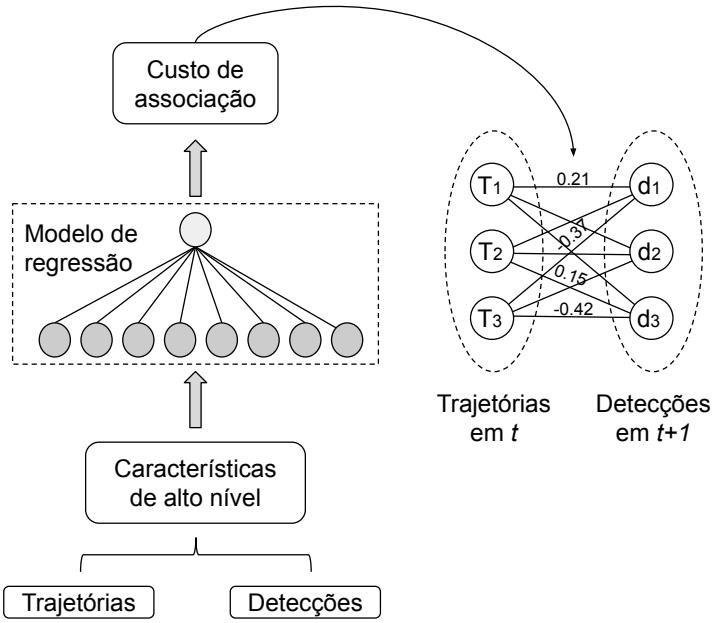
(Figura 32). Neste trabalho, foi considerada uma rede neural do tipo MLP treinada a partir do algoritmo *Backpropagation* (CUN, 1988) e descrita com maior profundidade na seção 4.2. Assim, dado o vetor de características $\boldsymbol{f}_{T_i,d_j}$, relacionado à i-ésima trajetória T_i e à j-ésima detecção d_j, o modelo de regressão computa seu respectivo custo de associação $c_{i,j} \in [-1, 1]$. O modelo realiza tal computação para todas as combinações possíveis de detecções e trajetórias (Figura 32). A seção 4.2 discute mais detalhadamente o processo de indução deste modelo.

Uma vez que o modelo de regressão tenha computado todos os custos de associação, o algoritmo de rastreamento resolve o grafo bipartido a partir da aplicação do Método Húngaro (KUHN, 2005)(linha 6 do algoritmo 1). Além disso, descartam-se associações cujos respectivos custos ultrapassem um determinado limiar C_{max}, já que as mesmas são consideradas inadmissíveis. Tendo em vista a simetria do intervalo de $c_{i,j}$, C_{max} foi adotado como 0, de modo a maximizar a margem que separa associações inadmissíveis daquelas aceitáveis. Vale ressaltar que C_{max} corresponde ao único hiper-parâmetro relacionado à etapa de associação do método SmartSORT.

4.2 Modelo de Regressão

A principal contribuição do método proposto está em seu modelo de regressão, o qual é capaz de estimar a similaridade entre um objeto o_i com trajetória $T_i = \{d_i^{k_1}, ..., d_i^{k_Z} \mid k_Z \leq t\}$ de comprimento Z até o instante t e um objeto o_j apontado pela detecção d_j^{t+1}, valendo-se de suas características de aparência e movimentação (linha 5 do algoritmo 1). De modo a induzi-lo,

Figura 32 – Ilustração do modelo induzido neste trabalho para estimar o custo de associação entre detecções. Dado um conjunto de trajetórias $\{T_1, T_2, ..., T_i, ...\}$, sendo $T_i = \{d_i^{k_1}, ..., d_i^{k_Z} \mid k_Z \leq t\}$ uma trajetória de comprimento Z, relacionadas a objetos $\{o_1, o_2, ..., o_i, ...\}$ rastreados até o instante t e um conjunto de novas detecções $\{d_1^{t+1}, d_2^{t+1}, ..., d_j^{t+1}, ...\}$ realizadas no instante atual $t + 1$, o modelo de regressão estima o custo $c_{i,j}$ da associação entre cada par (T_i, d_j^{t+1}). A saída obtida é utilizada para a construção de um grafo bipartido, o qual é resolvido através do Método Húngaro (KUHN, 2005).



Fonte: o próprio autor.

inicialmente considerou-se como entrada o vetor de características $\boldsymbol{f}$, o qual é definido como:

$$\boldsymbol{f}_{d_i^{k_Z}, d_j^{t+1}} = [u, v, h, r, \Delta u, \Delta v, \Delta h, \Delta r, \Delta \boldsymbol{a}, \Delta t], \tag{4.2}$$

onde (u, v), h e r correspondem à posição e às dimensões da última detecção $d_i^{k_Z}$ associada à T_i, respectivamente; Δu, Δv, Δh e Δr representam a diferença normalizada entre a posição e as dimensões das detecções d_j^{t+1} e $d_i^{k_Z}$, respectivamente; $\Delta \boldsymbol{a}$ é a distância do cosseno entre os descritores das regiões delimitadas por d_j^{t+1} e $d_i^{k_Z}$; e $\Delta t = (t + 1) - k^Z$ refere-se ao intervalo de tempo entre a realização das detecções d_j^{t+1} e $d_i^{k_Z}$.

A intuição por trás da escolha dos atributos u, v, h e r está na sua capacidade de fornecer ao modelo informações a respeito da posição e das dimensões absolutas estimadas para o objeto o_i no instante t. Através destas informações pode-se descrever a movimentação de o_i em relação à câmera. Os atributos Δu e Δv podem ser úteis para a identificação de situações nas quais a associação entre d_j^{t+1} e T_i é impraticável devido à distância existente entre as detecções d_j^{t+1} e $d_i^{k_Z}$, enquanto que Δh e Δr permitem que o método de rastreamento compreenda a geometria de ambos os objetos o_i e o_j. Por sua vez, a distância $\Delta \boldsymbol{a}$ favorece a discriminação de d_j^{t+1} e $d_i^{k_Z}$ com

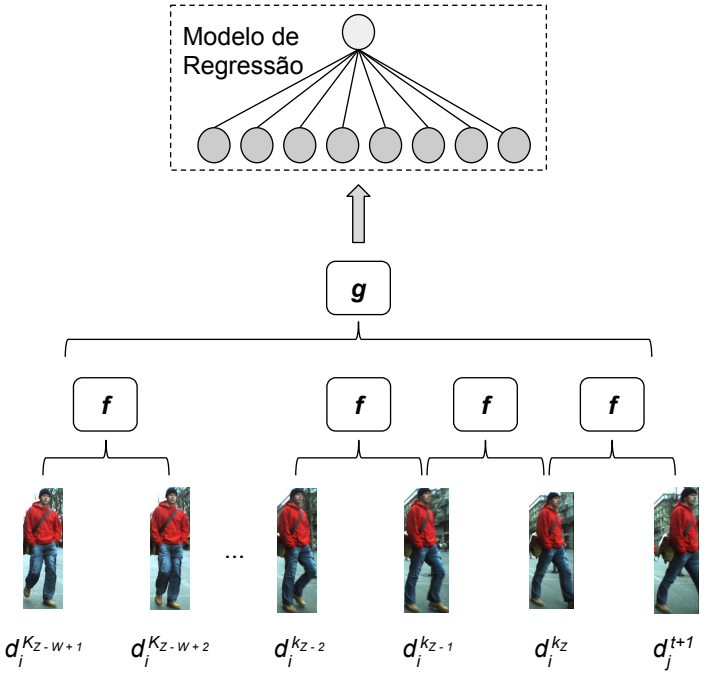
base nas características visuais de seus respectivos objetos o_j e o_i (*i.e.* características profundas extraídas por meio de uma CNN). Finalmente, Δt permite que o SmartSORT compreenda a variação temporal das características de movimentação e de aparência do objeto o_i ao longo do rastreamento, o que pode ser especialmente útil para o tratamento de oclusões.

De modo a aprimorar a capacidade do SmartSORT de entender a movimentação dos objetos rastreados, considerou-se expandir o vetor $\boldsymbol{f}$ a partir de múltiplas detecções pertencentes à trajetória T_i. Assim, adotou-se uma estratégia de janela deslizante, na qual o novo vetor de características de entrada $\boldsymbol{g}$ é definido como:

$$\boldsymbol{g}_{T_i,\, d_j^{t+1}} = [\boldsymbol{f}_{d_i^{k_Z}, d_j^{t+1}}, \boldsymbol{f}_{d_i^{k_{Z-1}}, d_i^{k_Z}}, \ldots, \boldsymbol{f}_{d_i^{k_{Z-W+1}}, d_i^{k_{Z-W+2}}}], \tag{4.3}$$

onde $\boldsymbol{f}$ corresponde ao vetor de características definido pela Equação 4.2; d_j^{t+1} é uma nova detecção realizada no instante $t+1$; $d_i^{k_z}$ denota uma detecção realizada no instante $k_z \mid k_z \leq t$ e pertencente à trajetória T_i de comprimento Z; e W representa o tamanho da janela deslizante. Esta estratégia permite que o método SmartSORT compreenda tanto a movimentação de um objeto o_i como também a variação de sua aparência ao longo do tempo. A Figura 33 ilustra a estratégia de janela deslizante proposta.

Figura 33 – Ilustração da estratégia de janela deslizante aplicada para calcular o custo de associação entre uma trajetória $T_i = \{d_i^{k_1}, ..., d_i^{k_Z} \mid k_Z \leq t\}$ e uma detecção d_j^{t+1}. São computados W vetores de características $\boldsymbol{f}$ relacionados à d_j^{t+1} e às detecções $\{d_i^{k_{Z-W+1}}, ..., d_i^{k_Z}\}$ pertencentes à T_i. Ao final, os W vetores $\boldsymbol{f}$ são concatenados para formar o vetor $\boldsymbol{g}$, o qual é apresentado como entrada para o modelo de regressão no instante $t + 1$.



Fonte: o próprio autor.

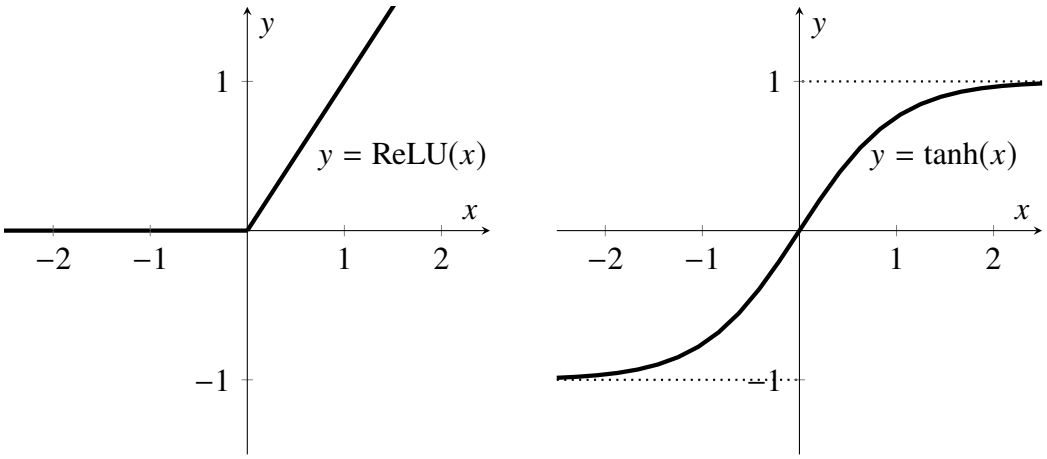
Como já mencionado, o modelo de regressão foi projetado para estimar o custo de associação $c_{i,j} \in [-1, 1]$ entre uma trajetória T_i e uma detecção d_j^{t+1}. Inspirado por Son et al. (2017), Sadeghian, Alahi e Savarese (2017), tal modelo foi obtido através de um algoritmo de aprendizado supervisionado. Para isso, criou-se uma base de dados composta por exemplos positivos E_p e negativos E_n já rotulados. Cada exemplo $e_p \in E_p$ corresponde ao par $(\{d_i^{t-W+1}, ..., d_i^t\}\ ,\ d_i^{t+1})$, e ilustra a associação entre uma trajetória T_i e uma detecção d_i^{t+1} referentes ao mesmo objeto o_i. Como o custo deste tipo de associação é mínimo, cada exemplo e_p é rotulado com o valor $L_p = -1$. Em contrapartida, cada exemplo $e_n \in E_n$ é definido como o par $(\{d_i^{t-W+1}, ..., d_i^t\}\ ,\ d_j^{t+1}) \mid i \neq j$, o qual representa a associação entre uma trajetória T_i e uma detecção d_j^{t+1} referentes a objetos distintos o_i e o_j, respectivamente. Considerou-se que este tipo de associação tem custo máximo, de modo que cada exemplo negativo e_n é rotulado com o valor $L_n = 1$. Assim, o algoritmo de aprendizado teve como objetivo induzir uma função de regressão $r(\boldsymbol{g}_e)$ a partir da apresentação dos exemplos $E = E_p \cup E_n$, onde $e \in E$.

Tendo em vista que o vetor $\boldsymbol{f}$ apresenta baixa dimensionalidade e que os seus atributos correspondem a características com alto nível semântico, o modelo de regressão foi induzido neste trabalho a partir de uma rede neural do tipo MLP. A principal motivação para esta escolha

encontra-se na aplicação do modelo em cenários de rastreamento *online* e em tempo real. Sendo S a quantidade de atributos de $\boldsymbol{f}$, a rede foi projetada com $(W-1)\times S$ neurônios em sua camada de entrada L_0, Y neurônios em cada uma de suas H camadas escondidas $\{L_1, ..., L_H\}$ e um neurônio na camada de saída L_{H+1}. W, Y e H foram tratados como hiper-parâmetros a serem fixados durante as experimentações com o rastreador. Já a função de ativação aplicada ao final de cada camada $\{L_0, ..., L_H\}$ correspondeu à Unidade Linear Retificada (do inglês, *Rectified Linear Unit* ou ReLU), ao passo que sobre a saída da última camada L_{H+1} aplicou-se a função Tangente Hiperbólica (tanh), cuja imagem corresponde a $[-1, 1]$. A Figura 34 ilustra ambas as funções. Já a Figura 35 apresenta a arquitetura projetada para a MLP.

Figura 34 – Funções de ativação aplicadas sobre a saída de cada camada da rede MLP.

(a) Unidade Linear Retificada (ReLU). (b) Tangente hiperbólica (tanh).



Fonte: o próprio autor.

Para induzir a função de regressão $r(\boldsymbol{g}_e)$, a MLP foi treinada com base no algoritmo *Backpropagation* (CUN, 1988). Durante este processo, buscou-se minimizar a função de erro R a partir da apresentação de M exemplos $\{e_1, ..., e_M\}$. Esta função é definida como:

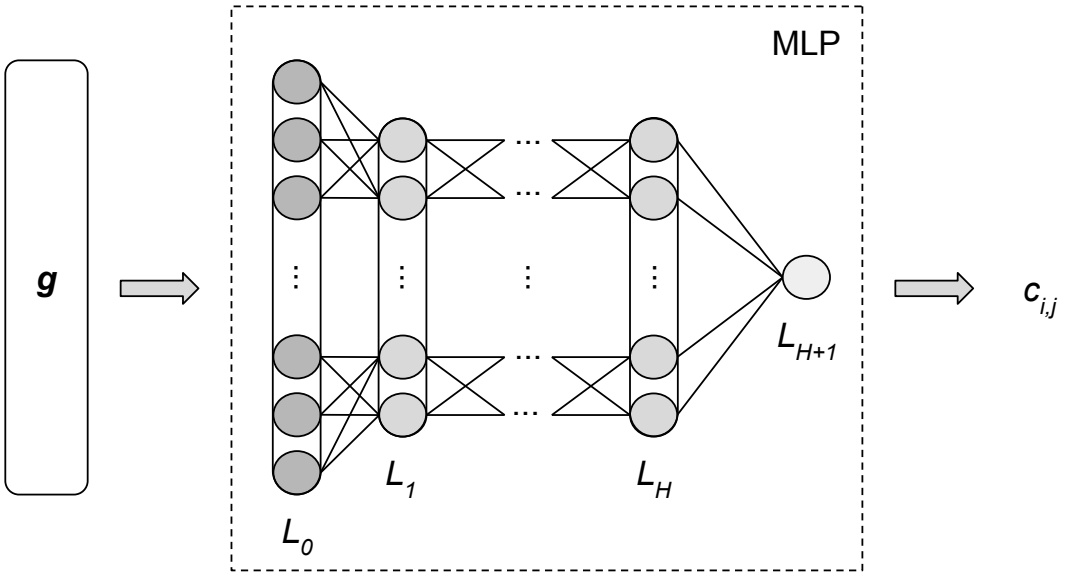
$$R(X, L) = \frac{1}{M}\sum h, \tag{4.4}$$

onde h é a função de erro de *Huber* (HUBER, 1964), dada por:

$$h = \begin{cases} \frac{1}{2}(X_i - L_i)^2, & |X_i - L_i| \leq 1 \\ |X_i - L_i| - \frac{1}{2}, & \text{caso contrário} \end{cases}, \tag{4.5}$$

sendo X um vetor de dimensão M onde cada elemento X_i corresponde ao valor de saída da função $r(\boldsymbol{g}_{e_i})$, e L um vetor também de dimensão M no qual cada elemento L_i refere-se ao rótulo do exemplo e_i. A função h, também conhecida como $L1$ suavizada, combina as funções de erro absoluto $L1(x) = |x|$ e de erro quadrático $L2(x) = x^2$, de maneira que para $|x| \leq 1$ seu gradiente permite um ajuste mais fino da função r do que o alcançado com $L1$, enquanto que o efeito de

Figura 35 – Ilustração da arquitetura projetada para a rede MLP. O vetor de características $\boldsymbol{g}$ é apresentado aos neurônios da camada inicial L_0. Suas saídas servem como entradas para os neurônios da camada seguinte L_1, os quais geram as entradas dos neurônios da próxima camada e assim sucessivamente até que se alcance a última camada L_{H+1}. Esta tem como função de ativação a Tangente Hiperbólica, de modo que sua saída corresponde ao custo de associação $c_{i,j} \in [-1, 1]$ entre a trajetória T_i e a detecção d_j.

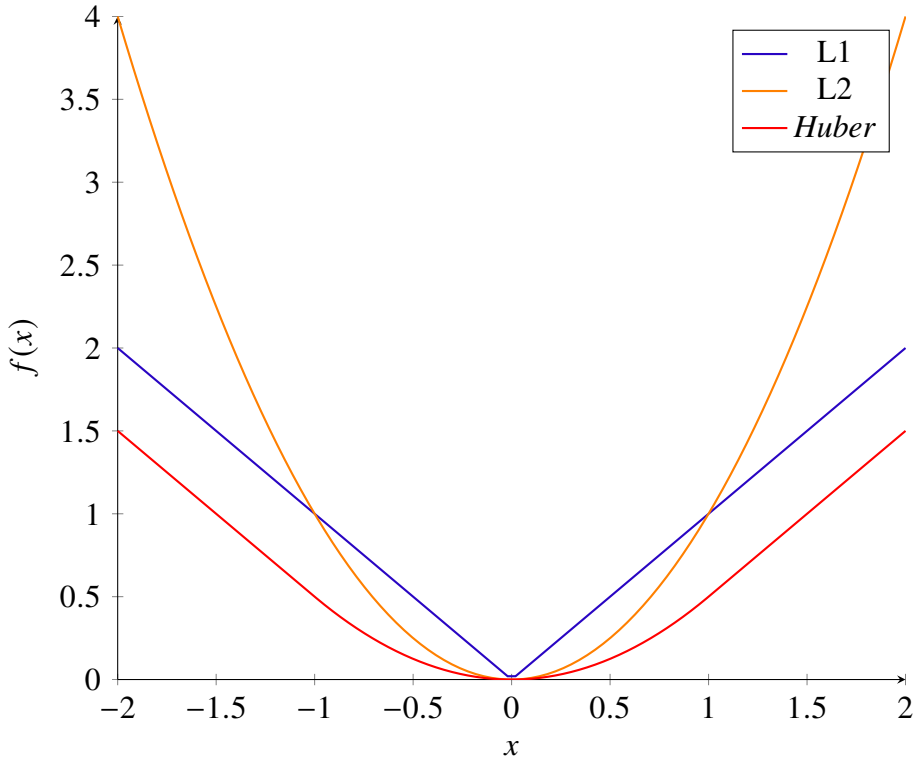


Fonte: o próprio autor.

exemplos anômalos (*i.e. outliers* para os quais $|x| > 1$) sobre o somatório de R é menor do que aquele gerado com $L2$ (GIRSHICK, 2015). A Figura 36 apresenta uma comparação entre tais funções.

Uma vez induzido, o modelo de regressão pode estimar numa única execução os custos das associações entre M pares de trajetórias $\{T_1, T_2, ...T_i, ...\}$ e detecções $\{d_1^{t+n}, d_2^{t+n}, ..., d_j^{t+n}, ...\}$ a partir do lote de vetores de características $B = [\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_M]^T$. O algoritmo 2 detalha o uso do modelo pelo SmartSORT durante sua etapa de estimação de custos (linha 5 do algoritmo 1).

Figura 36 – Comparação entre as funções de erro $L1$, $L2$ e *Huber*.



Fonte: o próprio autor.

Algoritmo 2: Estimação dos custos de associação.

Dados: Conjunto de trajetórias $\{T_1, T_2, ...T_i, ...\}$, conjunto de novas detecções D, limiar de custo C_{max}.
Resultado: Conjunto de custos de associações C entre trajetórias e novas detecções.

$B \leftarrow \emptyset$;
para cada $T \in \{T_1, T_2, ...T_i, ...\}$ **faça**
  para cada $d \in D$ **faça**
    $B \leftarrow B \cup \{\boldsymbol{g}_{T,d}\}$;
  fim
fim
$C \leftarrow modeloRegressao(B)$;
para cada $c \in C$ **faça**
  se $c > C_{max}$ **então**
    $C \leftarrow C - \{c\}$;
  fim
fim
retorna C;

5 Experimentos

O método SmartSORT foi avaliado a partir de experimentos realizados inicialmente em dois cenários distintos: o de rastreamento de pedestres e o de rastreamento de passageiros de ônibus. As seções a seguir discutem ambos os casos.

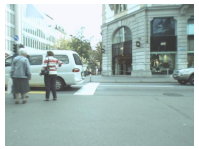
5.1 Rastreamento de Pedestres

A avaliação do método SmartSORT no contexto do rastreamento de pedestres foi conduzida através do *benchmark* MOT Challenge 2016 (MILAN et al., 2016). Este *benchmark* foi selecionado devido à sua utilização por grande parte dos trabalhos discutidos no Capítulo 2, e por corresponder a uma atualização do *benchmark* MOT Challenge 2015, a qual conta com sequências mais desafiadoras e atuais. As subseções apresentadas a seguir descrevem as etapas realizadas durante a avaliação, juntamente com os resultados obtidos.

5.1.1 Base de associações

A etapa inicial do processo de avaliação do método SmartSORT consistiu na criação de uma base de dados composta por exemplos de associações corretas e incorretas entre detecções. Para tanto, foram consideradas as sequências de treinamento do *benchmark* MOT Challenge 2016. Estas sequências correspondem a sete vídeos, que registram o deslocamento de múltiplos pedestres ao longo de diferentes cenários. A Tabela 4 apresenta mais detalhes a respeito de cada uma destas sequências. Ao todo, os sete vídeos contêm 517 trajetórias de pedestres identificados ao longo de 5316 imagens.

Tabela 4 – Descrição das sequências de treinamento do *benchmark* MOT Challenge 2016 (MILAN et al., 2016).

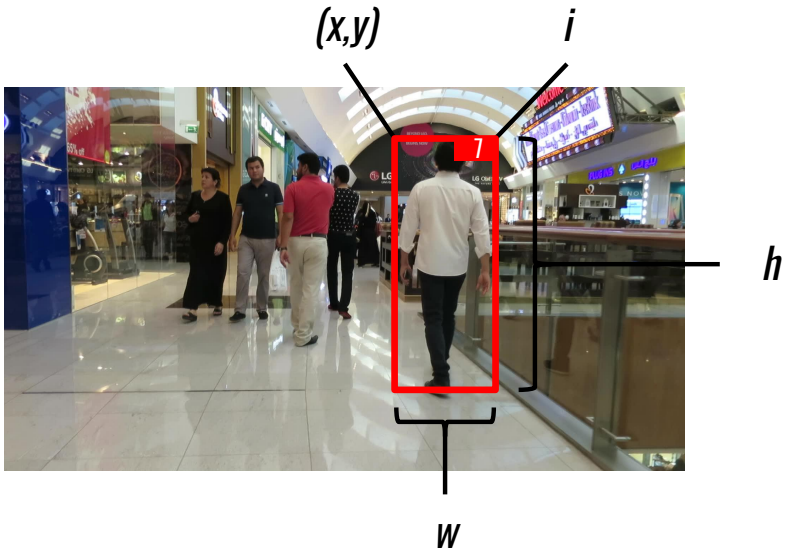
Amostra	Nome	FPS	Resolução	Imagens	Trajetórias	Marcações	Densidade	Descrição
	MOT16-13	25	1920x1080	750	107	11450	15,3	Câmera estática sobre ônibus.
	MOT16-11	30	1920x1080	900	69	9174	10,2	Câmera dinâmica em shopping.
	MOT16-10	30	1920x1080	654	54	12318	18,8	Câmera dinâmica num calçadão à noite.
	MOT16-09	30	1920x1080	525	25	5257	10,0	Câmera estática em frente a uma loja.
	MOT16-05	14	640x480	837	125	6818	8,1	Câmera dinâmica em calçada.
	MOT16-04	30	1920x1080	1050	83	47557	45,3	Câmera estática sobre rua à noite.
	MOT16-02	30	1920x1080	600	54	17883	29,7	Câmera estática em praça.

Além dos vídeos, este *benchmark* também disponibiliza um arquivo de anotações para cada sequência, o qual aponta por meio de marcações retangulares a localização, as dimensões e a identificação de cada pedestre ao longo das suas respectivas imagens. Mais especificamente, a marcação $M^t_{i,h}$ de um objeto o_i sobre a imagem I_t, a qual pertence à sequência S_h, possui o formato:

$$M^t_{i,h} = [t, i, x, y, w, h, q, c, v], \tag{5.1}$$

onde t representa o índice da imagem I_t, i corresponde ao identificador único do objeto o_i, (x, y) são as coordenas em duas dimensões do canto superior esquerdo da marcação, w corresponde à sua largura, h representa sua altura, q refere-se a uma variável de controle utilizada pelo *benchmark*, c indica a categoria do objeto o_i e v corresponde à sua taxa de visibilidade. A Figura 37 ilustra alguns dos atributos de uma marcação fornecida pelo *benchmark*.

Figura 37 – Ilustração de marcação retangular fornecida pelo *benchmark* MOT Challenge 2016 (MILAN et al., 2016). Dentre os atributos apresentados estão as coordenadas (u, v) do seu canto superior esquerdo, sua altura h, sua largura w e seu identificador i.



Fonte: o próprio autor.

O processo de criação da base de dados consistiu na amostragem estratificada uniforme de conjuntos com $W \geq 2$ marcações presentes ao longo de cada sequência de treinamento do *benchmark*. A metodologia de amostragem adotada é descrita pelo algoritmo 3. Para cada sequência $S_h \in \{S_1, S_2, ...\}$ são coletados os identificadores de todos os objetos $\{o_1, o_2, ..., o_i, ...\}$ contidos nas imagens de S_h (linha 4 do algoritmo 3). Em seguida, para cada objeto o_i seleciona-se o subconjunto de N imagens $\{I_1, ..., I_N\}$ da sequência S_h nas quais o_i está presente. Para cada $I_t \in \{I_1, ..., I_{N-W}\}$ seleciona-se um novo subconjunto ordenado de imagens $\{A_{k_1}, ..., A_{k_W}\}$, o qual é formado por I_t e por $W - 1$ amostras extraídas aleatoriamente de $\{I_{t+1}, ..., I_N\}$ (linha 8 do

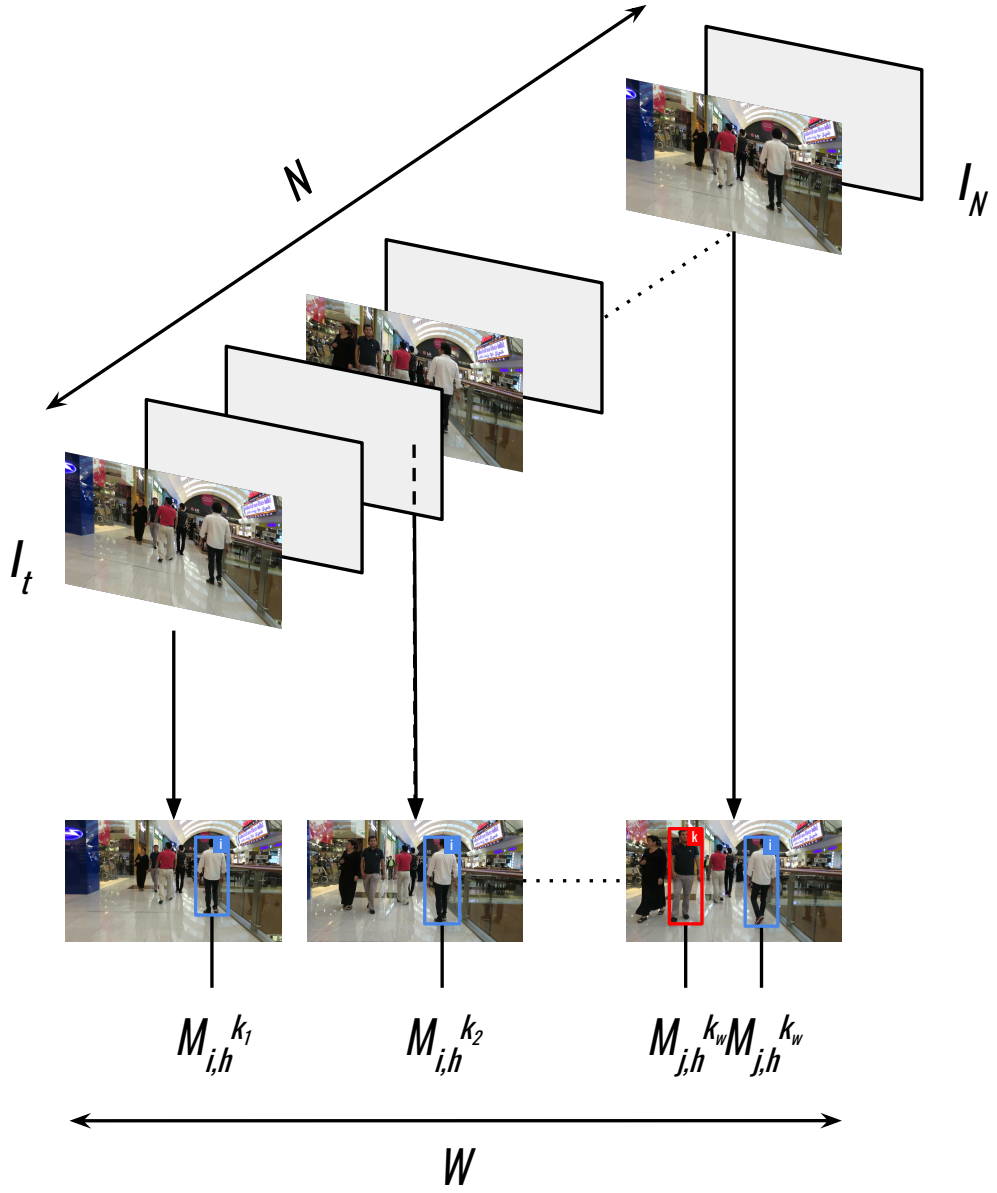
algoritmo 3). Em seguida, os exemplos de associações positivas e_p e negativas e_n são construídos a partir da concatenação das marcações $\{M_{i,h}^{k_1}, ..., M_{i,h}^{k_W}\}$, as quais referenciam o objeto o_i ao longo das imagens $\{A_{k_1}, ..., A_{k_W}\}$. A última marcação incluída no exemplo e_n, no entanto, tem o formato $M_{j,h}^{k_z}$, sendo $j \neq i$ o índice do objeto o_j presente na imagem A_{k_W}. Este índice é escolhido aleatoriamente com base nos objetos presentes na imagem A_{k_W}. Assim, ao final de cada iteração são gerados um exemplo positivo $e_p = \{M_{i,h}^{k_1}, ..., M_{i,h}^{k_W}\}$, o qual descreve a associação entre detecções referentes a um mesmo objeto o_i, e um exemplo negativo $e_n = \{M_{i,h}^{k_1}, ..., M_{i,h}^{k_{W-1}}\} + M_{j,h}^{k_W}$, onde este representa a associação entre detecções referentes a objetos distintos o_i e o_j. Os conjuntos de todos os exemplos positivos E_p e negativos E_n compõem a base $B_w = \{E_p, E_n\}$ gerada ao final do processo de amostragem. A Figura 38 ilustra este processo.

Algoritmo 3: Metodologia de amostragem de marcações.

Dados: Sequências $\{S_1, S_2, ...\}$, quantidade W de marcações por exemplo.
Resultado: Conjunto de exemplos positivos E_p, conjunto de exemplos negativos E_n.

```
1  E_p ← ∅;
2  E_n ← ∅;
3  para cada S_h ∈ {S_1, S_2, ...} faça
4      Ids ← identificadoresDosObjetos(S_h);
5      para cada i ∈ Ids faça
6          {I_1, ..., I_N} ← imagensComObjeto(i, S_h);
7          para cada I_t ∈ {I_1, ..., I_{N−W}} faça
8              {A_{k_1}, ..., A_{k_W}} ← {I_t} ∪ aleatorio({I_{t+1}, ..., I_N}, W − 1);
9              e_p ← ∅;
10             e_n ← ∅;
11             para cada A_{k_z} ∈ {A_{k_1}, ..., A_{k_{W−1}}} faça
12                 e_p ← e_p ∪ {M_{i,h}^{k_z}};
13                 se z ≠ W então
14                     e_n ← e_n ∪ {M_{i,h}^{k_z}};
15                 fim
16                 senão
17                     J ← identificadoresDosObjetos(A_{k_w});
18                     J ← J − {i};
19                     se J ≠ ∅ então
20                         j ← aleatorio(J);
21                         e_n ← e_n ∪ {M_{j,h}^{k_w}};
22                     fim
23                 fim
24             fim
25             E_p ← E_p ∪ e_p;
26             E_n ← E_n ∪ e_n;
27         fim
28     fim
29 fim
30 retorna E_p, E_n;
```

Figura 38 – Ilustração da metodologia adotada para amostragem de W marcações fornecidas pelo *benchmark* MOT Challenge 2016, onde $W \geq 2$. Dada uma sequência S_h composta pelo conjunto de imagens $\{I_1, ..., I_N\}$, seleciona-se o subconjunto de imagens $\{A_{k_1}, ..., A_{k_W}\}$, o qual é formado por I_t e por $W - 1$ amostras extraídas aleatoriamente de $\{I_{t+1}, ..., I_N\}$. O conjunto de marcações $e_p = \{M_{i,h}^{k_1}, ..., M_{i,h}^{k_W}\}$ é considerado um exemplo de associação entre detecções referentes a um mesmo objeto o_i ao longo das imagens $\{A_{k_1}, ..., A_{k_W}\}$. Por outro lado, o conjunto $e_n = \{M_{i,h}^{k_1}, ..., M_{i,h}^{k_{W-1}}\} + M_{j,h}^{k_W}$ é tratado como um exemplo de associação entre detecções referentes a objetos distintos o_i e o_j.

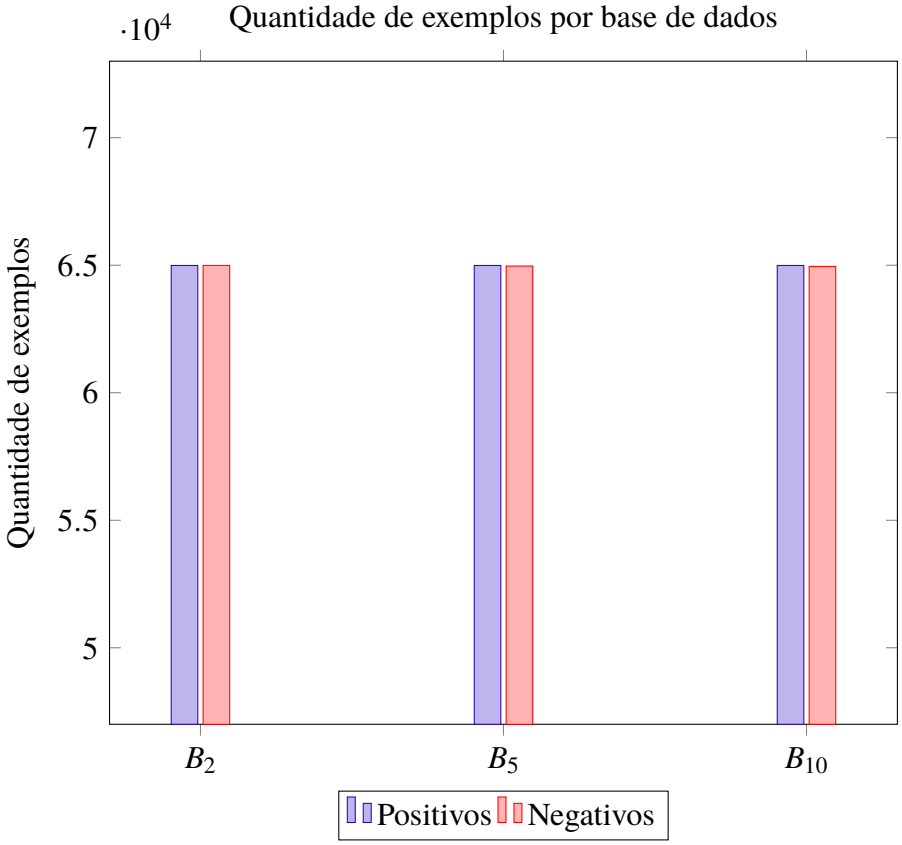


Fonte: o próprio autor.

A metodologia de amostragem descrita pelo algoritmo 3 permite a construção de bases de associações B_w com exemplos formados a partir da concatenação de W marcações, o que remete à estratégia de janela deslizante discutida no Capítulo 4. Tendo em vista que o comprimento

W da janela corresponde a um hiper-parâmetro a ser definido durante a preparação do método proposto, o algoritmo 3 foi utilizado para a criação de múltiplas bases $\{B_w\}$, $\forall\ W \in \{2, 5, 10\}$. A Figura 39 apresenta a distribuição de seus respectivos exemplos positivos e negativos. Nota-se que através da metodologia de amostragem estratificada foi possível gerar bases balanceadas para todos os valores de W considerados.

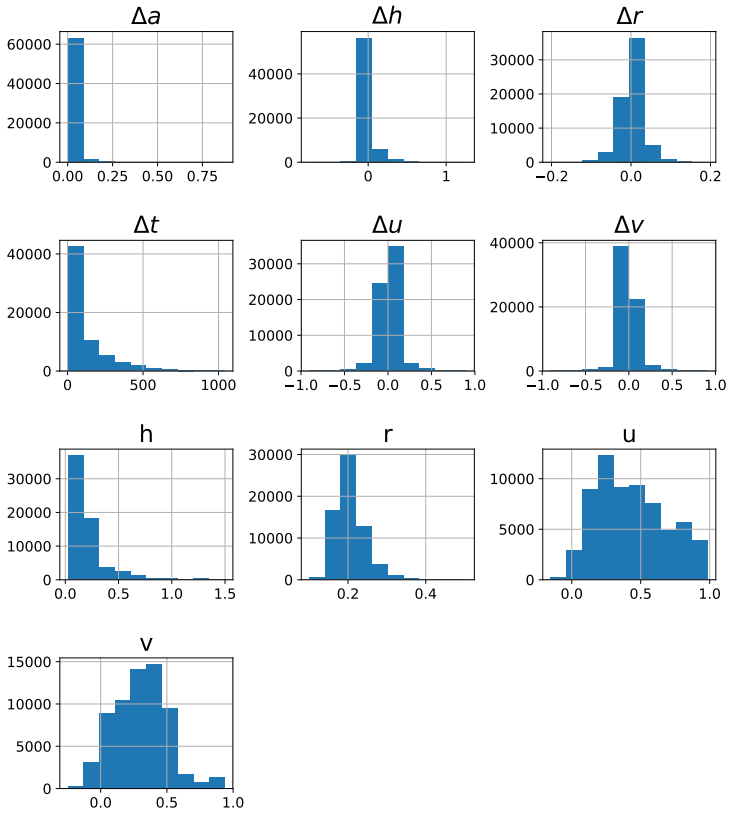
Figura 39 – Quantidade de exemplos positivos e negativos que compõem as bases de dados $\{B_w\}$, $\forall\ W \in \{2, 5, 10\}$, as quais foram construídas neste trabalho a partir da metodologia de amostragem estratificada descrita pelo algoritmo 3.



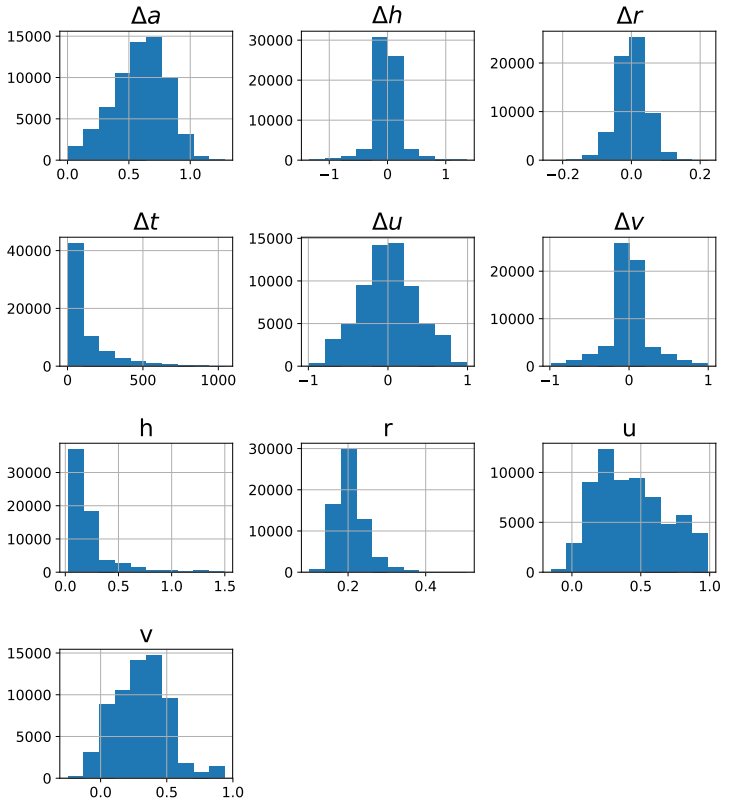
Finalmente, cada marcação $M_{i,h}^{k_z}$ pertencente às bases $\{B_2, B_5, B_{10}\}$ foi mapeada para uma detecção $d_i^{k_z} = [u, v, w, h]$, onde (u, v) corresponde às coordenadas do centro geométrico de $M_{i,h}^{k_z}$ sobre a imagem A_{k_z}, w é sua largura e h refere-se à sua altura. Os exemplos $E = E_p \cup E_n$ de cada base $B_w \in \{B_2, B_5, B_{10}\}$ passaram a ter o formato definido no Capítulo 4, onde $e_p \in E_p$ e $e_n \in E_n$ correspondem aos pares $(\{d_i^{k_1}, ..., d_i^{k_{W-1}}\}\ ,\ d_i^{k_W})$ e $(\{d_i^{k_1}, ..., d_i^{k_{W-1}}\}\ ,\ d_j^{k_W}) \mid i \neq k$, respectivamente. A partir desde novo formato foi possível extrair de cada exemplo $e \in E$ o vetor de características $\boldsymbol{g}$ utilizado como entrada pelo modelo de regressão. A Figura 40 ilustra a distribuição dos exemplos positivos E_p e negativos E_n da base de dados B_2 em detrimento de suas características. É possível notar que Δa, Δh, Δu e Δv referentes aos elementos de E_p possuem menor desvio padrão do que as respectivas características dos exemplos E_n. Além disso, percebe-se que as distribuições dos exemplos positivos e negativos em detrimento de seus respectivos Δt são semelhantes, o que demonstra novamente o balanceamento de B_2.

Figura 40 – Distribuição dos exemplos positivos E_p e negativos E_n da base de dados B_2 em detrimento de suas características.

(a) Exemplos positivos.



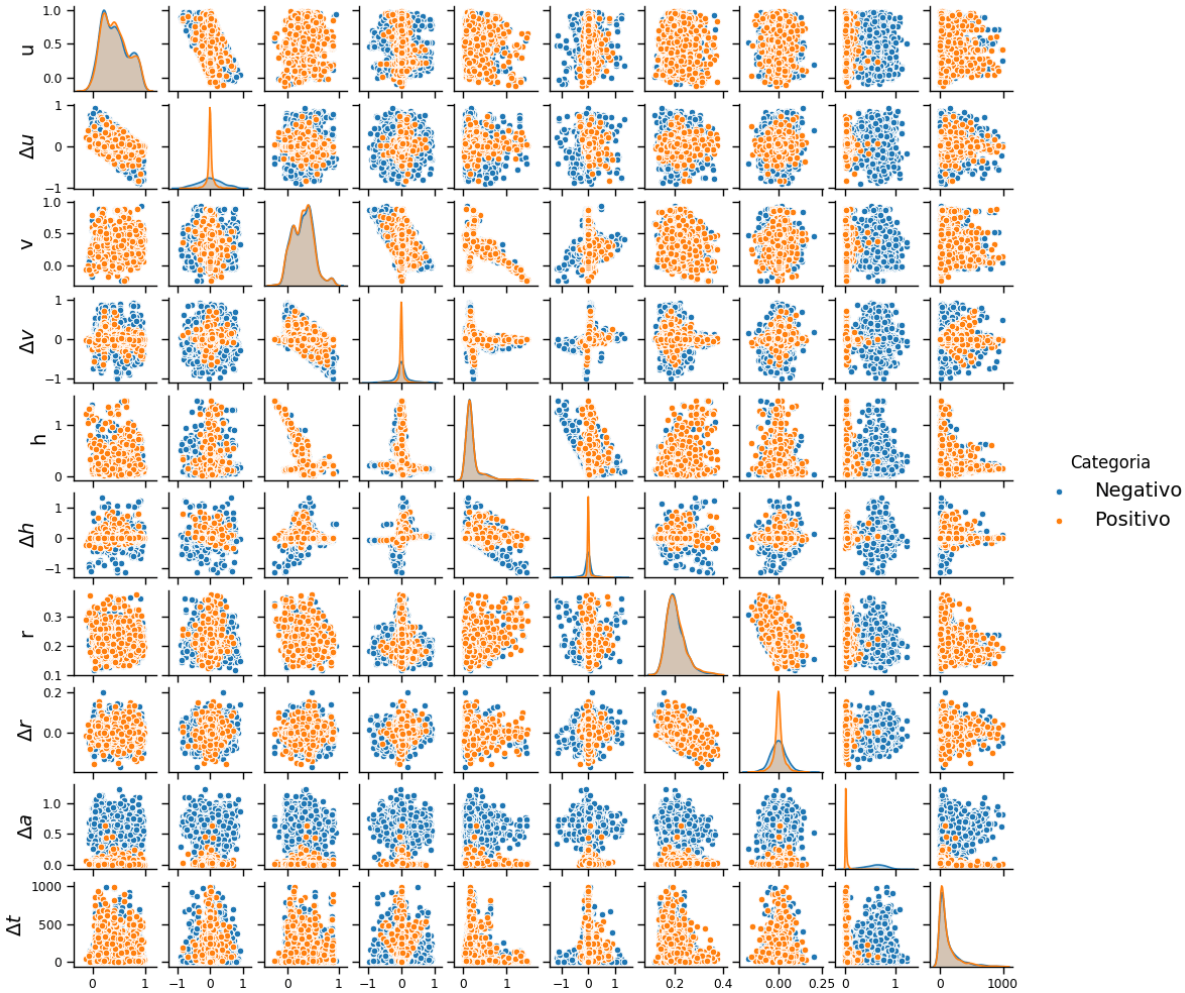
(b) Exemplos negativos.



Fonte: o próprio autor.

Já a Figura 41 ilustra a correlação par-a-par entre as características normalizadas dos exemplos de B_2. Percebe-se que mesmo características como u e v, sobre as quais as distribuições de E_p e E_n são semelhantes (Figura 40), apresentam níveis de correlação com outras variáveis que permitem a separação entre exemplos de categorias diferentes. Vale notar que a característica Δa apresentada tanto na Figura 40 quanto na Figura 41 foi extraída a partir de descritores de aparência obtidos por meio de uma CNN (WOJKE; BEWLEY; PAULUS, 2017).

Figura 41 – Gráfico de dispersão dos exemplos positivos E_p e negativos E_n contidos na base de dados B_2. A partir deste gráfico é possível visualizar a correlação par-a-par entre características extraídas de cada exemplo.



Fonte: o próprio autor.

5.1.2 Preparação do método

O protocolo de preparação do método de rastreamento consistiu na execução das seguintes etapas: 1) seleção dos hiper-parâmetros do modelo de regressão; 2) seleção dos hiper-parâmetros do método de SmartSORT; 3) indução do melhor modelo a partir de tais parâmetros. Para tanto, consideraram-se como entradas o conjunto de sequências de treinamento $\{S_1, S_2, ...\}$ do *benchmark* e o conjunto de bases $\{B_2, B_5, B_{10}\}$ obtido durante o processo de amostragem já

discutido na subseção 5.1.1. O algoritmo 4 descreve o protocolo de preparação. Inicialmente, o conjunto $\{S_1, S_2, ...\}$ é dividido em dois subconjuntos S^i e S^v. O primeiro é formado pelas sequências MOT-02, MOT-05, MOT-09 e MOT-13 do *benchmark*, enquanto que as sequências MOT-04, MOT-10, MOT-11 compõem S^v (linha 3 do algoritmo 4). Cada base $B_w \in \{B_2, B_5, B_{10}\}$ é então dividida em duas novas bases mutualmente excludentes B^t_w e B^v_w, formadas pelos exemplos de B_w extraídos das sequências S^i e S^v, respectivamente. Sobre B^t_w aplica-se um algoritmo de *grid search* (discutido mais à frente), o qual retorna os melhores hiper-parâmetros P_{melhor} encontrados para o modelo de regressão baseado em B^t_w. Através destes parâmetros e de todos os exemplos da base B^t_w, realiza-se a indução do modelo (linha 7 do algoritmo 4). Este é avaliado primeiramente sobre a base B^v_w em termos de acurácia como classificador. Em seguida, insere-se este modelo no SmartSORT e avalia-se sua qualidade como rastreador sobre as sequências de validação S^v. Após a realização deste processo para todas as bases $\{B_2, B_5, B_{10}\}$, seleciona-se o modelo m_{melhor} a partir do qual o rastreador registrou o maior valor de acurácia (linha 12 do algoritmo 4). Também seleciona-se a base de dados B_{melhor} utilizada para sua indução. Esta base é aleatoriamente dividida em duas partições mutualmente excludentes B^t_{melhor} e B^v_{melhor} (linha 14 do algoritmo 4), onde $\alpha = 9$ é a razão entre a quantidade de exemplos em B^t_{melhor} e B^v_{melhor}. O modelo m_{melhor} é então induzido a partir de B^t_{melhor}, sendo a base de validação B^v_{melhor} utilizada para determinar o critério de parada do processo de indução.

Algoritmo 4: Protocolo para preparação do método de rastreamento.

Dados: Conjunto de sequências $\{S_1, S_2, ...\}$, conjunto de bases de dados $\{B_2, B_5, B_{10}\}$, número de partições k, combinações P de hiper-parâmetros do modelo, razão α entre exemplos para indução e para validação.

Resultado: Modelo de regressão m_{melhor}.

1 $M \leftarrow \emptyset$;
2 **para cada** $B_w \in \{B_2, B_5, B_{10}\}$ **faça**
3 $\quad S^i, S^v \leftarrow divideInducaoValidacao(\{S_1, S_2, ...\})$;
4 $\quad B^i_w \leftarrow exemplosDasSequencia(B_w, S^i)$;
5 $\quad B^v_w \leftarrow exemplosDasSequencia(B_w, S^v)$;
6 $\quad P_{melhor} \leftarrow gridSearch(B^i_w, k, P)$;
7 $\quad m \leftarrow induz(B^i_w, P_{melhor})$;
8 $\quad avaliaModelo(m, B^v_w)$;
9 $\quad avaliaRastreador(m, S^v)$;
10 $\quad M \leftarrow M \cup \{m\}$;
11 **fim**
12 $m_{melhor} \leftarrow modeloMelhorRastreador(M)$;
13 $B_{melhor} \leftarrow baseMelhorRastreador(m_{melhor})$;
14 B^i_{melhor} , $B^v_{melhor} \leftarrow divideInducaoValidacao(B_{melhor}, \alpha)$;
15 $induz(m_{melhor}, B^i_{melhor}, B^v_{melhor})$;
16 **retorna** m_{melhor};

O algoritmo 5 descreve o protocolo utilizado para avaliar o rastreador SmartSORT com base num modelo m (linha 9 do algoritmo 4). Para cada sequência $S_j \in \{S_1, S_2, ...\}$ cria-se um conjunto de trajetórias $U = \{T_1, T_2, ..., T_i, ...\}$ e um arquivo de estados F. Sobre cada imagem $I_t \in S_j$ aplica-se um detector de pedestres, o qual retorna o conjunto de detecções D. Este conjunto é apresentado ao rastreador SmartSORT, juntamente com o conjunto de trajetórias U, o limiar de custo $C_{max} = 0$ e o valor máximo de perdas $L_{max} = 3$. Após o rastreador atualizar o conjunto U, são armazenados em F os estados $\{s_1, s_2, ..., s_i, ...\}$ dos seus respectivos objetos $\{o_1, o_2, ..., o_i, ...\}$ sobre a imagem I_t. Esse processo é repetido para todas as sequências $\{S_1, S_2, ...\}$. Ao final, suas marcações são coletadas e comparadas aos estados em F. O resultado desta comparação corresponde ao conjunto de métricas de avaliação A. Assim como Wojke, Bewley e Paulus (2017), as detecções fornecidas ao rastreador foram geradas por um *framework* Faster R-CNN (YU et al., 2016). Além disso, de modo similar àquele trabalho, a avaliação foi conduzida com valor máximo de perdas $L_{max} = 3$ e as detecções D foram limiarizadas com base no valor de confiança $c = 0.3$. Os descritores de aparência extraídos com base nas detecções foram obtidos por meio da aplicação de uma CNN (WOJKE; BEWLEY; PAULUS, 2017). Finalmente, o conjunto $\{A\}$ foi formado pelas métricas acurácia de rastreamento (MOTA) e troca de identidades (IDS) (BERNARDIN; STIEFELHAGEN, 2008).

Algoritmo 5: Protocolo para avaliação do rastreador.

Dados: Conjunto de sequências $\{S_1, S_2, ...\}$.
Resultado: Conjunto de métricas A.

```
1  A ← ∅;
2  para cada S_j ∈ {S_1, S_2, ...} faça
3      U ← ∅;
4      F ← ∅;
5      para cada I_t ∈ S_j faça
6          D ← detectaPedestres(I_t) ;
7          U ← smartSORT(D, U, C_max = 0, L_max = 3) ;
8          registraEstados(U, F) ;
9      fim
10 fim
11 M ← marcacoes({S_1, S_2, ...});
12 A ← compara(F, M);
13 retorna A;
```

Já o protocolo de *grid search* utilizado para selecionar os melhores hiper-parâmetros do modelo de regressão (linha 6 do algoritmo 4) é descrito pelo algoritmo 6. Inicialmente geram-se $k = 3$ partições mutualmente excludentes $\{B_w^1, B_w^2, B_w^3\}$ a partir da base de entrada B_w (linha 3 do algoritmo 6). Em seguida, para cada combinação P de hiper-parâmetros é conduzida a validação cruzada do tipo *k-fold*: para cada partição $B_w^i \in \{B_w^1, B_w^2, B_w^3\}$ realiza-se a indução do modelo de regressão com base nas partições $\{B_w^1, B_w^2, B_w^3\} - \{B_w^i\}$ e na combinação P. O modelo induzido m é avaliado sobre B_w^i (linha 8 do algoritmo 6) e sua acurácia é registrada. Ao fim da validação

k-fold, obtém-se a acurácia média a_{media} de m ao longo das k iterações (linha 11 do algoritmo 6). Todo este processo é repetido para cada combinação de parâmetros P. Ao final são obtidos os hiper-parâmetros do modelo m_{melhor} com maior valor de acurácia média a_{melhor} ao longo do processo de busca (linha 17 do algoritmo 6).

Algoritmo 6: Protocolo de *grid search* com validação cruzada *k-fold*.

Dados: Base de dados B_w, número de partições k, combinações P de hiper-parâmetros do modelo.
Resultado: Melhor conjunto de hiper-parâmetros P_{melhor}.

1 $m_{melhor} \leftarrow \emptyset$;
2 $a_{melhor} \leftarrow 0$;
3 $\{B_w^1, ..., B_w^k\} \leftarrow divideParticoes(B_w, k)$;
4 **para cada** $p \in P$ **faça**
5 $a_{media} \leftarrow 0$;
6 **para cada** $B_w^i \in \{B_w^1, ..., B_w^k\}$ **faça**
7 $m \leftarrow induz(\{B_w^1, ..., B_w^k\} - \{B_w^i\}, p)$;
8 $avalia(m, B_w^i)$;
9 $a_{media} \leftarrow a_{media} + acuracia(m)$;
10 **fim**
11 $a_{media} \leftarrow \frac{a_{media}}{k}$;
12 **se** $a_{media} > a_{melhor}$ **então**
13 $m_{melhor} \leftarrow m$;
14 $a_{media} \leftarrow a_{melhor}$;
15 **fim**
16 **fim**
17 $P_{melhor} \leftarrow parametros(m_{melhor})$;
18 **retorna** P_{melhor};

Como já discutido no Capítulo 4, para a implementação do método de rastreamento proposto considerou-se como modelo de regressão uma rede neural do tipo MLP. Dessa forma, os hiper-parâmetros buscados ao longo do processo de *grid search* correspondem ao número de camadas escondidas H e à quantidade de neurônios Y em cada uma destas camadas. Já o seu processo de indução ao longo de todo o protocolo de preparação do método consistiu na aplicação do algoritmo *Backpropagation* (FACELI et al., 2011). Os parâmetros utilizados para este algoritmo corresponderam à taxa $l = 0.002$ de aprendizado da rede, ao tamanho $b = 256$ do lote de exemplos apresentados à rede em cada iteração, ao momento $m = 0,9$ do otimizador baseado no algoritmo de gradiente descendente estocástico (RUDER, 2016) e ao seu número $e = 25$ de iterações (*i.e.* épocas). Este último parâmetro foi desconsiderado apenas durante a indução final do modelo m_{melhor} (linha 15 do algoritmo 4), na qual adotou-se como critério de parada o número $s = 5$ de épocas consecutivas em que o valor da acurácia do modelo sobre a base de validação B_{melhor}^v não aumenta. A utilização desta estratégia, conhecida como *early stopping* (PRECHELT, 2012), teve como objetivo impedir a superespecialização do modelo sobre a base de indução B_{melhor}^t.

É importante ressaltar que a escolha da acurácia como medida de avaliação ao longo de toda a preparação do método teve como principal justificativa a rotulação binária dos exemplos apresentados ao modelo. No cenário real de rastreamento o custo da associação entre a trajetória T_i do objeto o_i e a detecção d_j do objeto o_j, estimado pelo modelo, corresponde a um valor $c_{i,j}$ pertencente ao intervalo real $[-1, 1]$. No entanto, por questões de simplificação, os exemplos utilizados para sua indução foram rotulados como positivos e negativos. Muito embora estes rótulos tenham sido representados numericamente como -1 e 1, respectivamente, seus valores não refletem necessariamente o custo da associação entre T_i e d_j. Estes valores, na verdade, apenas informam se o_i e o_j são suficientemente similares para que a associação entre T_i e d_j aconteça. Dessa forma, considerou-se mais prudente durante a indução do modelo avaliá-lo em termos de sua acurácia como classificador binário, sendo sua saída considerada -1 sempre que $c_{i,j} \leq 0$ e 1, caso contrário. Vale notar que esta estratégia é válida apenas durante a indução do modelo, já que a computação de custos binários ao longo do rastreamento torna impraticável a etapa de associação realizada pelo SmartSORT via o Método Húngaro.

A Tabela 5 apresenta os melhores hiper-parâmetros encontrados ao final da execução do *grid search*. Já a Tabela 6 disponibiliza os resultados obtidos após a validação dos modelos $\{m_2, m_5, m_{10}\}$ e do rastreador baseado nestes modelos. É possível notar que a relação de superioridade entre os modelos a partir dos seus respectivos valores de acurácia não é a mesma que aquela baseada na qualidade final do rastreador. Uma vez consideradas as métricas deste último, o modelo m_{melhor} selecionado correspondeu a uma MLP com 40 neurônios de entrada (referentes a 10 características ao longo de uma janela com tamanho $W = 5$) e apenas uma camada escondida com 7 neurônios. Durante sua indução final, baseada na estratégia *early stopping*, foram atingidas 33 épocas, com acurácia de $98,59\%$ sobre a base de validação B^v_{melhor}.

Tabela 5 – Resultados obtidos ao longo da busca de hiper-parâmetros para o modelo de regressão através de *grid search* com validação cruzada *k-fold*.

Modelo	Melhores hiper-parâmetros	↑ Acurácia Média
m_2	H = 1, Y = 7	96,11%
m_5	H = 1, Y = 7	96,42%
m_{10}	H = 1, Y = 32	96,27%

Tabela 6 – Resultados obtidos ao longo do processo de validação dos modelos induzidos.

Modelo	**Validação do Modelo**		**Validação do Rastreador**	
	↓ Erro Absoluto	↑ Acurácia	↑ Acurácia (MOTA)	↓ Troca de Identidades (IDS)
m_2	2,92E-01	97,79%	59,2%	294
m_5	2,88E-01	97,60%	59,2%	250
m_{10}	2,66E-01	98,15%	59,2%	278

5.1.3 Avaliação do método

Com base no modelo de regressão m_{melhor} obtido ao final de sua preparação, o método SmartSORT foi avaliado sobre as sequências de teste do *benchmark* MOT Challenge 2016. O protocolo adotado foi semelhante ao já apresentado pelo algoritmo 5, assim como as ferramentas e os hiper-parâmetros definidos durante a preparação do método. Nesta avaliação, a qualidade do rastreador foi aferida a partir das métricas CLEAR (BERNARDIN; STIEFELHAGEN, 2008), as quais correspondem a:

- Acurácia do rastreamento de múltiplos objetos (MOTA) - acurácia geral do rastreador em termos de troca de identidades, falsos positivos e falsos negativos;
- Precisão do rastreamento de múltiplos objetos (MOTP) - precisão das marcações dos objetos previstas pelo rastreador;
- Majoritariamente rastreados (MT) - porcentagem de trajetórias cobertas pelo rastreador ao longo de pelo menos 80% do seu tempo de vida;
- Majoritariamente perdidos (ML) - porcentagem de trajetórias cobertas pelo rastreador ao longo de no máximo 20% do seu tempo de vida;
- Troca de identidades (IDS) - total de vezes em que um identificador diferente foi atribuído a uma mesma trajetória;
- Fragmentação (FM) - total de vezes em que o registro de uma trajetória foi interrompido pelo rastreador.

5.1.4 Resultados

A Figura 42 compara os resultados do método de rastreamento proposto sobre o *benchmark* MOT Challenge 2016 contra a performance de sua principal *baseline* em termos de acurácia e velocidade de execução. Já que ambos os rastreadores compartilham da mesma rotina de extração de descritores de aparência, os valores de velocidade reportados não consideram o tempo levado para a execução daquela tarefa.

A Tabela 7 apresenta os resultados gerais do método proposto sobre o *benchmark*, como também a performance de sua *baseline* direta e de outros rastreadores submetidos ao mesmo desafio. Entre estes encontram-se métodos baseados na modelagem de movimentação através do Filtro de Kalman e de filtro de partículas (SORT e EA-PHD-PF), e na modelagem de aparência através de redes neurais profundas (POI, CNNMT e RAN). Finalmente, a Figura 43 ilustra alguns resultados qualitativos do rastreador proposto sobre o *benchmark*.

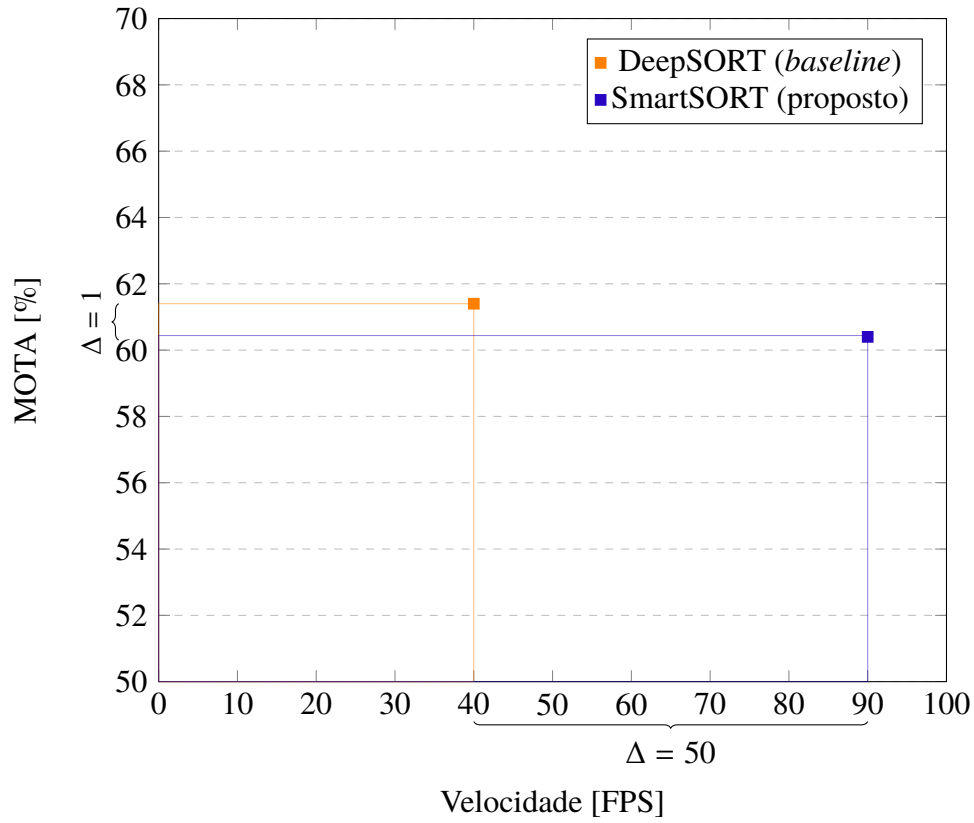
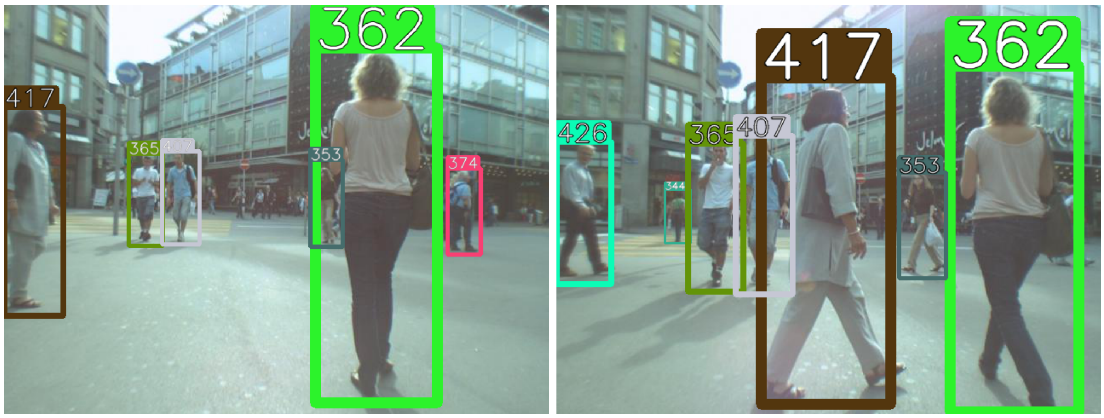


Figura 42 – Comparação entre as performances do método proposto e de sua *baseline* sobre o *benchmark* MOT Challenge 2016, em termos de acurácia de rastreamento *versus* velocidade de execução. As velocidades apresentadas não consideram a etapa de extração de descritores de aparência, já que a mesma é realizada de maneira equivalente por ambos os métodos.

Tabela 7 – Desempenho de rastreadores *online* sobre o *benchmark* MOT Challenge 2016. Todos os métodos listados utilizaram detectores próprios.

	↑ **MOTA**	↑**MOTP**	↑**MT**	↓**ML**	↓**IDS**	↓**FM**	↑**Velocidade**
RAN (FANG; XIANG; SAVARESE, 2017)	63,0	78,8	33,9%	22,1%	**482**	**1251**	1,6 FPS
CNNMT (MAHMOUDI; AHADI; RAHMATI, 2018)	65,2	78,4	32,4%	21,3%	946	2283	11 FPS
EA-PHD-PF (SANCHEZ-MATILLA; POIESI; CAVALLARO, 2016)	52,5	78,8	19,0%	34,9%	910	1321	12 FPS
POI (YU et al., 2016)	**66,1**	79,5	**34,0%**	20,8%	805	3093	10 FPS
IOU (BOCHINSKI; EISELEIN; SIKORA, 2017)	57,1	77,1	23,6%	32,9%	2167	3028	**3000 FPS**
SORT (BEWLEY et al., 2016)	59,8	**79,6**	25,4%	22,7%	1423	1835	60 FPS
DeepSORT (WOJKE; BEWLEY; PAULUS, 2017)	61,4	79,1	32,8%	18,2%	781	2008	17 FPS
SmartSORT (proposto)	60,4	78,9	21,9%	**16,1%**	1135	2230	27 FPS

Figura 43 – Resultados qualitativos do método de rastreamento proposto sobre a sequência de teste MOT16-06 do *benchmark* MOT Challenge 2016.



(a) Resultado sobre a imagem 972. (b) Resultado sobre a imagem 990.

Fonte: o próprio autor.

5.1.5 Discussão

De acordo com a Figura 42, o rastreador SmartSORT foi capaz de executar a uma frequência de 90 FPS contra 40 FPS alcançados por sua *baseline* (WOJKE; BEWLEY; PAULUS, 2017). Este resultado representa um ganho de 125% em velocidade de processamento ao custo de 1% em acurácia. Este ganho pode ser explicado pela computação em lote realizada pelo modelo de regressão, o qual calcula o custo da associação entre todos os pares de detecções atualmente observadas e trajetórias atualmente mantidas numa única execução. Ao mesmo tempo, o método SmartSORT não realiza a aplicação de filtro algum, ao contrário de sua *baseline*. Desse modo, sua rotina de gerenciamento de trajetórias tem menor custo computacional.

O mesmo resultado é confirmado quando comparados o rastreador SmartSORT e os demais métodos apresentados na Tabela 7. Mesmo considerando o tempo levado para extrair os descritores de aparência das detecções, o SmartSORT executa mais rápido que a maioria dos rastreadores considerados. As únicas exceções são os métodos IOU e SORT, cujas taxas de troca de identidade são, no mínimo, 25% mais altas que a apresentada pelo SmartSORT. Por outro lado, este último executa a uma frequência no mínimo 59% mais alta que a daqueles métodos que utilizam redes neurais profundas para a discriminação de detecções.

O ganho em velocidade do rastreador SmartSORT é ainda maior quando comparado ao método RAN, o qual computa a similaridade entre detecções através de uma única rede neural profunda do tipo LSTM (SHERSTINSKY, 2018). Este resultado demonstra que apesar do método proposto basear-se numa rede de arquitetura rasa, através da apresentação de características de alto nível já pré-processadas, é possível induzir uma função de regressão que estime a similaridade entre detecções e que seja executada significantemente mais rápido do que uma rede de arquitetura mais complexa. Ao mesmo tempo, sua acurácia de rastreamento é menos de 3%

inferior à apresentada pelo método RAN.

Apesar da acurácia e da velocidade do método SmarSORT serem competitivos, sua taxa de troca de identidades (IDS) e de fragmentação de trajetórias (FM) foram, respectivamente, 45% e 11% mais altas que aquelas apresentadas por sua *baseline*. Considerando que ambos os métodos empregam o mesmo extrator de características profundas e que o último aplica o Filtro de Kalman para modelar a movimentação dos objetos rastreados, este resultado pode sugerir que o principal ponto falho do modelo de regressão proposto está relacionado à estratégia de janela deslizante para predizer a trajetória de um objeto com base em suas posições passadas. Mais precisamente, esta abordagem é vulnerável a associações incorretas: uma vez que o rastreador associa uma trajetória T_i a uma detecção d_j, onde T_i e d_j estão respectivamente relacionadas a objetos distintos o_i e o_j, sua janela deslizante é contaminada com características de aparência e de movimentação de o_j. Este ruído pode trazer instabilidade para o modelo, a qual persiste até que a janela seja preenchida apenas com características de novas detecções associadas corretamente a T_i. Alternativas para solucionar este problema incluem aprimorar o protocolo de treinamento do modelo, de modo a considerar a apresentação de exemplos positivos e_p ruidosos, e a substituição da arquitetura da rede MLP por uma rede neural recorrente rasa (*vanilla* RNN), a qual é mais adequada para a indução de modelos baseados em dados sequencias, como é o caso do rastreamento.

De todo modo, os resultados obtidos ao final da submissão do rastreador SmartSORT para o *benchmark* MOT Challenge 2016 demonstram que o modelo de regressão induzido permite o desenvolvimento de métodos para o rastreamento *online* de pedestres cujo custo-benefício final em termos de acurácia e velocidade é competitivo.

5.2 Rastreamento de Passageiros de Ônibus

A avaliação do método SmartSORT no contexto do rastreamento de passageiros de ônibus foi conduzida através da base de dados BUS Challenge 2018, construída ao longo deste trabalho (Apêndice A). Sua escolha teve como principal motivação a escassez deste tipo de cenário dentre os *benchmarks* de rastreamento de múltiplos objetos (Capítulo 2), somada às suas condições visuais adversas ao rastreamento, como altas taxas de variação de luz e de oclusão de pessoas. Assim, pôde-se comparar o método SmartSORT à sua principal *baseline*, o algoritmo DeepSORT (WOJKE; BEWLEY; PAULUS, 2017). As subseções apresentadas a seguir descrevem as etapas realizadas durante a avaliação, juntamente com os resultados obtidos.

5.2.1 Base de associações

Inicialmente, construiu-se uma base de dados a partir de exemplos de associações corretas e incorretas entre detecções. Nesta etapa, foram utilizadas as sequências de treinamento da base de dados BUS Challenge 2018, cujas características são apresentadas pela Tabela 8. Ao

Tabela 8 – Descrição das sequências de treinamento da base de dados BUS Challenge 2018.

Amostra		
Nome	BUS18-03	BUS18-01
FPS	4	4
Resolução	352x240	352x240
Imagens	3601	3601
Trajetórias	15	7
Marcações	16123	14476
Densidade	4,5	4,0
Descrição	Câmera sobre a catraca com vista para o corredor.	Câmera sobre o cobrador com vista para a catraca.

todo, estas sequências contém 22 trajetórias de passageiros e cobradores de ônibus ao longo de 7202 imagens. Estas sequências correspondem a dois vídeos já rotulados, cujas marcações são definidas pela Equação 5.1, já apresentada na subseção anterior. Assim como naquele caso, utilizou-se o algoritmo 3 para a criação das bases de dados $\{B_2, B_3, B_5, B_7, B_{10}\}$. A Figura 44 ilustra a distribuição dos exemplos ao longo de cada base gerada. É possível notar que todas as bases encontram-se balanceadas.

Após a amostragem, cada marcação $M_{i,j}^{t_z}$ pertencente às bases $\{B_2, B_3, B_5, B_7, B_{10}\}$ foi mapeada para uma detecção $d_i^{t_z} = [u, v, w, h]$, de modo semelhante ao descrito na subseção 5.1.1. A Figura 45 ilustra a distribuição dos exemplos positivos E_p e negativos E_n da base de dados B_2 em detrimento de suas características. A princípio, nota-se a semelhança entre as distribuições dos exemplos positivos e negativos em detrimento de seus respectivos Δt, o que demonstra novamente o balanceamento de B_2. Além disso, as distribuições de ambos os exemplos em detrimento das características u e v denunciam, respectivamente, a baixa taxa de ocupação da região central referente à catraca e o maior nível de concentração de pessoas na região superior das imagens, na qual se localizam os assentos mais ao fundo do veículo. Por fim, percebe-se que as características Δa, Δh, Δu, Δv e Δr referentes aos elementos de E_p possuem menor desvio padrão do que as respectivas características dos exemplos E_n.

Já a Figura 46 ilustra a correlação par-a-par entre as características normalizadas dos exemplos de B_2. Nota-se que quando combinados os gráficos de correlação permitem a separação entre exemplos de categorias diferentes, com destaque para aqueles relacionados às distâncias Δu, Δv, Δh e Δa. Vale notar que esta última, apresentada tanto na Figura 40 quanto na Figura 41, foi extraída a partir de descritores de aparência obtidos por meio de uma CNN (WOJKE; BEWLEY; PAULUS, 2017).

Figura 44 – Quantidade de exemplos positivos e negativos que compõem as bases de dados $\{B_2, B_3, B_5, B_7, B_{10}\}$, construídas neste trabalho a partir da metodologia de amostragem estratificada descrita pelo algoritmo 3 sobre as sequências de treinamento da base de dados BUS Challenge 2018.

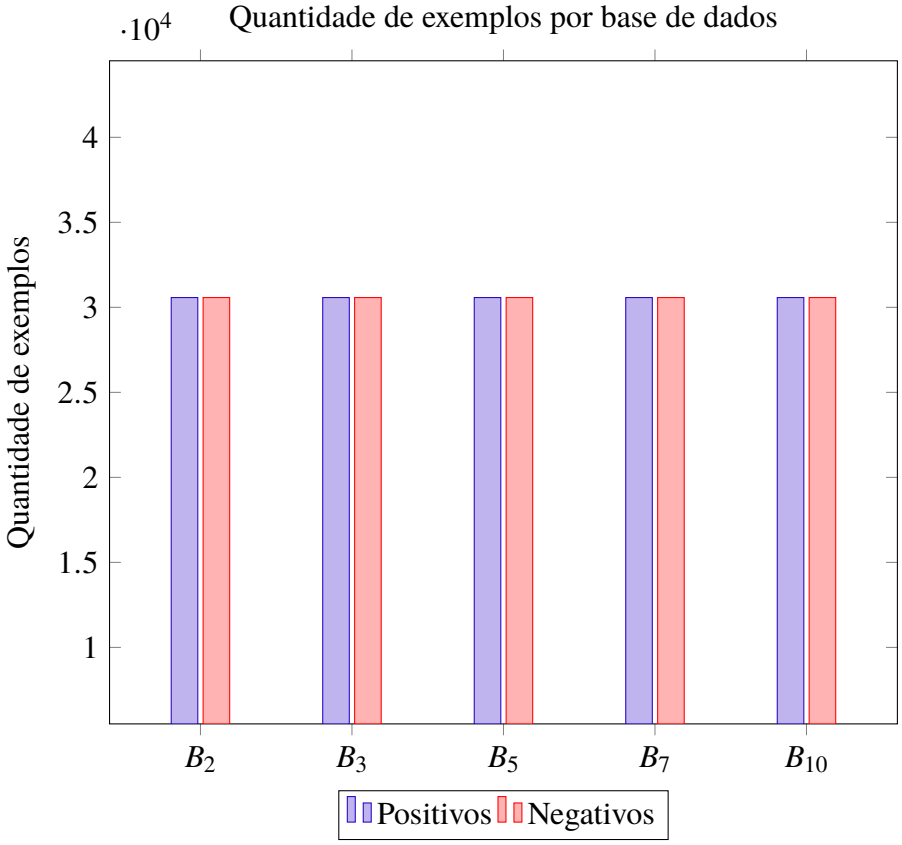


Figura 45 – Distribuição dos exemplos positivos $\{e_p\}$ e negativos $\{e_n\}$ da base de dados B_2 em detrimento de suas características.

(a) Exemplos positivos.

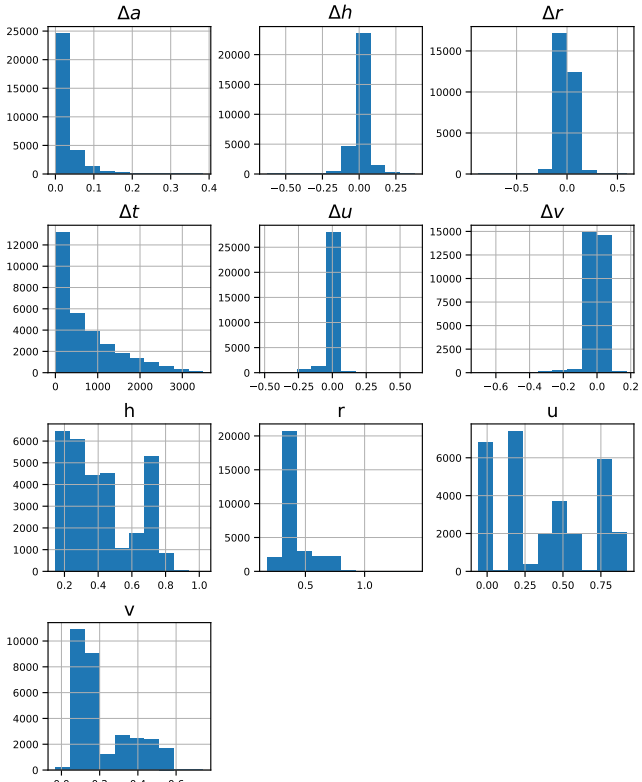
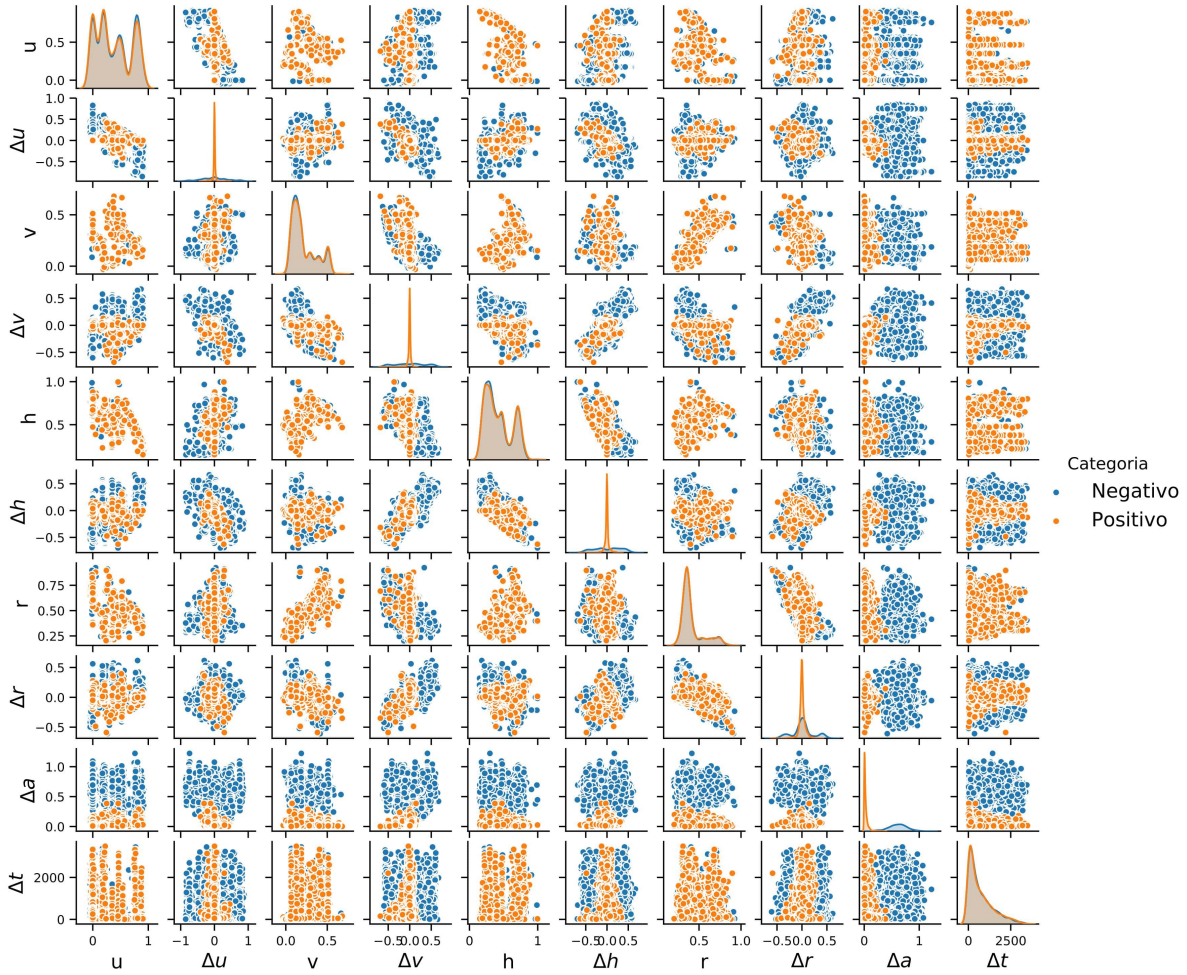


Figura 46 – Gráfico de dispersão dos exemplos positivos $\{e_p\}$ e negativos $\{e_n\}$ contidos na base de dados B_2. A partir deste gráfico é possível visualizar a correlação par-a-par entre características extraídas de cada exemplo.



Fonte: o próprio autor.

5.2.2 Preparação do método

A preparação do método SmartSORT para rastrear passageiros de ônibus foi inspirada no protocolo descrito pelo algoritmo 4, já utilizado durante o experimento de rastreamento de pedestres (seção 5.1). No entanto, devido à quantidade reduzida de sequências de treinamento da base de dados BUS Challenge 2018, a preparação do método para este experimento não considerou a avaliação do *rastreador* sobre nenhuma base de validação. O algoritmo 7 descreve seu protocolo. Inicialmente, sobre cada base B_w obtida durante o processo de amostragem descrito na subseção 5.2.1 aplica-se um algoritmo de *grid search* (já detalhado pelo algoritmo 6). Este algoritmo retorna os melhores hiper-parâmetros P_{melhor} encontrados para um modelo de regressão baseado em B_w. Em seguida, esta é aleatoriamente dividida em duas partições mutualmente excludentes B_w^t e B_w^v (linha 4 do algoritmo 7), onde $\alpha = 9$ é a razão entre a quantidade de exemplos em B_w^t e B_w^v. Através dos hiper-parâmetros P_{melhor} e de todos os exemplos da base B_w^t,

realiza-se a indução do modelo m (linha 5 do algoritmo 4). Durante cada etapa de sua indução, a acurácia a de m como classificador é mensurada sobre os exemplos da base de validação B_w^v. A partir da técnica de *early stopping*, o critério de parada para este processo consiste na quantidade de iterações consecutivas $s = 5$ em que a não apresenta melhoria. Após a indução, registra-se o valor de a e armazena-se o modelo m (linha 6 do algoritmo 7). Este processo é repetido para todas as bases $\{B_2, B_3, B_5, B_7, B_{10}\}$. Ao final, seleciona-se o modelo m_{melhor} que registrou o maior valor de acurácia.

Algoritmo 7: Protocolo de preparação do método SmartSORT para o rastreamento de passageiros de ônibus.

Dados: Conjunto de bases de dados $\{B_2, B_3, B_5, B_7, B_{10}\}$, número de partições k, combinações P de hiper-parâmetros do modelo, razão α entre exemplos para indução e para validação.
Resultado: Modelo de regressão m_{melhor}.

1 $M \leftarrow \emptyset$;
2 **para cada** $B_w \in \{B_2, B_3, B_5, B_7, B_{10}\}$ **faça**
3 $\quad P_{melhor} \leftarrow gridSearch(B_w, k, P)$;
4 $\quad B_w^i , B_w^v \leftarrow divideInducaoValidacao(B_w, \alpha)$;
5 $\quad m \leftarrow induz(B_w^i, B_w^v, P_{melhor})$;
6 $\quad M \leftarrow M \cup \{m\}$;
7 **fim**
8 $m_{melhor} \leftarrow melhorModelo(M)$;
9 **retorna** m_{melhor};

Assim como no experimento com pedestres (seção 5.1), a implementação do SmartSORT considerou um modelo de regressão induzido por uma rede MLP. Logo, os hiper-parâmetros P buscados durante o processo de *grid-search* corresponderam ao seu número de camadas escondidas H e à quantidade de neurônios Y em cada uma destas camadas. Sua indução foi alcançada por meio da aplicação do algoritmo *Backpropagation*, durante a qual utilizaram-se como parâmetros a taxa $l = 0,002$ de aprendizado da rede, o tamanho $b = 256$ do lote de exemplos apresentados à rede em cada iteração e o momento $m = 0,9$ do otimizador baseado no algoritmo de gradiente descendente estocástico (RUDER, 2016).

A Tabela 5 apresenta os melhores hiper-parâmetros encontrados ao final da execução do *grid search*. Já a Tabela 6 disponibiliza os resultados obtidos após a validação dos modelos $\{m_2, m_3, m_5, m_7, m_{10}\}$. Com base nestes resultados, e considerando o erro médio absoluto como critério de desempate, selecionou-se como melhor modelo m_{melhor} uma MLP com 40 neurônios de entrada (referentes a 10 características ao longo de uma janela com tamanho $W = 5$) e apenas uma camada escondida com 17 neurônios. Durante sua indução final foram atingidas 24 épocas, com acurácia de $99,66\%$ sobre a base de validação B_{melhor}^v.

Tabela 9 – Resultados obtidos ao longo da busca de hiper-parâmetros para o modelo de regressão através de *grid search* com validação cruzada *k-fold*.

Modelo	Melhores hiper-parâmetros	↑ Acurácia Média
m_2	H = 1, Y = 11	96,64%
m_3	H = 1, Y = 9	96,63%
m_5	H = 1, Y = 17	96,67%
m_7	H = 1, Y = 53	96,56%
m_{10}	H = 1, Y = 7	99,60%

Tabela 10 – Resultados obtidos ao longo do processo de validação dos modelos induzidos.

Modelo	↓ Erro Médio Absoluto	↑ Acurácia
m_2	8,19E-02	99,62%
m_3	8,72E-02	99,57%
m_5	**7,81E-02**	**99,66%**
m_7	8,49E-02	99,62%
m_{10}	8,89E-02	99,66%

5.2.3 Avaliação do método

O método SmartSORT foi avaliado sobre as sequências de teste da base de dados BUS Challenge 2018. Para isso, utilizou-se o modelo de regressão m_{melhor} obtido ao final de sua preparação. Além disso, a avaliação foi conduzida com valor máximo de perdas $L_{max} = 3$ e com descritores de aparência extraídos por meio da aplicação de uma CNN (WOJKE; BEWLEY; PAULUS, 2017). Já as detecções D apresentadas como entrada para o SmartSORT foram fornecidas pela própria base de dados. O protocolo adotado para avaliar o método foi o mesmo já apresentado pelo algoritmo 5, sendo sua qualidade aferida por meio das métricas CLEAR (BERNARDIN; STIEFELHAGEN, 2008). Ao final, estas métricas foram comparadas às alcançadas por meio da aplicação do rastreador DeepSORT (WOJKE; BEWLEY; PAULUS, 2017), principal *baseline* do método proposto. Ambos os métodos foram executados com base nas mesmas condições (*hardware* e *software*), sendo utilizado o código original do DeepSORT fornecido por seus autores [1].

5.2.4 Resultados

A Tabela 11 apresenta os resultados quantitativos obtidos ao final da aplicação do método SmartSORT sobre as sequências de teste da base de dados BUS Challenge 2018. Também são apresentadas métricas referentes à performance de sua *baseline* sobre as mesmas sequências. Como ambos os métodos compartilham da mesma rotina de extração de características visuais, esta não foi considerada para o cálculo de suas respectivas velocidades de execução.

Já a Figura 47 e a Figura 48 ilustram alguns dos resultados qualitativos obtidos após a execução do rastreador SmartSORT.

[1] <https://github.com/nwojke/deep_sort>

Tabela 11 – Desempenho de rastreadores *online* sobre a base de dados BUS Challenge 2018. Todos os métodos listados utilizaram detecções fornecidas pela própria base. As velocidades apresentadas não consideram a etapa de extração de descritores de aparência realizada por ambos os métodos.

	↑ **MOTA**	↑**MOTP**	↑**MT**	↓**ML**	↓**IDS**	↓**FM**	↑**Velocidade**
DeepSORT (*baseline*)	**99,2%**	93,6%	**48,0%**	**0,0%**	**3**	**4**	545 FPS
SmartSORT (proposto)	**99,2%**	**96,5%**	46,0%	**0,0%**	27	5	**778 FPS**

Figura 47 – Resultados qualitativos do método de rastreamento SmartSORT sobre as sequências de teste da base de dados BUS Challenge 2018.

(a) Resultado sobre a imagem 3542 da sequência BUS18-02.

(b) Resultado sobre a imagem 3569 da sequência BUS18-02.

(c) Resultado sobre a imagem 370 da sequência BUS18-04.

(d) Resultado sobre a imagem 545 da sequência BUS18-04.

Fonte: o próprio autor.

5.2.5 Discussão

A partir dos resultados apresentados pela Tabela 11 percebe-se que a acurácia (MOTA) do SmartSORT foi equivalente à apresentada pelo DeepSORT, o que representa um avanço em relação ao resultado observado no experimento referente ao rastreamento de pedestres (Tabela 7). Ambos os métodos apresentaram acurácia de $99,2\%$, o que reflete a qualidade das detecções fornecidas pela base de dados e a eficiência dos rastreadores para operar com base em tais detecções. Além disso, de modo semelhante àquele experimento, neste o SmartSORT foi capaz de executar a uma velocidade $42,8\%$ maior que a do DeepSORT. Novamente, este

Figura 48 – Ilustração de troca de identidades cometida pelo método SmartSORT sobre a sequência de teste BUS18-04 da base de dados BUS Challenge 2018.

(a) Resultado sobre a imagem 877. (b) Resultado sobre a imagem 878.

Fonte: o próprio autor.

ganho em performance pode ser creditado à computação em lote de seu regressor em contraste à aplicação do filtro de Kalman pelo DeepSORT. Também é possível notar com base na Tabela 11 a superioridade do SmartSORT no que se refere à precisão (MOTP) dos estados estimados pelo rastreador em termos de posicionamento e dimensões, sendo seu valor $2,9\%$ superior ao apresentado pelo DeepSORT.

Apesar de equiparar-se ao DeepSORT em termos de acurácia, o SmartSORT apresentou uma quantidade de troca de identidades (IDS) 9 vezes maior que a de sua *baseline*, como aponta a Tabela 11. Durante a execução do experimento, percebeu-se que a maior parte destas trocas ocorreu em situações pontuais nas quais os passageiros deslocavam-se para o fundo do veículo. Nestes casos a baixa resolução da imagem e as condições adversas de iluminação dificultam a discriminação de passageiros com base em suas aparências, de modo que suas características de movimentação tornam-se mais relevantes. Dessa forma, assim como no experimento referente ao rastreamento de pedestres (seção 5.1), pode-se inferir que a alta troca de identidades por parte do SmartSORT é justificada pela sua incapacidade de prever a posição futura de objetos com base na sua movimentação, cujos motivos correspondem aos já discutidos na seção 5.1.

De todo modo, a partir do experimento realizado foi possível verificar a adaptabilidade do método SmartSORT para um contexto de rastreamento específico. Além disso, pôde-se novamente constatar seu alto custo benefício em termos de acurácia *versus* velocidade de execução quando comparado ao seu principal trabalho relacionado.

6

Estudo de Caso: Contagem Automática de Passageiros de Ônibus

De modo a avaliar a performance do algoritmo SmartSORT numa aplicação real baseada no rastreamento de objetos, foi conduzido neste trabalho um estudo de caso concernente à contagem automática de passageiros de ônibus. As subseções apresentadas a seguir discutem a motivação deste estudo, sua metodologia e os resultados alcançados.

6.1 Motivação

Cerca de 45% dos deslocamentos urbanos da população brasileira no trânsito são realizados através de ônibus. É o que aponta a pesquisa Mobilidade da População Urbana 2017 (CNT; NTU, 2017), realizada pela Confederação Nacional do Transporte (CNT) em parceria com a Associação Nacional das Empresas de Transportes Urbanos (NTU). Segundo a pesquisa, além do ônibus ser o meio de transporte urbano predominante no Brasil, o mesmo também detém a maior representatividade dentre os transportes coletivos, de modo que sua participação encontra-se na ordem de $52,7\%$. Já dados divulgados pela NTU referentes a fevereiro de 2018 (NTU, 2018) revelam que o transporte por ônibus atendeu a mais de 30 milhões de passageiros pagantes por dia (totalizando aproximadamente 40 milhões de passageiros, quando considerados os $20,1\%$ de gratuidades), ao longo de 3313 municípios brasileiros. De acordo com a associação, para atender à tamanha demanda, o Brasil conta com 1800 empresas operadoras de ônibus, cuja frota total equivale a 107 mil veículos.

Apesar do expressivo mercado em que atuam, quase 30% das empresas de transporte coletivo urbano em todo o Brasil possuem dívidas que superam em média 40% dos seus faturamentos anuais (NTU, 2017). Tais dívidas devem-se principalmente à queda do número de passageiros ao longo dos anos, o que implica na redução da venda de bilhetes, principal fonte de financiamento do setor. Esta queda, por sua vez, pode ser justificada, dentre outros motivos, à insatisfação de mais de 70% dos passageiros com a qualidade do serviço prestado (FGV, 2014).

Em meio às principais reivindicações encontram-se o alto valor das tarifas cobradas, o tempo elevado das viagens e a sensação de insegurança.

Além das dificuldades relacionadas à diminuição da demanda de passageiros, além dos desafios decorrentes dos diversos custos operacionais associados ao setor (FNP; ANTP, 2017), as empresas operadoras de ônibus ainda precisam lidar com as perdas no faturamento oriundas de fraudes. Estas, por sua vez, não são uma exclusividade do setor de transporte: segundo um relatório divulgado pela consultoria Kroll (KROLL, 2018), aproximadamente 84% das empresas entrevistadas em todo o mundo relataram perdas oriundas de fraudes, sendo que 54% reportaram prejuízos equivalentes a 4% ou mais de seu faturamento anual. Ainda assim, o cenário do setor de transporte urbano brasileiro apresenta-se ainda pior: estima-se que as empresas operadoras de ônibus perdem, em média, 10% de todo o seu faturamento anual em decorrência de fraudes. Tal perda é significativa, considerando-se o atual estado de endividamento das operadoras, além da necessidade de investir na melhoria dos seus serviços.

Nesse sentido, assim como observado nos demais setores da indústria e do comércio (KROLL, 2018), empresas de transporte têm recorrido a ferramentas de cunho tecnológico para a identificação e a prevenção de diversos tipos de fraudes: desde a implantação de sistemas de bilhetagem eletrônica, os quais auxiliam no combate à evasão de receita (Da Silva, 2017), à utilização de ferramentas que identificam o uso indevido de gratuidades por parte de passageiros através de biometria facial (G1, 2017), até ao uso de sistemas de contagem automática de passageiros (ACOREL, 2017), os quais permitem que as operadoras reconheçam o transporte de passageiros sem o devido pagamento e também tenham acesso a informações comercialmente relevantes concernentes a sua clientela (*e.g.*, horários de pico, rotas mais lucrativas). Devido a sua relevância, tais sistemas contadores, inclusive, são adotados por empresas de diferentes setores além do de transporte, como promotoras de eventos (INSTANTCOUNTING, 1998) e lojas de varejo (V-COUNT, 2017), além de museus e bibliotecas (Retail Sensing, 2018).

Sistemas para a contagem automática de pessoas, em geral, utilizam tecnologias baseadas em sensores de infra-vermelho, tapetes sensíveis à pressão e câmeras (ELKOSANTINI; DARMOUL, 2013). Sistemas baseados neste último tipo de sensor apresentam como vantagem a possibilidade de aproveitamento das câmeras já utilizadas para monitorar o ambiente em que estão instaladas, haja vista o número significativo de espaços públicos e comerciais que já as utilizam como ferramenta de segurança. Esta vantagem, inclusive, é relevante no âmbito do transporte, uma vez que no Brasil o tamanho médio da frota das operadoras equivale a aproximadamente 60 veículos (NTU, 2018).

Na literatura científica, é possível encontrar referências a diferentes tipos de soluções de contagem automática que utilizam câmeras. As mais tradicionais baseiam-se em técnicas clássicas de visão monocular, como segmentação por subtração de plano de fundo (Heng-Xin Chen; Bin Fang; Yuan-Yan Tang, 2007), segmentação gaussiana (XIANG-YANG; HAO-WEI, 2016), detecção de bordas (Sihua Ye; Jiancong Wang, 2010), transformada circular de Hough

(MUKHERJEE et al., 2011) e histograma de fluxo óptico (ESCOLANO et al., 2016). A principal vantagem destas soluções está no fato de serem compatíveis com as câmeras monoculares usualmente já instaladas em ônibus urbanos. No entanto, tais técnicas são mais sensíveis a variações na iluminação do cenário, o que compromete a acurácia da contagem realizada em ambientes ruidosos, como o do transporte urbano.

Uma alternativa às técnicas de visão monocular corresponde ao uso de soluções baseadas em visão estereoscópica. Estas apresentam maior robustez a variações de luminosidade, haja vista que utilizam câmeras capazes de mensurar os diferentes níveis de profundidade do cenário monitorado e, com base na limiarização desta informação, segmentam as pessoas a serem contabilizadas (BERNINI et al., 2014). No entanto, sua principal desvantagem encontra-se no alto custo de aquisição de tais câmeras, cuja natureza diferencia-se daquelas usualmente já instaladas nos veículos de transporte urbano.

Como já discutido ao longo deste trabalho, nos últimos anos grandes avanços na área de visão computacional monocular puderam ser observados graças à aplicação bem-sucedida de técnicas de Aprendizado de Máquina, mais especificamente de algoritmos de Aprendizado Profundo (LECUN; BENGIO; HINTON, 2015). Devido à sua qualidade e robustez, estes algoritmos representam o atual estado-da-arte em diferentes aplicações concernentes ao processamento e à interpretação de imagens e vídeos. Dentre tais aplicações, em especial, encontra-se o rastreamento de múltiplos objetos (MILAN et al., 2016).

Dessa forma, percebe-se a oportunidade de avaliar o uso do algoritmo de rastreamento desenvolvido neste trabalho para a realização da contagem automática de passageiros de ônibus através do monitoramento das suas trajetórias ao longo do veículo. A partir deste algoritmo, seria possível desenvolver uma ferramenta capaz de auxiliar empresas operadoras a identificar fraudes referentes ao uso do transporte sem o devido pagamento e também fornecer informações relevantes a respeito do fluxo de passageiros ao longo das diversas linhas ofertadas. Finalmente, tal solução não apresentaria restrições quanto à natureza da câmera, uma vez que o algoritmo de rastreamento considera como entrada imagens em duas dimensões. Assim, câmeras previamente instaladas nos veículos poderiam ser aproveitadas por tal ferramenta, de modo a evitar custos com a instalação e a manutenção de novos sensores.

6.2 Metodologia

O estudo de caso consistiu na implementação e avaliação de uma ferramenta de contagem automática de passageiros baseada no algoritmo de rastreamento SmartSORT, desenvolvido ao longo deste trabalho. Para isso, utilizaram-se os mesmos hiper-parâmetros considerados durante o experimento descrito na seção 5.2 do Capítulo 5, além daquele mesmo regressor já treinado a partir da base de dados BUS Challenge 2018 (Apêndice A). O algoritmo 8 descreve o funcionamento desta ferramenta. A mesma recebe como entrada um conjunto de

N imagens $\{I_1, ..., I_N\}$. Este conjunto corresponde a um vídeo obtido a partir de uma câmera de monitoramento instalada no interior do veículo. Além disso, a ferramenta também recebe como entrada a tupla $F = (x_0, y_0, x_1, y_1)$, a qual informa os pontos extremos (x_0, y_0) e (x_1, y_1) do segmento de reta que define uma fronteira de contagem. Para cada imagem $I_t \in \{I_1, ..., I_N\}$ do vídeo, aplica-se um detector de pessoas, o qual retorna o conjunto de detecções D (linha 5 do algoritmo 8). Com base neste conjunto, executa-se o algoritmo SmartSORT, o qual atualiza o conjunto de trajetórias $U = \{T_1, T_2, ..., T_i, ...\}$ monitoradas (linha 6 do algoritmo 8). Finalmente, verifica-se o cruzamento da fronteira de contagem por cada objeto o_i com trajetória T_i (linha 7 do algoritmo 8). Esta verificação é realizada com base nas coordenadas dos centros geométricos das duas últimas detecções adicionadas a T_i. Caso o objeto o_i tenha cruzado a fronteira F na direção denominada de avanço (linha 8 do algoritmo 8), incrementa-se o valor da contagem acumulada c. Caso o cruzamento tenha ocorrido na direção oposta (linha 11 do algoritmo 8), decrementa-se o valor da contagem. Por fim, este valor é registrado (linha 14 do algoritmo 8). Ao final de sua execução, a ferramenta gera como saída um conjunto C formado por registros da contagem acumulada ao longo de todo o vídeo.

Algoritmo 8: Funcionamento da ferramenta de contagem automática de passageiros.

Dados: Conjunto $\{I_1, ..., I_N\}$ de N imagens de um vídeo, fronteira F de contagem.
Resultado: Conjunto C de registros da contagem acumulada ao longo de todo o vídeo.

```
1  c ← 0;
2  C ← ∅;
3  U ← ∅;
4  para cada I_t ∈ {I_1, ..., I_N} faça
5      D ← detectaPessoas(I_t) ;
6      U ← smartSORT(D, U) ;
7      para cada T_i ∈ U faça
8          se avancouFronteira(T_i, F) então
9              c ← c + 1;
10         fim
11         senão se recuouFronteira(T_i, F) então
12             c ← c − 1;
13         fim
14         C ← C ∪ {c};
15     fim
16 fim
17 retorna C;
```

Durante este estudo de caso, a ferramenta de contagem foi executa sobre um único vídeo fornecido pela empresa de transporte urbano Auto Viação Modelo[1], o qual foi obtido através de uma câmera de monitoramento instalada no interior de um dos seus veículos. O vídeo em questão

[1] Mais detalhes acerca da empresa podem ser encontrados através do seu endereço eletrônico: <http://www.viacaomodelo.com.br/institucional.php>

foi capturado no dia 31 de agosto de 2018, entre os horários 12:00:31 e 13:00:31, totalizando exatamente 1 hora de duração. Além disso, o mesmo é composto por imagens com resolução $1080x720$ *pixels*, as quais foram capturados a uma taxa de 4 imagens por segundo. A Figura 49 ilustra o vídeo utilizado neste estudo. É possível notar que a câmera encontra-se posicionada sobre a catraca, com vista tanto para o cobrador quanto para o fundo do veículo. Esta escolha deve-se a uma demanda interna da própria empresa fornecedora do vídeo, a qual demonstrou interesse em registrar com maior resolução o fluxo de passageiros que atravessam a catraca ao longo do tempo. Dessa forma, a fronteira fornecida à ferramenta correspondeu a um segmento de reta horizontal posicionado na altura da catraca (destacada em vermelho na Figura 49), de maneira que a atualização da contagem ocorresse com base em cruzamentos na direção vertical da imagem.

Figura 49 – Ilustração do vídeo utilizado como entrada durante o estudo de caso de contagem automática de passageiros. Também é possível visualizar o segmento de reta horizontal (em vermelho) artificialmente projetada sobre ambos os quadros, a qual foi utilizada como fronteira pela ferramenta de contagem implementada.

(a) Primeiro quadro do vídeo.

(b) Último quadro do vídeo.

Fonte: o próprio autor.

Para avaliar a ferramenta de contagem durante este estudo, foi gerado um registro de referência formado pela contagem acumulada real de passageiros e pelos instantes em que este valor foi atualizado. Para tanto, assistiu-se ao vídeo de entrada e coletaram-se os horários em que a catraca do veículo foi rotacionada. Além disso, realizou-se a contagem manual de rotações da catraca ao longo de todo o vídeo. Finalmente, os valores desta contagem foram associados aos horários coletados. A Figura 50 ilustra o registro de referência obtido. Nota-se que neste registro foram incluídos os valores da contagem acumulada nos instantes inicial e final do vídeo. Esta inclusão foi motivada pelo interesse em detectar-se a ocorrência de falsas atualizações da contagem por parte da ferramenta avaliada ao longo de todo o vídeo.

O protocolo utilizado para avaliar a ferramenta é descrito pelo algoritmo 9. Para cada valor da contagem acumulada real c_{ref}^{t+1} armazenada no registro de referência C_{ref} e relacionado ao instante $t+1$ do vídeo, calcula-se o seu incremento Δc_{ref} em relação ao valor da contagem de referência c_{ref}^{t} no instante anterior t (linha 3 do algoritmo 9). Em seguida, seleciona-se o valor da contagem acumulada c_{t+1} registrada pela ferramenta no instante $t+1$ e calcula-se seu incremento Δc em relação ao valor da contagem c^t registrado pela ferramenta no instante t (linha 4 do algoritmo 9). Finalmente, atualiza-se o conjunto de erros E com base na diferença absoluta entre os incrementos Δc e Δc_{ref} (linha 5 do algoritmo 9). Nota-se, assim, que a qualidade da ferramenta é aferida com base no incremento da sua contagem entre cada instante do vídeo em que foi registrada uma rotação na catraca. Ao não analisar diretamente o valor da contagem, esta estratégia busca evitar a ocorrência de distorções na avaliação, as quais poderiam ocorrer devido ao acúmulo de erro ao longo da contagem real de passageiros. Além disso, por observar os valores da contagem acumulada apenas nos instantes em que foram registradas rotações reais na catraca, reduzem-se ruídos causados pela movimentação de pessoas próximas à fronteira de contagem.

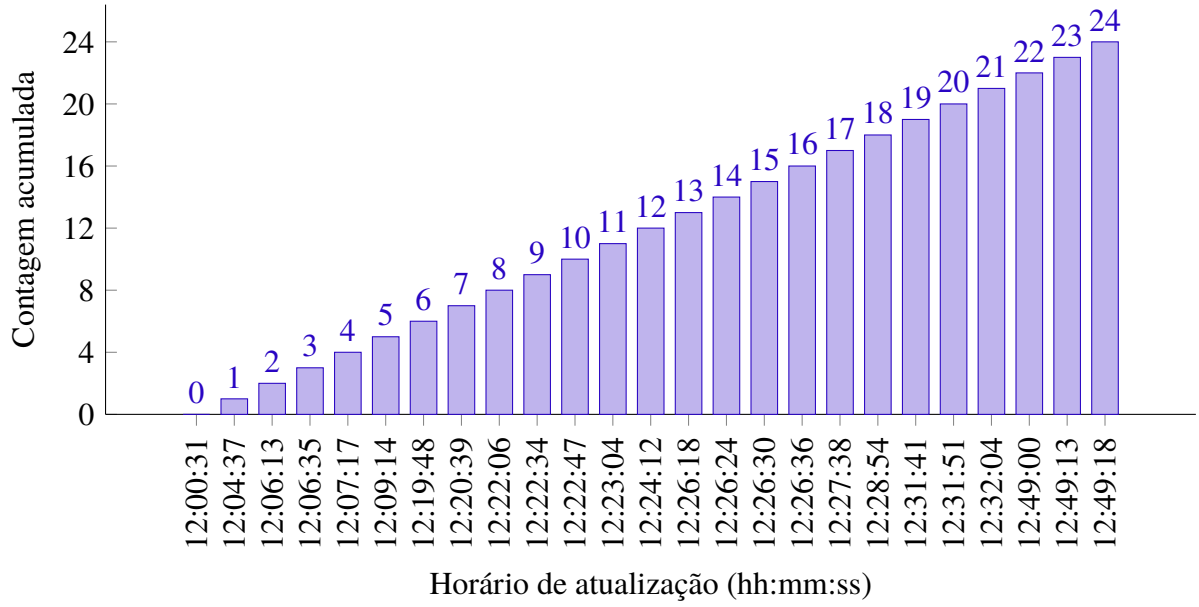
Algoritmo 9: Protocolo de avaliação da ferramenta de contagem.

Dados: Conjunto C de registros gerados pela ferramenta, conjunto C_{ref} de registros reais de referência.
Resultado: Conjunto E de erros no incremento da contagem computada pela ferramenta.

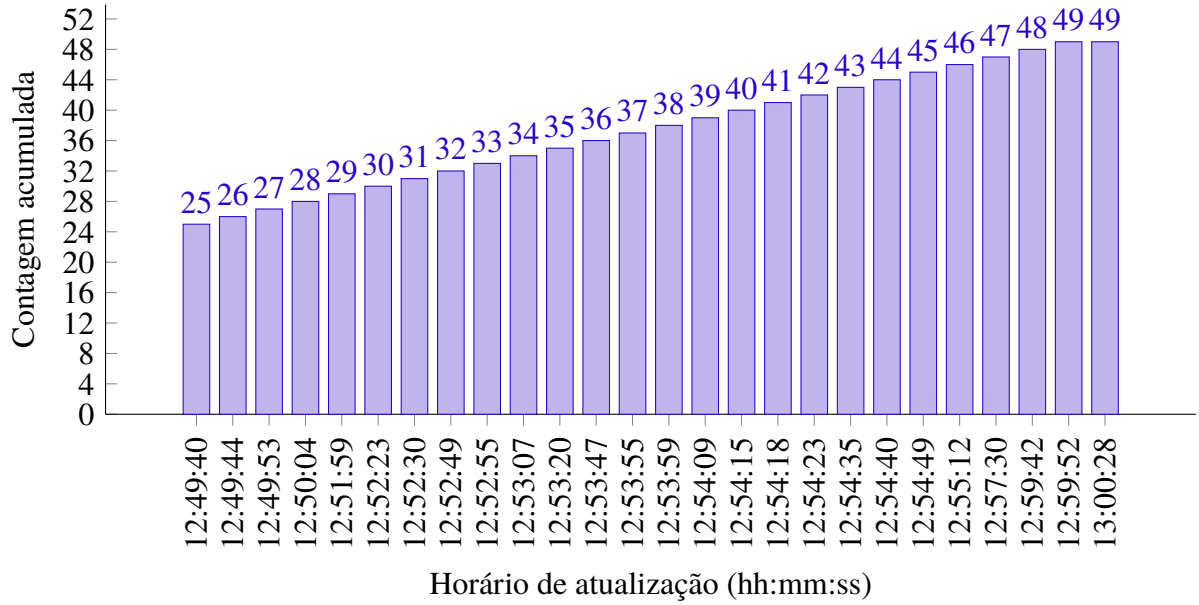
1 $E \leftarrow \emptyset$;
2 **para cada** $c_{ref}^{t+1} \in C_{ref}$ **faça**
3 $\quad \Delta c_{ref} = c_{ref}^{t+1} - c_{ref}^{t}$;
4 $\quad \Delta c = c^{t+1} - c^{t}$;
5 $\quad E \leftarrow E \cup \{|\Delta c - \Delta c_{ref}|\}$;
6 **fim**

Este estudo também considerou a execução da ferramenta de contagem com base no algoritmo de rastreamento DeepSORT (WOJKE; BEWLEY; PAULUS, 2017), já utilizado como *baseline* para o método SmartSORT durante os experimentos discutidos no Capítulo 5. Sendo μ_1 e μ_2 as médias dos erros absolutos gerados, respectivamente, pelas contagens baseadas

Figura 50 – Ilustração de contagem de passageiros realizada com auxílio de catraca e utilizada como referência para o estudo de caso.



(a) Primeira parte.



(b) Segunda parte.

Fonte: o próprio autor.

no SmartSORT e no DeepSORT, a metodologia final de avaliação deste estudo consistiu na comparação entre μ_1 e μ_2, de maneira a apontar qual dos algoritmos de rastreamento foi responsável pela contagem com menor erro médio absoluto e, consequentemente, apresentou maior desempenho durante tal aplicação. Para aferir a extensão da diferença entre μ_1 e μ_2, realizou-se um teste de significância estatística com base nas seguintes hipóteses:

- $H_0 : \mu_1 = \mu_2$;

- $H_1 : \mu_1 \neq \mu_2$.

Neste estudo utilizou-se o teste estatístico de Wilcoxon (DERRAC et al., 2011), o qual corresponde a um teste não-paramétrico para amostras pareadas. Já o nível de significância adotado para o teste correspondeu a 5%, de maneira a garantir com 95% de confiança que a hipótese H_0 não foi erroneamente rejeitada.

Finalmente, vale ressaltar que as detecções apresentadas aos rastreadores SmartSORT e DeepSORT durante as execuções da ferramenta de contagem foram obtidas a partir do *framerwork* YOLO (REDMON; FARHADI, 2018a). Este é baseado numa arquitetura convolucional e pertence à categoria de detectores *single shot*. A Figura 51 ilustra as detecções obtidas através da aplicação do YOLO sobre algumas imagens do vídeo de entrada deste estudo.

Figura 51 – Ilustração de detecções (retângulos pretos) obtidas através da aplicação do *framework* YOLO (REDMON; FARHADI, 2018a) sobre o vídeo de entrada do estudo de caso.

(a) Detecções obtidas no instante 12 : 04 : 37.

(b) Detecções obtidas no instante 12 : 31 : 41.

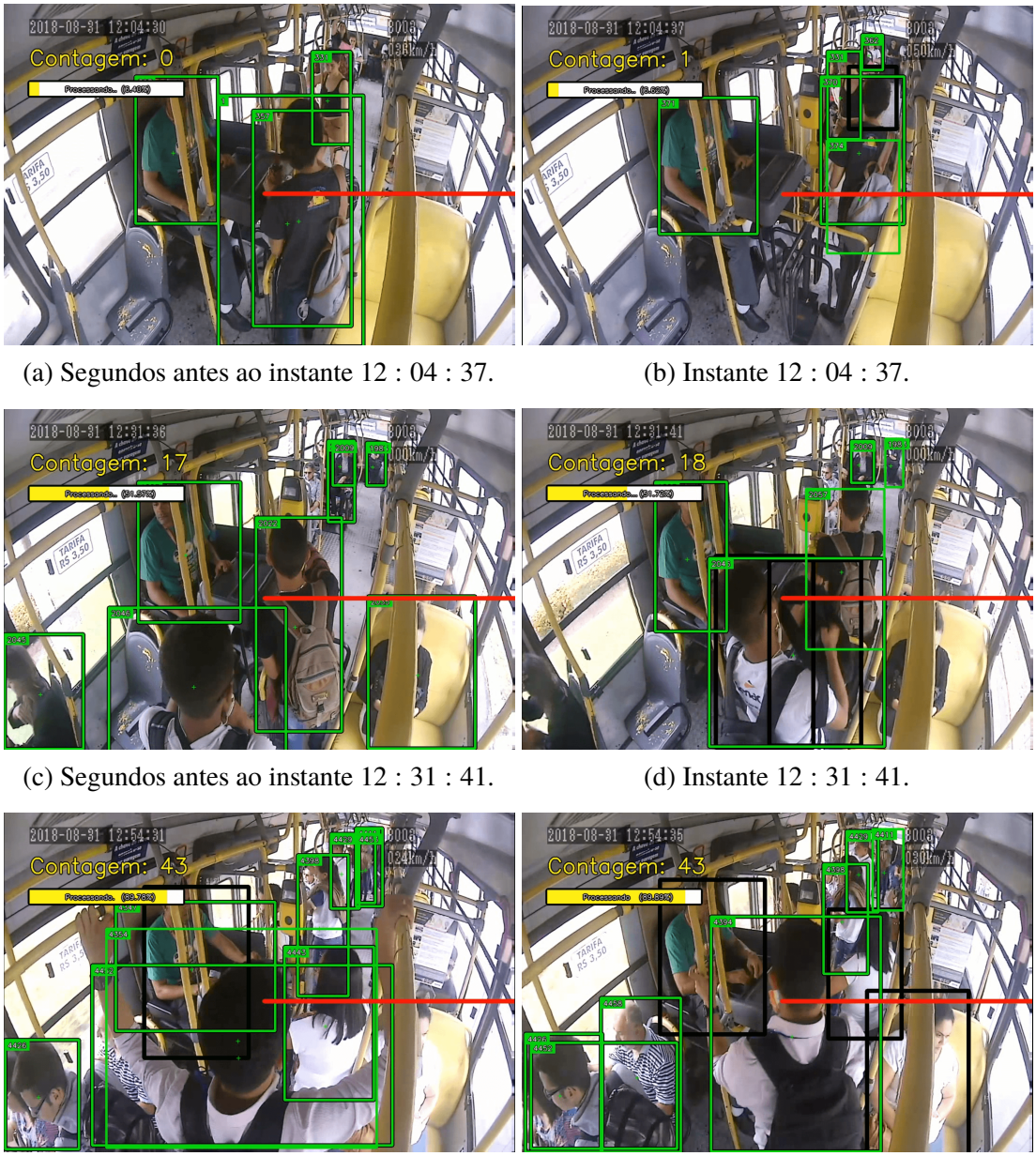
(c) Detecções obtidas no instante 12 : 54 : 35.

Fonte: o próprio autor.

6.3 Resultados

A Figura 52 ilustra resultados qualitativos obtidos durante a execução da ferramenta de contagem de passageiros baseada no rastreador SmartSORT. É possível notar a presença da fronteira de contagem (representada por um segmento de reta vermelho) posicionada sobre a catraca, além de um indicador da contagem acumulada pela ferramenta (em amarelo, sobre uma barra de progresso), das detecções (representadas por retângulos pretos) apresentadas como entrada ao rastreador e do estado s_i (representado por retângulos verdes) estimado pelo rastreador para cada objeto o_i com trajetória T_i.

Figura 52 – Ilustração de resultados qualitativos da contagem de passageiros baseada no rastreador SmartSORT em diferentes instantes.



(a) Segundos antes ao instante 12 : 04 : 37.

(b) Instante 12 : 04 : 37.

(c) Segundos antes ao instante 12 : 31 : 41.

(d) Instante 12 : 31 : 41.

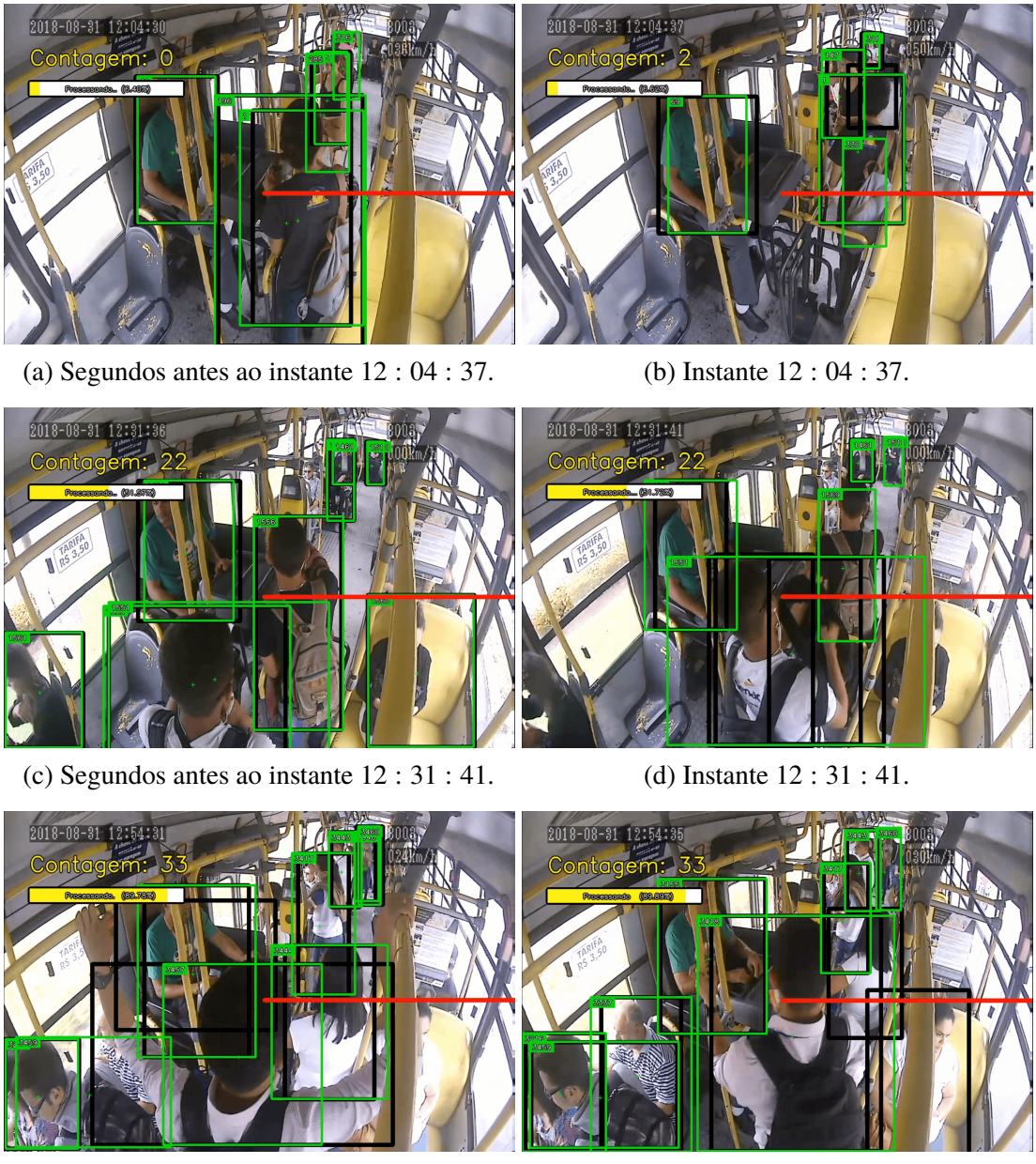
(e) Segundos antes ao instante 12 : 54 : 35.

(f) Instante 12 : 54 : 35.

Fonte: o próprio autor.

Já a Figura 53 ilustra resultados qualitativos gerados por meio da execução da ferramenta de contagem baseada no rastreador DeepSORT. Estes foram obtidos nos mesmos instantes daqueles ilustrados pela Figura 52.

Figura 53 – Ilustração de resultados qualitativos da contagem de passageiros baseada no rastreador DeepSORT, utilizado como *baseline*.



(a) Segundos antes ao instante 12 : 04 : 37.

(b) Instante 12 : 04 : 37.

(c) Segundos antes ao instante 12 : 31 : 41.

(d) Instante 12 : 31 : 41.

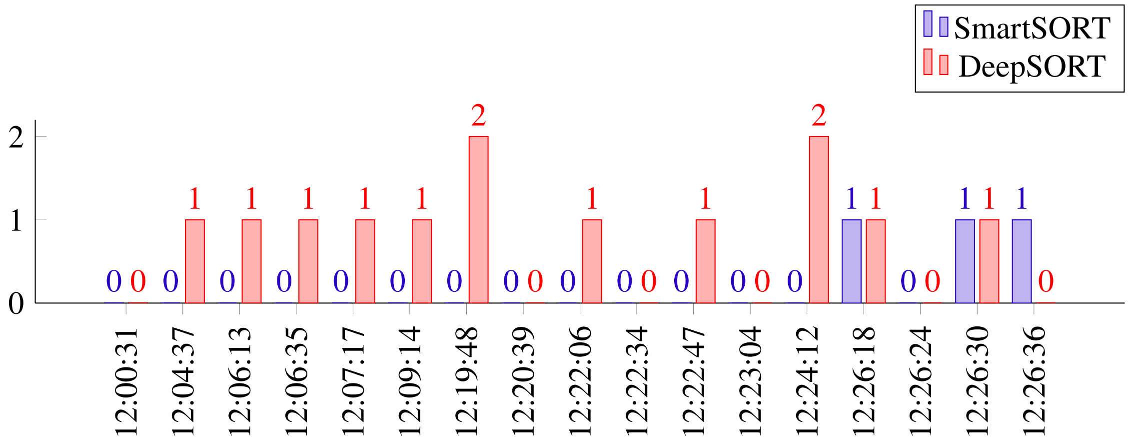
(e) Segundos antes ao instante 12 : 54 : 35.

(f) Instante 12 : 54 : 35.

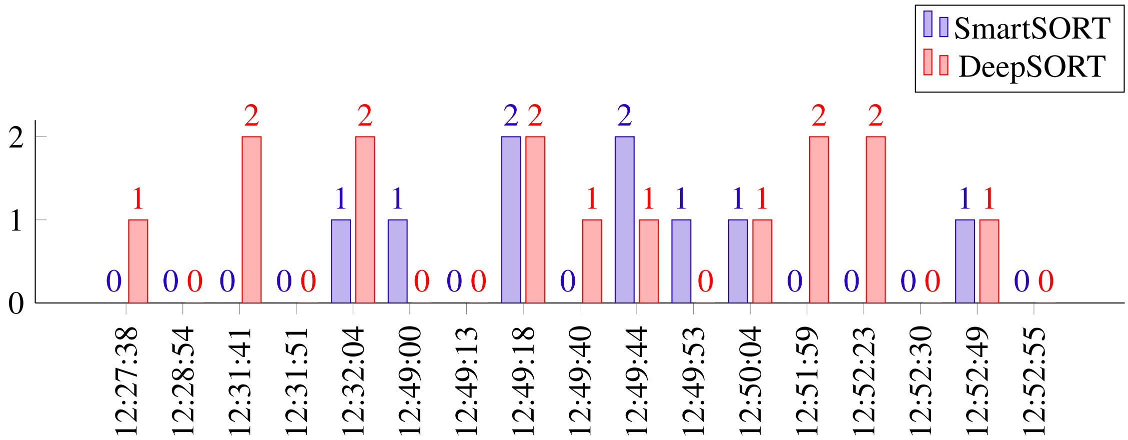
Fonte: o próprio autor.

Por sua vez, a Figura 54 compara o erro apresentado por cada versão da ferramenta de contagem em relação ao incremento da contagem real de passageiros ao longo do tempo. O erro apresentado para cada método foi obtido através do algoritmo 9.

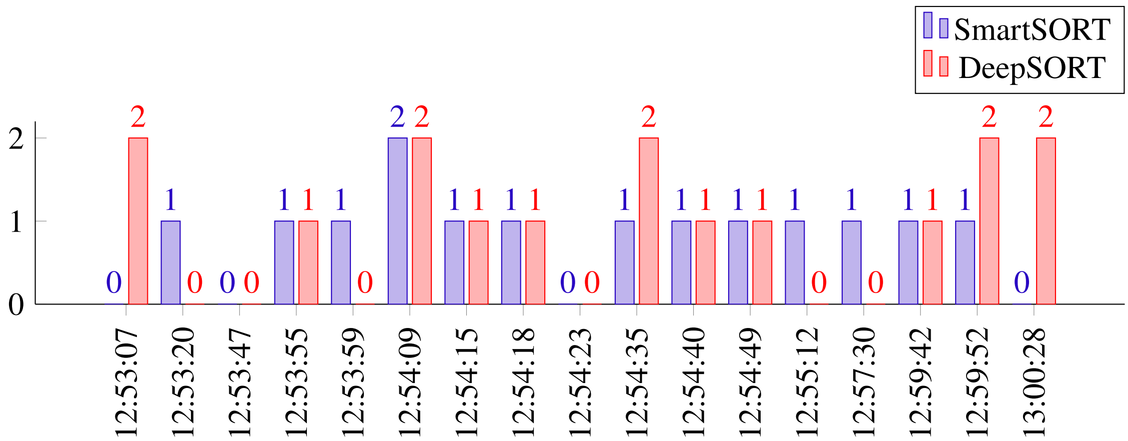
Figura 54 – Comparação entre o erro no incremento das contagens baseadas nos rastreadores SmartSORT e DeepSORT em relação ao incremento da contagem de referência.



(a) Primeira parte.



(b) Segunda parte.



(c) Terceira parte.

Fonte: o próprio autor.

Já a Tabela 12 sumariza, em termos de média e desvio padrão, os resultados relacionados ao erro absoluto total apresentado pelas contagens baseadas nos rastreadores DeepSORT e SmartSORT. Finalmente, a execução do teste não-paramétrico de Wilcoxon sobre os conjuntos

referentes aos erros de ambas as contagens gerou como resultado um p-valor equivalente a $0,01974$. Este teste foi realizado com o auxílio da biblioteca de testes estatísticas da linguagem R (R Core Team, 2018). Uma vez que a hipótese alternativa H_1 é bilateral e como o p-valor obtido é inferior a $0,025$, pode-se refutar a hipótese nula H_0 com uma probabilidade de acerto igual a 95%.

Tabela 12 – Resultados relacionados ao erro absoluto total apresentado pelas contagens baseadas nos rastreadores DeepSORT e SmartSORT.

	↓Erro médio absoluto	**↓Desvio padrão**
DeepSORT (*baseline*)	0,86	0,78
SmartSORT (proposto)	**0,51**	**0,61**

6.4 Discussão

A partir dos resultados apresentados pela Tabela 12, percebe-se que a qualidade da contagem baseada no rastreador SmartSORT foi superior a daquela realizada com o auxílio do método DeepSORT, haja vista que seu erro absoluto médio foi $40,7\%$ inferior ao desta última. Além disso, dada a rejeição da hipótese nula H_0 durante a aplicação do teste estatístico de Wilcoxon, pode-se atestar com 95% de confiança que os resultados apresentados por ambas as contagens são estatisticamente diferentes. Dessa forma, ao combinar o resultado deste teste aos apresentados pela Tabela 12, é possível afirmar que a performance da ferramenta de contagem automática de passageiros desenvolvida neste estudo e baseada no rastreador SmartSORT foi estatisticamente superior àquela aferida com base no método DeepSORT.

Uma das justificativas para o resultado alcançado ao final deste estudo está relacionada à movimentação brusca dos passageiros monitorados. Esta envolve a abertura de braços e o balançar do corpo devido à vibração do veículo, como observado na Figura 52 e na Figura 53. A partir destas figuras, percebe-se que a contagem baseada no algoritmo DeepSORT foi mais prejudicada pelo comportamento dos passageiros, haja vista que este rastreador assume de antemão, por meio da aplicação do Filtro de Kalman, que os objetos rastreados movimentam-se de modo linear e suave. Assim, observa-se na Figura 53 a ocorrência de falsos negativos durante a contagem (Figura 53c e Figura 53d, Figura 53e e Figura 53f). Estas ocorrências se devem à descontinuidade do rastreamento dos passageiros durante sua travessia pela catraca, uma vez que o DeepSORT considera impraticáveis associações entre trajetórias e detecções separadas por movimentos bruscos. Em contrapartida, a Figura 52 demonstra que a contagem baseada no SmartSORT foi mais tolerante à movimentação brusca dos passageiros, uma vez que o modelo de regressão utilizado por este método foi induzido com base em exemplos que descrevem o comportamento real de passageiros. Além disso, este regressor não realiza nenhuma suposição *a priori* quanto à dinâmica do movimento tratado.

Além da movimentação dos passageiros, outra justificativa para os resultados alcançados encontra-se na alta taxa de sobreposição das detecções apresentadas aos rastreadores, como pode-se observar na Figura 51. Esta taxa deve-se à sensibilidade do *framework* de detecção utilizado e ao nível de oclusão de pessoas inerente ao cenário do vídeo processado. Devido a estes fatores, somados às falhas oriundas do próprio detector, é possível notar na Figura 51 a ocorrência de múltiplas detecções que delimitam a mesma pessoa. Quando estas detecções são apresentadas ao extrator de características convolucionais utilizado por ambos os métodos de rastreamento, são obtidos descritores semelhantes. No entanto, em geral apenas uma destas detecções é realizada continuamente. Assim, caso os métodos de rastreamento deem mais peso às características visuais do que às de movimentação, múltiplas trajetórias referentes ao mesmo objeto são criadas e mantidas durante a passagem pela catraca, o que leva à realização de falsos incrementos na contagem. Este cenário foi observado neste estudo com maior frequência durante o uso do método DeepSORT (Figura 53a e Figura 53b), uma vez que os pesos atribuídos por tal rastreador às características visuais e de movimentação são definidos manualmente por seu usuário, ao contrário do SmartSORT, cujos pesos são definidos automaticamente durante a indução do seu regressor.

Por fim, vale destacar que o primeiro erro da contagem com o SmartSORT observado na Figura 54 ocorreu somente no instante $12:26:18$, após 13 acertos consecutivos. Ao analisar os instantes utilizados como referência durante este intervalo, percebe-se que em média a catraca foi rotacionada aproximadamente 1 vez a cada 2 minutos, sendo o intervalo mínimo entre duas rotações consecutivas igual a 13 segundos. Através desta informação, nota-se maior êxito da contagem realizada a partir do SmartSORT durante o trecho do vídeo com menor fluxo de passageiros (Figura 52a e Figura 52b). Já durante os trechos com maior fluxo, foram observados erros semelhantes aos discutidos anteriormente com base no DeepSORT (Figura 52a e Figura 52b). Estes erros podem ser justificados pela maior sobreposição dos passageiros, a qual ocasionou maior instabilidade durante a aquisição de detecções e menor capacidade de discriminação por parte das características visuais obtidas a partir do extrator convolucional. Ainda assim, observa-se na Figura 54 que mesmo durante os trechos com maior fluxo de passageiros a contagem com o SmartSORT cometeu menos erros que a baseada no DeepSORT.

Dessa forma, com base nos resultados alcançados ao longo deste estudo de caso foi possível demonstrar o uso do método SmartSORT numa aplicação relevante para o setor de transporte coletivo, com ganho significativo em performance sobre sua *baseline*. Além disso, a qualidade dos resultados obtidos a partir do método foi limitada pelas condições oferecidas durante este estudo, de maneira que em uma aplicação final há espaço para aprimoramentos através do treinamento de um detector, de um extrator de características visuais e até mesmo de um regressor de custo de associação específicos para as imagens processadas e para o cenário em questão.

7

Conclusão

Este trabalho explorou o rastreamento de múltiplos objetos em vídeo com base no paradigma *tracking-by-detection*. Por meio de uma revisão sistemática da literatura, foram apresentadas as principais técnicas e abordagens empregadas para a implementação daquele paradigma. A partir da discussão levantada, constatou-se a predominância de algoritmos de rastreamento cuja análise da similaridade entre objetos é feita através de funções de custo construídas manualmente. Dado que o ajuste de parâmetros destas funções dificulta sua adaptação para cenários distintos, este trabalho explorou o uso de técnicas de aprendizado de máquina para a indução automática dessas funções. Foi apresentado um modelo de regressão capaz de estimar o custo de associação entre a trajetória de um objeto já identificado e uma nova detecção com base em características de alto nível relacionadas à aparência e à movimentação. Com base neste modelo, foi proposto um método de rastreamento *online* denominado SmartSORT, o qual é capaz de lidar com variações temporais de aparência e de movimentação por meio de uma única janela deslizante, não sendo necessária a utilização de filtros.

A qualidade do método SmartSORT foi aferida por meio de três cenários de experimentação. No primeiro, o método foi avaliado durante o rastreamento de múltiplos pedestres através do *benchmark* MOT Challenge 2016. Durante a análise dos resultados, observou-se que a frequência de processamento do SmartSORT foi superior à apresentada pela maior parte dos rastreadores *online* listados neste trabalho como estado da arte, com ênfase para aqueles totalmente baseados em modelos de aprendizado profundo (sobre os quais o ganho mínimo foi de 59%). Além disso, a acurácia de $60,4\%$ obtida pelo SmartSORT é similar à apresentada por aqueles rastreadores. Estes resultados demonstram a qualidade da função de regressão induzida automaticamente a partir da metodologia desenvolvida neste trabalho, uma vez que através daquela função o método SmartSORT foi capaz de obter resultados competitivos em relação aos demais algoritmos de rastreamento submetidos ao mesmo *benchmark*.

Este trabalho também avaliou o método SmartSORT num segundo cenário de experi-

mentação, o qual envolveu o rastreamento de passageiros de ônibus através de uma base de dados construída localmente, denominada BUS Challenge 2018. Durante análise dos resultados alcançados, constatou-se que a acurácia de 99,2% apresentada pelo SmartSORT foi equivalente a de sua principal *baseline*, ao passo que a velocidade de execução do primeiro foi 42,8% superior a desta última. Estes resultados demonstram a capacidade de adaptação do método proposto a diferentes cenários, haja vista a manutenção do ganho em velocidade em relação à sua principal *baseline*, ao mesmo tempo em que a diferença entre suas acurácias foi minimizada. Soma-se a isso o fato de que a preparação do SmartSORT envolveu apenas a indução automática de seu modelo de regressão exclusivamente a partir da base BUS Challenge 2018.

Como terceiro cenário de experimentação, este trabalho apresentou um estudo de caso envolvendo a contagem automática de passageiros de ônibus através de algoritmos de rastreamento. Neste estudo foi possível verificar, em termos de erro médio absoluto, a superioridade da contagem realizada com base no SmartSORT em detrimento daquela conduzida por meio de sua principal *baseline*. Mais especificamente, o erro médio da primeira foi 40,7% menor que o da segunda. Este resultado, ratificado via teste estatístico com confiança de 95%, demonstra a capacidade de generalização da função de regressão utilizada pelo SmartSORT, a qual já havia sido induzida no experimento anterior a partir de imagens registradas com resolução, posicionamento de câmera e iluminação diferentes. Além disso, com base na performance de sua *baseline*, também foi possível verificar a dificuldade de adaptação de rastreadores baseados em funções de custo manuais para novos cenários.

Dessa forma, ao final deste trabalho foi obtido um método de rastreamento de múltiplos objetos *online*, adaptável a diferentes cenários e capaz de executar em tempo real. Além disso, foi construída uma nova base de dados formada por imagens de passageiros de ônibus já rotuladas, a qual permite a preparação e a avaliação de novos algoritmos de rastreamento. Por fim, uma ferramenta de contagem automática de pessoas também foi obtida. Tanto o método de rastreamento quanto a nova base e a ferramenta de contagem são disponibilizados através do endereço eletrônico <https://git.dcomp.ufs.br/michel.meneses/smartsort>.

7.1 Trabalhos Futuros

Algumas linhas de melhoramento deste trabalho foram pensadas mas não realizadas em virtude do tempo. Percebeu-se a possibilidade de experimentar o uso de arquiteturas rasas de aprendizado de máquina especialmente projetadas para a modelagem de dados sequenciais (*e.g.*, *vanilla* RNN) ao invés de uma MLP combinada a uma janela deslizante, porém mantendo-se a apresentação de características de alto-nível ao modelo de regressão. Acredita-se que através desta alteração seja possível tornar o método SmartSORT menos vulnerável a associações incorretas, aumentar sua capacidade de predição do posicionamento futuro de objetos com base em sua movimentação e operar em tempo real, haja vista a manutenção da arquitetura simplificada do

seu modelo de regressão. Por fim, também observou-se a possibilidade de realizar experimentos com diferentes algoritmos de treinamento, além do *Backpropagation*, já que é possível aprimorar a eficiência do treinamento do modelo de regressão e, consequentemente, aumentar a acurácia de rastreamento do SmartSORT.

Referências

ABDELALI, H.; ESSANNOUNI, F.; ABOUTAJDINE, D. Object tracking in video via particle filter. *International Journal of Intelligent Engineering Informatics*, v. 4, p. 340, 01 2016. Citado na página 59.

ACOREL. *Automatic People Counting systems - big data and the future of public transport*. Saint-Péray: [s.n.], 2017. Disponível em: <https://intelligenttransport.com/wp-content/uploads/Whitepaper-Automatic-people-counting-systems.pdf>. Citado na página 99.

AGARWAL, S.; TERRAIL, J. O. D.; JURIE, F. *Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks*. 2018. Citado 2 vezes nas páginas 46 e 54.

AL-SHAKARJI, N. M. et al. Robust multi-object tracking with semantic color correlation. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017. p. 1–7. ISBN 978-1-5386-2939-0. Disponível em: <http://ieeexplore.ieee.org/document/8078507/>. Citado na página 36.

ALAHI, A. et al. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2016. p. 961–971. ISBN 978-1-4673-8851-1. Citado na página 22.

ALTCHé, F.; FORTELLE, A. de L. An lstm network for highway trajectory prediction. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2017. p. 353–359. ISSN 2153-0017. Citado na página 59.

ANDRILUKA, M. et al. PoseTrack: A Benchmark for Human Pose Estimation and Tracking. 2018. Disponível em: <https://posetrack.net/workshops/iccv2017/>. Citado na página 20.

ANDRILUKA, M.; ROTH, S.; SCHIELE, B. People-tracking-by-detection and people-detection-by-tracking. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008. p. 1–8. ISBN 978-1-4244-2242-5. Disponível em: <http://ieeexplore.ieee.org/document/4587583/>. Citado na página 36.

ARULAMPALAM, M. et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, v. 50, n. 2, p. 174–188, 2002. ISSN 1053587X. Disponível em: <http://ieeexplore.ieee.org/document/978374/>. Citado na página 40.

BA, S. et al. An On-line Variational Bayesian Model for Multi-Person Tracking from Cluttered Scenes. sep 2015. Disponível em: <http://arxiv.org/abs/1509.01520http://dx.doi.org/10.1016/j.cviu.2016.07.006>. Citado na página 35.

BAYSAL, S.; DUYGULU, P. Sentioscope: A Soccer Player Tracking System Using Model Field Particles. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 26, n. 7, p. 1350–1362, jul 2016. ISSN 1051-8215. Disponível em: <http://ieeexplore.ieee.org/document/7156105/>. Citado na página 20.

BENFOLD, B.; REID, I. Stable multi-target tracking in real-time surveillance video. In: *CVPR 2011*. IEEE, 2011. p. 3457–3464. ISBN 978-1-4577-0394-2. Disponível em: <http://ieeexplore.ieee.org/document/5995667/>. Citado na página 36.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, v. 2008, p. 1–10, 2008. ISSN 1687-5176. Citado 4 vezes nas páginas 37, 83, 86 e 95.

BERNINI, N. et al. An embedded system for counting passengers in public transportation vehicles. In: *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. IEEE, 2014. p. 1–6. ISBN 978-1-4799-2280-2. Disponível em: <http://ieeexplore.ieee.org/document/6935562/>. Citado na página 100.

BEWLEY, A. et al. Simple online and realtime tracking. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. p. 3464–3468. ISBN 978-1-4673-9961-6. Disponível em: <http://ieeexplore.ieee.org/document/7533003/>. Citado 5 vezes nas páginas 22, 39, 40, 60 e 88.

BIRESAW, T. A. et al. ViTBAT: Video tracking and behavior annotation tool. In: *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016. p. 295–301. ISBN 978-1-5090-3811-4. Disponível em: <http://ieeexplore.ieee.org/document/7738055/>. Citado 3 vezes nas páginas 12, 129 e 130.

BOCHINSKI, E.; EISELEIN, V.; SIKORA, T. High-Speed tracking-by-detection without using image information. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017. p. 1–6. ISBN 978-1-5386-2939-0. Disponível em: <http://ieeexplore.ieee.org/document/8078516/>. Citado 6 vezes nas páginas 22, 36, 39, 41, 60 e 88.

BORAGULE, A.; JEON, M. Joint cost minimization for multi-object tracking. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017. p. 1–6. ISBN 978-1-5386-2939-0. Disponível em: <http://ieeexplore.ieee.org/document/8078481/>. Citado na página 39.

BORAGULE, A.; JEON, M. Joint cost minimization for multi-object tracking. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017. p. 1–6. ISBN 978-1-5386-2939-0. Disponível em: <http://ieeexplore.ieee.org/document/8078481/>. Citado 2 vezes nas páginas 41 e 61.

BOUCHARD, M.; JOUSSELME, A.-L.; DORé, P.-E. A proof for the positive definiteness of the jaccard index matrix. *International Journal of Approximate Reasoning*, v. 54, n. 5, p. 615 – 626, 2013. ISSN 0888-613X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0888613X1300008X>. Citado na página 60.

BULLINGER, S.; BODENSTEINER, C.; ARENS, M. Instance Flow Based Online Multiple Object Tracking. mar 2017. Disponível em: <http://arxiv.org/abs/1703.01289>. Citado na página 36.

CHEN, L. et al. Online multi-object tracking with convolutional neural networks. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017. p. 645–649. ISBN 978-1-5090-2175-8. Disponível em: <http://ieeexplore.ieee.org/document/8296360/>. Citado 2 vezes nas páginas 39 e 40.

CHEN, W. et al. Monocular semantic SLAM in dynamic street scene based on multiple object tracking. In: *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2017. p. 599–604.

ISBN 978-1-5386-3135-5. Disponível em: <http://ieeexplore.ieee.org/document/8274845/>. Citado na página 20.

CHENG, Y. et al. Human motion prediction using adaptable neural networks. 10 2018. Citado na página 59.

CHONG, E. et al. Visual 3D tracking of child-adult social interactions. In: *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2017. p. 399–406. ISBN 978-1-5386-3715-9. Disponível em: <http://ieeexplore.ieee.org/document/8329835/>. Citado na página 20.

CNT; NTU. *Pesquisa Mobilidade da População Urbana 2017*. [S.l.], 2017. 96p p. Disponível em: <https://www.ntu.org.br/novo/upload/Publicacao/Pub636397002002520031.pdf>. Citado na página 98.

CUN, Y. L. A theoretical framework for back-propagation. In: TOURETZKY, D.; HINTON, G.; SEJNOWSKI, T. (Ed.). *Proceedings of the 1988 Connectionist Models Summer School, CMU, Pittsburg, PA*. [S.l.]: Morgan Kaufmann, 1988. p. 21–28. Citado 2 vezes nas páginas 67 e 71.

CZYZEWSKI, A.; DALKA, P. Examining kalman filters applied to tracking objects in motion. In: *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. [S.l.: s.n.], 2008. p. 175–178. ISSN 2158-5873. Citado na página 57.

Da Silva, W. H. N. *SISTEMA DE BILHETAGEM ELETRÔNICA: TENDÊNCIAS NO MODAL D E TRANSPORTE COLETIVO*. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2017. Disponível em: <https://monografias.ufrn.br/jspui/bitstream/123456789/5108/1/WanderleyHNS{_}Monografia.> Citado na página 99.

DAI, J. et al. *R-FCN: Object Detection via Region-based Fully Convolutional Networks*. 2016. Citado na página 51.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, v. 2, 06 2005. Citado na página 46.

DENG, J. et al. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009. p. 248–255. ISBN 978-1-4244-3992-8. Disponível em: <http://ieeexplore.ieee.org/document/5206848/>. Citado 2 vezes nas páginas 55 e 133.

DEQIN, X. et al. A multi-target trapping and tracking algorithm for Bactrocera Dorsalis based on cost model. *Computers and Electronics in Agriculture*, v. 123, p. 224–231, apr 2016. ISSN 01681699. Citado na página 20.

DERRAC, J. et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 3 – 18, 2011. ISSN 2210-6502. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2210650211000034>. Citado na página 105.

ELKOSANTINI, S.; DARMOUL, S. Intelligent Public Transportation Systems: A review of architectures and enabling technologies. In: *2013 International Conference on Advanced Logistics and Transport*. IEEE, 2013. p. 233–238. ISBN 978-1-4799-0313-9. Disponível em: <http://ieeexplore.ieee.org/document/6568465/>. Citado na página 99.

ESCOLANO, C. O. et al. Passenger demand forecast using optical flow passenger counting system for bus dispatch scheduling. In: *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016. p. 1875–1878. ISBN 978-1-5090-2597-8. Disponível em: <http://ieeexplore.ieee.org/document/7848347/>. Citado na página 100.

EVERINGHAM, M. et al. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, Springer Science and Business Media LLC, v. 88, n. 2, p. 303–338, set. 2009. Disponível em: <https://doi.org/10.1007/s11263-009-0275-4>. Citado na página 131.

FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. Rio de Janeiro: Grupo Gen - LTC, 2011. ISBN 9788521618805. Citado na página 84.

FAN, L. et al. A survey on multiple object tracking algorithm. In: *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2016. p. 1855–1862. ISBN 978-1-5090-4102-2. Disponível em: <http://ieeexplore.ieee.org/document/7832121/>. Citado 5 vezes nas páginas 22, 56, 57, 62 e 63.

FANG, K.; XIANG, Y.; SAVARESE, S. Recurrent autoregressive networks for online multi-object tracking. *CoRR*, abs/1711.02741, 2017. Disponível em: <http://arxiv.org/abs/1711.02741>. Citado na página 88.

FARAGHER, R. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal Processing Magazine*, v. 29, n. 5, p. 128–132, Sep. 2012. ISSN 1053-5888. Citado 2 vezes nas páginas 9 e 58.

FEIJÓ, G. d. O. et al. An algorithm to track laboratory zebrafish shoals. *Computers in Biology and Medicine*, v. 96, p. 79–90, may 2018. ISSN 00104825. Citado na página 20.

FELZENSZWALB, P.; HUTTENLOCHER, D. Efficient graph-based image segmentation. *International Journal of Computer Vision*, v. 59, p. 167–181, 09 2004. Citado na página 47.

FENG, P. et al. Social Force Model-Based MCMC-OCSVM Particle PHD Filter for Multiple Human Tracking. *IEEE Transactions on Multimedia*, v. 19, n. 4, p. 725–739, apr 2017. ISSN 1520-9210. Disponível em: <http://ieeexplore.ieee.org/document/7779037/>. Citado na página 35.

FGV, F. G. V. Mobilidade Urbana e Cidadania - Percepções dos usuários de transporte público no Brasil. v. 3, 2014. Disponível em: <www.dapp.fgv.br>. Citado na página 98.

FNP, F. N. d. P.; ANTP, A. N. d. T. P. *Custos dos serviços de transporte público por ônibus: método de cálculo*. [S.l.], 2017. Disponível em: <http://files.antp.org.br/2017/8/21/1.-metodo-de-calculo--final-impresso.pdf>. Citado na página 99.

FU, Z. et al. Particle PHD filter based multi-target tracking using discriminative group-structured dictionary learning. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017. p. 4376–4380. ISBN 978-1-5090-4117-6. Disponível em: <http://ieeexplore.ieee.org/document/7952983/>. Citado na página 36.

G1. *Sistema flagra fraudes em cartões de gratuidades no transporte público de Juiz de Fora | Zona da Mata | G1*. 2017. Disponível em: <https://g1.globo.com/mg/zona-da-mata/noticia/sistema-flagra-fraudes-em-cartoes-de-gratuidades-em-onibus-em-juiz-de-fora.ghtml>. Citado na página 99.

GEIGER, A. et al. 3D Traffic Scene Understanding From Movable Platforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 5, p. 1012–1025, may 2014. ISSN 0162-8828. Disponível em: <http://ieeexplore.ieee.org/document/6613480/>. Citado na página 22.

GIRON-SIERRA, J. M. Kalman filter, particle filter and other bayesian filters. In: *Signals and Communication Technology*. Springer Singapore, 2016. p. 3–148. Disponível em: <https://doi.org/10.1007/978-981-10-2540-2_1>. Citado na página 58.

GIRSHICK, R. *Fast R-CNN*. 2015. Citado 5 vezes nas páginas 8, 48, 49, 51 e 72.

GIRSHICK, R. et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2013. Citado 3 vezes nas páginas 8, 47 e 48.

GONZALEZ, R. *Digital image processing*. Upper Saddle River, N.J: Prentice Hall, 2008. ISBN 013168728X. Citado na página 54.

HAMUDA, E. et al. Improved image processing-based crop detection using Kalman filtering and the Hungarian algorithm. *Computers and Electronics in Agriculture*, v. 148, p. 37–44, may 2018. ISSN 01681699. Citado 2 vezes nas páginas 20 e 21.

HE, K. et al. *Mask R-CNN*. 2017. Citado na página 51.

Heng-Xin Chen; Bin Fang; Yuan-Yan Tang. A method for multiple morph targets tracking based on region growing. In: *2007 International Conference on Wavelet Analysis and Pattern Recognition*. IEEE, 2007. p. 194–197. ISBN 978-1-4244-1065-1. Disponível em: <http://ieeexplore.ieee.org/document/4420662/>. Citado na página 99.

HENRIQUES, J. F. et al. High-Speed Tracking with Kernelized Correlation Filters. apr 2014. Disponível em: <http://arxiv.org/abs/1404.7584http://dx.doi.org/10.1109/TPAMI.2014.2345390>. Citado na página 41.

HILKE, K. et al. Online multi-person tracking using Integral Channel Features. *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2016. Citado na página 36.

HORN, Z. et al. Performance of convolutional neural networks for feature extraction in froth flotation sensing. *IFAC-PapersOnLine*, Elsevier BV, v. 50, n. 2, p. 13–18, dec 2017. Disponível em: <https://doi.org/10.1016/j.ifacol.2017.12.003>. Citado na página 55.

HUANG, J. et al. Speed/accuracy trade-offs for modern convolutional object detectors. 11 2016. Citado na página 133.

HUBER, P. J. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 35, n. 1, p. 73–101, mar. 1964. Disponível em: <https://doi.org/10.1214/aoms/1177703732>. Citado na página 71.

IBRAHIM, M. N. et al. Segmenting and Labeling blood vessels in choroidal Haller's layer: A multiple target tracking approach. In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2017. p. 113–116. ISBN 978-1-5090-4179-4. Disponível em: <http://ieeexplore.ieee.org/document/7897218/>. Citado na página 20.

INSTANTCOUNTING. *Real Use*. 1998. Disponível em: <http://www.instantcounting.com/use.html>. Citado na página 99.

ITER, D.; KUCK, J.; ZHUANG, P. Target Tracking with Kalman Filtering, KNN and LSTMs. 2016. Disponível em: <http://cs229.stanford.edu/proj2016/report/IterKuckZhuang-TargetTrackingwithKalmanFilteringKNNandLSTMs-report.pdf>. Citado na página 59.

JOSEPH, E.; GALEANO, P.; LILLO, R. E. The Mahalanobis distance for functional data with applications to classification. apr 2013. Disponível em: <http://arxiv.org/abs/1304.4786>. Citado na página 41.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, v. 82, p. 35–45, 1960. Citado 2 vezes nas páginas 40 e 58.

KITCHENHAM, B. Systematic review in software engineering: Where we are and where we should be going. In: *Proceedings of the 2Nd International Workshop on Evidential Assessment of Software Technologies*. New York, NY, USA: ACM, 2012. (EAST '12), p. 1–2. ISBN 978-1-4503-1509-8. Disponível em: <http://doi.acm.org/10.1145/2372233.2372235>. Citado na página 25.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, v. 60, p. 84–90, 2012. Citado 3 vezes nas páginas 9, 46 e 56.

KROLL. *Global Fraud & Risk Report Forging New Paths in Times of Uncertainty*. [S.l.], 2018. Disponível em: <file:///tmp/mozilla{_}michel0/Kroll{_}Global{_}Fraud{_}Risk{_}Repor>. Citado na página 99.

KUHN, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics*, v. 52, n. 1, p. 7–21, feb 2005. ISSN 0894-069X. Disponível em: <http://doi.wiley.com/10.1002/nav.20053>. Citado 5 vezes nas páginas 10, 40, 62, 67 e 68.

KüNSCH, H. R. Particle filters. *Bernoulli*, Bernoulli Society for Mathematical Statistics and Probability, v. 19, n. 4, p. 1391–1403, 09 2013. Disponível em: <https://doi.org/10.3150/12-BEJSP07>. Citado na página 59.

LEAL-TAIXÉ, L.; FERRER, C. C.; SCHINDLER, K. Learning by tracking: Siamese CNN for robust target association. apr 2016. Disponível em: <http://arxiv.org/abs/1604.07866>. Citado na página 22.

LEAL-TAIXÉ, L. et al. Tracking the Trackers: An Analysis of the State of the Art in Multiple Object Tracking. 2017. Disponível em: <https://arxiv.org/pdf/1704.02781.pdf>. Citado 3 vezes nas páginas 21, 43 e 53.

LEAL-TAIXÉ, L. et al. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. 2015. Disponível em: <http://motchallenge.net/vis/>. Citado na página 36.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, 2015. Disponível em: <http://dx.doi.org/10.1038/nature14539>. Citado na página 100.

LEI, Z. et al. Tracking moving weak objects in celestial image sequences. *IEEE Transactions on Aerospace and Electronic Systems*, v. 52, n. 3, p. 1257–1266, jun 2016. ISSN 0018-9251. Disponível em: <http://ieeexplore.ieee.org/document/7511856/>. Citado 2 vezes nas páginas 20 e 21.

LI, H. et al. *An Analysis of Pre-Training on Object Detection*. 2019. Citado na página 133.

LI, P. et al. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, v. 76, p. 323 – 338, 2018. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320317304612>. Citado na página 55.

LI, Q. et al. Kalman filter and its application. In: . [s.n.], 2016. p. 74–77. Cited By 10. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84991824530&doi=10.1109%2fICINIS.2015.35&partnerID=40&md5=10469122a2d3b49c646cdd215e9e6ea6>. Citado na página 58.

LI, X. et al. *A Survey of Appearance Models in Visual Object Tracking*. 2013. Citado na página 54.

LI, X. et al. A multiple object tracking method using kalman filter. In: *The 2010 IEEE International Conference on Information and Automation*. [S.l.: s.n.], 2010. p. 1862–1866. Citado na página 57.

Li Zhang; Yuan Li; NEVATIA, R. Global data association for multi-object tracking using network flows. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008. p. 1–8. ISBN 978-1-4244-2242-5. Disponível em: <http://ieeexplore.ieee.org/document/4587584/>. Citado na página 22.

LIN, H. et al. Online Weighted Clustering for Real-time Abnormal Event Detection in Video Surveillance. In: *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*. New York, New York, USA: ACM Press, 2016. p. 536–540. ISBN 9781450336031. Citado na página 20.

LIN, T. Y. et al. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 8693 LNCS, n. PART 5, p. 740–755, 2014. ISSN 16113349. Citado na página 133.

LIN, Z. et al. Online multi-object tracking based on hierarchical association and sparse representation. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017. p. 655–659. ISBN 978-1-5090-2175-8. Disponível em: <http://ieeexplore.ieee.org/document/8296362/>. Citado 3 vezes nas páginas 39, 40 e 60.

LIU, L. et al. *Deep Learning for Generic Object Detection: A Survey*. 2018. Citado 2 vezes nas páginas 45 e 52.

LIU, W. et al. Ssd: Single shot multibox detector. 2015. Citado na página 51.

LUO, W. et al. Multiple Object Tracking: A Literature Review. sep 2014. Disponível em: <http://arxiv.org/abs/1409.7618>. Citado na página 21.

MAHMOUDI, N.; AHADI, S. M.; RAHMATI, M. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, Aug 2018. ISSN 1573-7721. Disponível em: <https://doi.org/10.1007/s11042-018-6467-6>. Citado na página 88.

MARTINEZ, J.; BLACK, M. J.; ROMERO, J. *On human motion prediction using recurrent neural networks*. 2017. Citado na página 59.

MENG, S.; SHEN, H.-B. A robust cell tracking framework by fusing global and local optimization algorithms. In: *2016 Chinese Control and Decision Conference (CCDC)*. IEEE, 2016. p. 2079–2084. ISBN 978-1-4673-9714-8. Disponível em: <http://ieeexplore.ieee.org/document/7531327/>. Citado 2 vezes nas páginas 20 e 21.

MILAN, A. et al. MOT16: A Benchmark for Multi-Object Tracking. mar 2016. Disponível em: <http://arxiv.org/abs/1603.00831>. Citado 8 vezes nas páginas 10, 13, 36, 74, 75, 76, 100 e 128.

MOHAN, K. R. S. A survey on image feature descriptors. In: . [S.l.: s.n.], 2014. Citado na página 54.

MRITHU, A. S.; FRANCIS, A. B. An efficient implementation of video based traffic analysis system. In: *2016 International Conference on Emerging Technological Trends (ICETT)*. IEEE, 2016. p. 1–6. ISBN 978-1-5090-3751-3. Disponível em: <http://ieeexplore.ieee.org/document/7873681/>. Citado 2 vezes nas páginas 20 e 21.

MUKHERJEE, S. et al. Anovel framework for automatic passenger counting. In: *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011. p. 2969–2972. ISBN 978-1-4577-1303-3. Disponível em: <http://ieeexplore.ieee.org/document/6116284/>. Citado na página 100.

NGUYEN, H. V.; BAI, L. Cosine similarity metric learning for face verification. In: *Computer Vision – ACCV 2010*. Springer Berlin Heidelberg, 2011. p. 709–720. Disponível em: <https://doi.org/10.1007/978-3-642-19309-5_55>. Citado na página 61.

NTU, A. N. d. E. d. T. U. *Pesquisa revela crise no transporte público urbano*. 2017. Disponível em: <https://www.ntu.org.br/novo/NoticiaCompleta.aspx?idNoticia=822{&}idArea=10{&}idSegundoNiv>. Citado na página 98.

NTU, A. N. d. E. d. T. U. *Dados do Transporte Público por Ônibus*. 2018. Disponível em: <https://www.ntu.org.br/novo/AreasInternas.aspx?idArea=7{&}idSegundoNivel=>. Citado 2 vezes nas páginas 98 e 99.

ORON, S.; BAR-HILLE, A.; AVIDAN, S. Extended Lucas-Kanade Tracking. In: . [S.l.: s.n.], 2014. p. 142–156. Citado na página 22.

ORON, S.; BAR-HILLEL, A.; AVIDAN, S. Real-time tracking-with-detection for coping with viewpoint change. *Machine Vision and Applications*, v. 26, n. 4, p. 507–518, may 2015. ISSN 0932-8092. Citado na página 22.

PARK, S.-H.; LEE, K.; YOON, K.-J. Robust online multiple object tracking based on the confidence-based relative motion network and correlation filter. In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. p. 3484–3488. ISBN 978-1-4673-9961-6. Disponível em: <http://ieeexplore.ieee.org/document/7533007/>. Citado 2 vezes nas páginas 39 e 40.

PRECHELT, L. Early stopping — but when? In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012. p. 53–67. Disponível em: <https://doi.org/10.1007/978-3-642-35289-8_5>. Citado na página 84.

R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018. Disponível em: <https://www.R-project.org/>. Citado na página 110.

RABINER, L.; JUANG, B. An introduction to hidden Markov models. *IEEE ASSP Magazine*, v. 3, n. 1, p. 4–16, 1986. ISSN 0740-7467. Disponível em: <http://ieeexplore.ieee.org/document/1165342/>. Citado na página 40.

REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. *arXiv*, 2018. Citado 4 vezes nas páginas 12, 51, 105 e 106.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. apr 2018. Disponível em: <http://arxiv.org/abs/1804.02767>. Citado na página 133.

REN, S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. jun 2015. Disponível em: <http://arxiv.org/abs/1506.01497>. Citado 5 vezes nas páginas 8, 21, 49, 50 e 51.

REN, S. et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 6, p. 1137–1149, jun 2017. ISSN 0162-8828. Disponível em: <http://ieeexplore.ieee.org/document/7485869/>. Citado na página 133.

Retail Sensing. *Count footfall in shopping centres and malls*. 2018. Disponível em: <https://www.retailsensing.com/footfall-counters.html>. Citado na página 99.

RISTANI, E.; TOMASI, C. Tracking Multiple People Online and in Real Time. In: . [S.l.: s.n.], 2015. p. 444–459. Citado na página 22.

RODRIGUES, E. et al. Multi-objective Tracking Applied to Bat Populations. In: *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*. IEEE, 2016. p. 155–159. ISBN 978-1-5090-4149-7. Disponível em: <http://ieeexplore.ieee.org/document/7517269/>. Citado na página 20.

ROWLEY, H.; BALUJA, S.; KANADE, T. Neural network-based face detection. In: . [S.l.: s.n.], 1996. v. 20, p. 203–208. Citado na página 46.

RUDER, S. *An overview of gradient descent optimization algorithms*. 2016. Citado 2 vezes nas páginas 84 e 94.

SADEGHIAN, A.; ALAHI, A.; SAVARESE, S. Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017. p. 300–311. ISBN 978-1-5386-1032-9. Disponível em: <http://ieeexplore.ieee.org/document/8237303/>. Citado 2 vezes nas páginas 23 e 70.

SANCHEZ-MATILLA, R.; POIESI, F.; CAVALLARO, A. Online multi-target tracking with strong and weak detections. In: *ECCV Workshops*. [S.l.: s.n.], 2016. Citado na página 88.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. 2018. Citado 3 vezes nas páginas 12, 133 e 134.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. FaceNet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. p. 815–823. ISBN 978-1-4673-6964-0. Disponível em: <http://ieeexplore.ieee.org/document/7298682/>. Citado na página 61.

SHEN, J. et al. Fast Online Tracking With Detection Refinement. *IEEE Transactions on Intelligent Transportation Systems*, v. 19, n. 1, p. 162–173, jan 2018. ISSN 1524-9050. Disponível em: <http://ieeexplore.ieee.org/document/8103346/>. Citado 2 vezes nas páginas 39 e 41.

SHERSTINSKY, A. *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network*. 2018. Citado na página 89.

SHI, R.; NGAN, K. N.; LI, S. Jaccard index compensation for object segmentation evaluation. In: *2014 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2014. p. 4457–4461. ISSN 1522-4880. Citado na página 60.

Sihua Ye; Jiancong Wang. Applications of automated monitoring system in road passenger transportation. In: *6th Advanced Forum on Transportation of China (AFTC 2010)*. IET, 2010. p. 229–239. ISBN 978-1-84919-316-0. Disponível em: <http://digital-library.theiet.org/content/conferences/10.1049/cp.2010.1134>. Citado na página 99.

SINGH, G.; RAJAN, S.; S., M. A greedy data association technique for multiple object tracking. In: *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*. [S.l.: s.n.], 2017. p. 177–184. Citado na página 43.

SON, J. et al. Multi-object Tracking with Quadruplet Convolutional Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. p. 3786–3795. ISBN 978-1-5386-0457-1. Disponível em: <http://ieeexplore.ieee.org/document/8099886/>. Citado 2 vezes nas páginas 23 e 70.

TAIGMAN, Y. et al. Deepface: Closing the gap to human-level performance in face verification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1701–1708. ISSN 1063-6919. Citado 2 vezes nas páginas 55 e 61.

TANG, S. et al. Multi-Person Tracking by Multicut and Deep Matching. aug 2016. Disponível em: <http://arxiv.org/abs/1608.05404>. Citado na página 22.

TANG, Y. et al. *Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamic*. 2018. Citado na página 59.

UCHIDA, Y. *Local Feature Detectors, Descriptors, and Image Representations: A Survey*. 2016. Citado na página 54.

UIJLINGS, J. et al. Selective search for object recognition. *International Journal of Computer Vision*, 2013. Disponível em: <http://www.huppelen.nl/publications/selectiveSearchDraft.pdf>. Citado 3 vezes nas páginas 8, 47 e 48.

ULLAH, M.; CHEIKH, F. A.; IMRAN, A. S. HoG based real-time multi-target tracking in Bayesian framework. In: *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016. p. 416–422. ISBN 978-1-5090-3811-4. Disponível em: <http://ieeexplore.ieee.org/document/7738080/>. Citado na página 35.

V-COUNT. *Case Studies*. 2017. Disponível em: <https://v-count.com/case-studies/>. Citado na página 99.

VIOLA, P.; JONES, M. Rapid object dete ction using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 2001. Citado na página 46.

WANG, L.; ZHANG, L.; YI, Z. Trajectory predictor by using recurrent neural networks in visual tracking. *IEEE Transactions on Cybernetics*, v. 47, n. 10, p. 3172–3183, Oct 2017. ISSN 2168-2267. Citado na página 59.

WANG, X. et al. Active colloids segmentation and tracking. *Pattern Recognition*, v. 60, p. 177–188, dec 2016. ISSN 00313203. Citado na página 20.

WOJKE, N.; BEWLEY, A.; PAULUS, D. Simple online and realtime tracking with a deep association metric. In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017. p. 3645–3649. ISBN 978-1-5090-2175-8. Disponível em: <http://ieeexplore.ieee.org/document/8296962/>. Citado 13 vezes nas páginas 22, 39, 41, 62, 66, 81, 83, 88, 89, 90, 91, 95 e 103.

XIANG-YANG, S.; HAO-WEI, W. Study on Method of Multi-feature Reduction Based on Rough Set in Passenger Counting. In: *2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*. IEEE, 2016. p. 307–310. ISBN 978-1-5090-3575-5. Disponível em: <http://ieeexplore.ieee.org/document/7823547/>. Citado na página 99.

YANG, M.; JIA, Y. Temporal Dynamic Appearance Modeling for Online Multi-Person Tracking. oct 2015. Disponível em: <http://arxiv.org/abs/1510.02906http://dx.doi.org/10.1016/j.cviu.2016.05.003>. Citado 2 vezes nas páginas 39 e 40.

YANG, M.; WU, Y.; JIA, Y. A Hybrid Data Association Framework for Robust Online Multi-Object Tracking. mar 2017. Disponível em: <http://arxiv.org/abs/1703.10764http://dx.doi.org/10.1109/TIP.2017.2745103>. Citado na página 35.

YOON, J. H. et al. Online Multi-object Tracking via Structural Constraint Event Aggregation. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. p. 1392–1400. ISBN 978-1-4673-8851-1. Disponível em: <http://ieeexplore.ieee.org/document/7780524/>. Citado 5 vezes nas páginas 22, 39, 41, 62 e 66.

YU, F. et al. Poi: Multiple object tracking with high performance detection and appearance feature. 2016. Citado 5 vezes nas páginas 53, 62, 66, 83 e 88.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014. p. 818–833. Disponível em: <https://doi.org/10.1007/978-3-319-10590-1_53>. Citado 2 vezes nas páginas 55 e 56.

ZHANG, W. et al. An overview of the PETS 2009 challenge. *Proceedings 11th IEEE International Workshop on PETS*, 2009. Citado na página 36.

ZHENG, L. et al. MARS: A Video Benchmark for Large-Scale Person Re-Identification. In: . [S.l.: s.n.], 2016. p. 868–884. Citado na página 54.

ZHOU, Y. et al. Large-Scale Fiber Tracking Through Sparsely Sampled Image Sequences of Composite Materials. *IEEE Transactions on Image Processing*, v. 25, n. 10, p. 4931–4942, oct 2016. ISSN 1057-7149. Disponível em: <http://ieeexplore.ieee.org/document/7536127/>. Citado na página 20.

ZOPH, B. et al. *Learning Data Augmentation Strategies for Object Detection*. 2019. Citado na página 134.

Apêndices

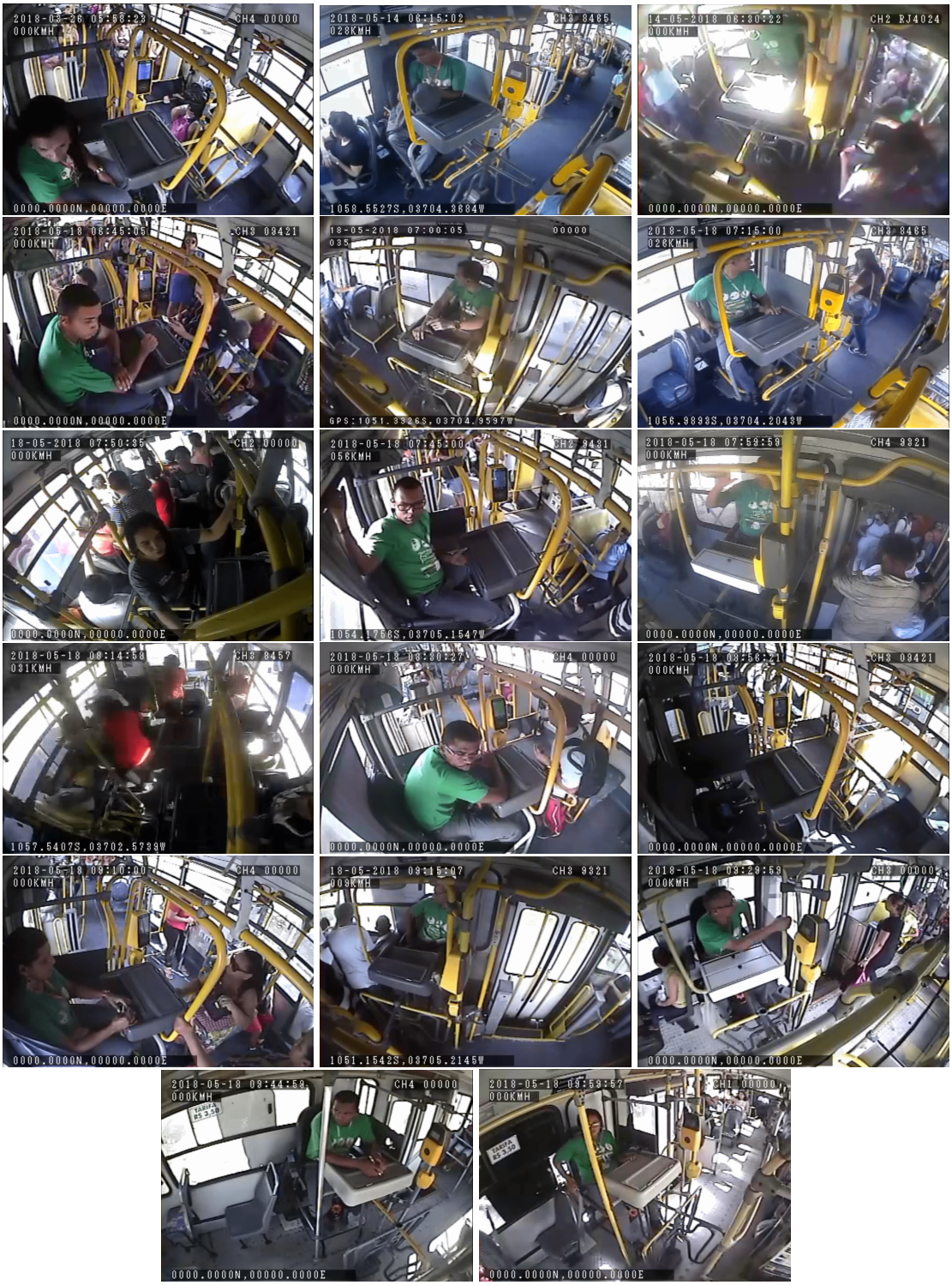
APÊNDICE A – Construção da base BUS Challenge 2018

Durante este trabalho foi construída a base de dados BUS Challenge 2018, a qual é voltada para o treinamento e a avaliação de rastreadores de múltiplos objetos no contexto do rastreamento de passageiros de ônibus. Como discutido na subseção 5.2.1 do Capítulo 5, o principal mérito desta base encontra-se no caráter inédito de seu conteúdo, tendo em vista os *benchmarks* de pedestres apresentados no Capítulo 2. Além disso, a BUS Challenge 2018 permite o estudo e o desenvolvimento de rastreadores com aplicações práticas relacionadas ao monitoramento, à vigilância e à contagem autônoma de passageiros de ônibus, por exemplo. Tais aplicações, inclusive, motivaram a colaboração da empresa de transporte urbano Auto Viação Modelo[1], sediada na cidade de Aracaju, para a construção deste *benchmark* através do fornecimento de vídeos obtidos a partir das câmeras de monitoramento instaladas em seus veículos. A metodologia de construção da BUS Challenge 2018, portanto, incluiu a seleção destes vídeos, a marcação das trajetórias dos passageiros ao longo de cada vídeo selecionado, e a separação dos vídeos e de suas respectivas marcações em sequências de treinamento e teste.

Inicialmente foram selecionados 17 vídeos, com duração média de 15 minutos cada, resolução de $320x280$ *pixels* e frequência de captura igual a 4 quadros por segundo. Cada vídeo foi obtido a partir de câmeras de monitoramento instaladas em veículos diferentes. Além disso, os vídeos selecionados foram registrados em horários subsequentes. Esta estratégia de seleção teve como objetivo maximizar a diversidade das características relacionadas à estrutura dos veículos, à densidade de passageiros, à iluminação do cenário e ao posicionamento das câmeras. A Figura 55 ilustra tais vídeos.

[1] Mais detalhes acerca da empresa podem ser encontrados através do seu endereço eletrônico: <http://www.viacaomodelo.com.br/institucional.php>

Figura 55 – Ilustração dos 17 vídeos inicialmente selecionados para compor a base de dados BUS Challenge 2018. É possível notar a diversidade de posicionamentos da câmera, de níveis de iluminação e de densidades de passageiros.



Fonte: o próprio autor.

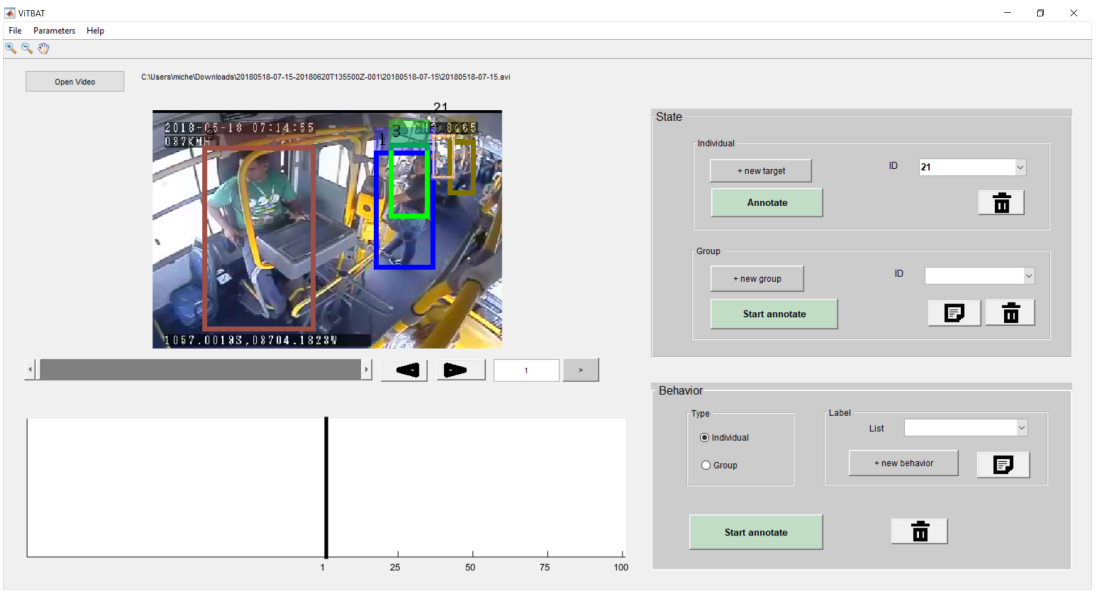
Após a seleção dos vídeos, deu-se início ao processo de marcação da trajetória dos passageiros de ônibus. O protocolo de marcação adotado inspirou-se naquele utilizado pelo *benchmark* MOT Challenge 2016 (MILAN et al., 2016). Este protocolo consiste nas seguintes

instruções:

1. Marcar todas as pessoas, independentemente da posição, postura ou ação que a mesma executa, desde que seja possível estimar suas dimensões com precisão;
2. Cada marcação deve incluir todos os *pixels* da pessoa ao mesmo tempo em que suas dimensões devem ser as menores possíveis;
3. Em caso de oclusão, os *pixels* pertencentes à pessoa devem ser estimados com base em outras características (*e.g.*, proporção do corpo, tamanho da sombra, reflexo em um espelho);
4. Mesmo que parte do corpo de uma pessoa esteja fora da imagem sua marcação também deve incluir todos os seus *pixels*, ou seja, a marcação também ultrapassará os limites da imagem (com base numa estimativa semelhante à realizada nos casos de oclusão);
5. As marcações devem ser feitas assim que uma nova pessoa aparece na imagem, desde que seja possível identificar sua posição e extensão;
6. Da mesma forma, as marcações devem terminar assim que não for mais possível identificar a posição e a extensão de uma pessoa;
7. Caso uma pessoa saia da imagem e apareça em outro ponto sem que sua posição durante o desaparecimento possa ser estimada, uma nova marcação deve ser iniciada;
8. A regra anterior também vale para oclusões.

De modo a marcar a trajetória dos passageiros ao longo dos vídeos selecionados, utilizou-se a ferramenta ViTBAT (BIRESAW et al., 2016). Esta permite a marcação da trajetória de diferentes objetos presentes ao longo de um vídeo, sendo que cada trajetória é automaticamente associada a um identificador gerado pela ferramenta. Além disso, de modo a tornar o processo de rotulação menos desgastante, a ferramenta é capaz de interpolar a marcação da trajetória de determinado objeto uma vez informados seus estados inicial e final. Por fim, a ferramenta ViTBAT gera como saída um arquivo *.txt* contendo em cada linha o identificador, o número do quadro e as dimensões das marcações de cada objeto rotulado. O formato das marcações é definido pela Equação 5.1, sendo idêntico ao utilizado pelo *benchmark* MOT *Challange* 2016. A Figura 56 ilustra o uso da ferramenta.

Figura 56 – Exemplo de rotulação de um dos vídeos da empresa Auto Viação Modelo através do *software* ViTBAT (BIRESAW et al., 2016).



Fonte: o próprio autor.

A aplicação do protocolo de marcação dos vídeos da BUS Challenge 2018 ocorreu entre os dias 15 de maio de 2018 e 24 de julho de 2018. Devido a limitações técnicas, no entanto, apenas 4 dos 17 vídeos selecionados tiveram suas trajetórias marcadas. A Tabela 13 descreve as sequências obtidas ao final do processo de marcação. Ao todo foram geradas 63984 marcações, as quais descrevem 80 trajetórias de passageiros. Como observado através da Tabela 13, em metade das sequências a câmera encontra-se sobre a catraca e direcionada para o corredor, enquanto que na outra metade a câmera encontra-se sobre o cobrador e com vista para a catraca. Além disso, todas as sequências possuem a mesma quantidade de imagens. Dessa forma, as sequências BUS18-03 e BUS18-01 foram destinadas para a partição de treinamento e as sequências BUS18-04 e BUS18-02 foram designadas para teste.

Tabela 13 – Descrição de todas as sequências da base de dados BUS Challenge 2018.

Amostra				
Nome	BUS18-04	BUS18-03	BUS18-02	BUS18-01
FPS	4	4	4	4
Resolução	352x240	352x240	352x240	352x240
Imagens	3601	3601	3601	3601
Trajetórias	33	15	25	7
Marcações	17622	16123	15763	14476
Densidade	4,89	4,5	4,34	4,0
Descrição	Câmera sobre a catraca com vista para o corredor.	Câmera sobre a catraca com vista para o corredor.	Câmera sobre o cobrador com vista para a catraca	Câmera sobre o cobrador com vista para a catraca.

Por fim, como etapa adicional à marcação dos vídeos que compõem a base BUS Challenge 2018, foram geradas detecções que apontam o posicionamento e as dimensões das pessoas presentes em cada quadro das sequências de teste BUS-02 e BUS-04. Estas detecções têm como objetivo servir de entrada para rastreadores baseados no paradigma *tracking-by-detection*, de modo a tornar mais justa a comparação destes algoritmos quando avaliados sobre a base BUS Challenge 2018. Num primeiro momento, buscou-se obter tais detecções de maneira automática através da aplicação de *frameworks* de detecção de objetos baseados em CNN. Para tanto, foi construído um banco de imagens a partir das sequências de treinamento BUS-01 e BUS-03, o qual foi utilizado para treinar os detectores. Dado que os mesmos seguem o paradigma de aprendizado supervisionado, foi preciso marcar em cada imagem de treinamento a posição e as dimensões das detecções desejadas como saída. O protocolo de marcação utilizado foi inspirado naquele aplicado por Everingham et al. (2009). Suas instruções correspondem a:

1. Marcar todas as pessoas, independentemente da posição, postura ou ação que a mesma executa, desde que seja possível visualizar suas dimensões com precisão;
2. Cada marcação deve incluir todos os *pixels* visíveis da pessoa ao mesmo tempo em que suas dimensões devem ser as menores possíveis;
3. Partes do corpo de uma pessoa que estejam fora da imagem devem ser desconsideradas por sua marcação, ou seja, esta não deve ultrapassar os limites da imagem;

A Figura 57 exemplifica a aplicação do protocolo de marcação sobre uma imagem da sequência de treinamento BUS-01. É possível notar que as dimensões das três pessoas marcadas podem ser determinadas com precisão, ao contrário das pessoas que ocupam a parte traseira do

veículo. Além disso, percebe-se que as marcações englobam apenas os *pixels* visíveis daquelas pessoas, de modo que partes ocultas, como pernas e braços, não são incluídas.

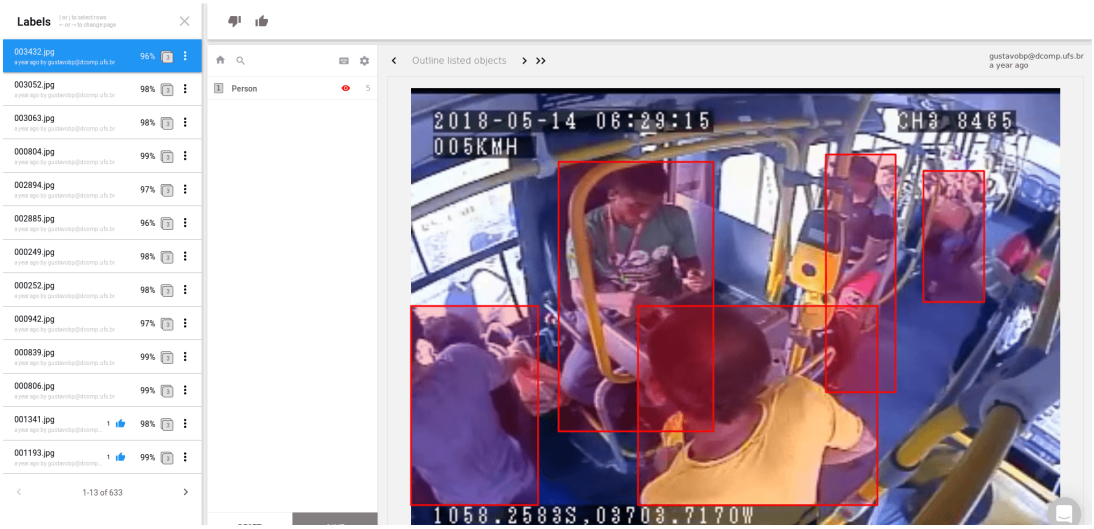
Figura 57 – Exemplo de aplicação do protocolo de marcação de detecções sobre imagem da sequência de treinamento BUS-01.

Fonte: o próprio autor.

O protocolo de marcação das imagens de treinamento dos detectores foi executado entre os dias 15 de julho de 2018 e 21 de setembro de 2018. Este processo foi realizado com o auxílio da ferramenta Labelbox[2], a qual permite a marcação colaborativa de imagens, além do gerenciamento e do controle de todo o processo através de uma interface *Web*. A Figura 58 ilustra a utilização desta ferramenta. Ao final do processo, foram obtidas 2075 marcações de pessoas ao longo de 525 imagens de treinamento.

[2] Para mais informações sobre a ferramenta, consultar sua página *Web* através do endereço eletrônico: <https://labelbox.com/>

Figura 58 – Exemplo de marcação de um dos quadros da sequência de treinamento BUS-03 através do *software* Labelbox. Ao todo foram geradas com o auxílio desta ferramenta 2075 marcações ao longo de 525 imagens.



Fonte: o próprio autor.

Uma vez obtidas as imagens de treinamento e suas respectivas marcações, deu-se início à preparação dos *frameworks* de detecção de objetos. Ao longo deste trabalho foram experimentados diferentes *frameworks*, como Faster-RCNN (REN et al., 2017), SSDLite (SANDLER et al., 2018) e YOLO (REDMON; FARHADI, 2018b). Para os dois primeiros, utilizou-se a API Object Detection (HUANG et al., 2016) disponibilizada pela empresa Google[3]. Já o *framework* YOLO foi manipulado com base na API disponibilizada pelos seus próprios criadores, denominada Darknet[4]. Dada a quantidade reduzida de imagens marcadas neste trabalho, o processo de treinamento destes *frameworks* consistiu no ajuste fino de seus pesos, disponibilizados nos respectivos repositórios das API utilizadas. Estes pesos foram obtidos a partir do treinamento daqueles *frameworks* sobre as bases Imagenet (DENG et al., 2009) e COCO (LIN et al., 2014). Já seu ajuste fino consistiu na aplicação da técnica *fine-tuning* (LI et al., 2019), segundo a qual apenas os pesos das camadas mais externas daqueles *frameworks* são alterados durante o treinamento. A Figura 59 ilustra algumas detecções obtidas sobre imagens das sequências de teste através do *framework* SSDLite, o qual foi ajustado por meio do processo *fine-tunning* ao longo de 45 mil iterações e com base nas imagens de treinamento marcadas ao longo deste trabalho.

[3] O repositório da API Object Detection pode ser acessada via o endereço eletrônico: <https://github.com/tensorflow/models/tree/master/research/object_detection>.

[4] O repositório da API Darknet pode ser acessada através do endereço eletrônico <https://github.com/pjreddie/darknet>.

Figura 59 – Ilustração de detecções obtidas sobre imagens das sequências de teste BUS-02 e BUS-04 através do *framework* SSDLite (SANDLER et al., 2018). Este foi refinado durante 45 mil iterações com base nas imagens de treinamento marcadas ao longo deste trabalho. Nota-se que o detector apresenta alto índice de falsos positivos e falsos negativos, sendo inadequado para o rastreamento por detecção.

Fonte: o próprio autor.

Como observado através da Figura 59, o *framework* SSD Lite ajustado através das imagens de treinamento marcadas ao longo deste trabalho apresenta alto índice de falsos positivos e falsos negativos, de modo que o mesmo é inadequado para o rastreamento por detecção sobre as sequências de teste. Resultados semelhantes foram obtidos com os *frameworks* Faster-RCNN e YOLO. Dentre as razões para estes resultados encontram-se a baixa resolução das imagens utilizadas, o ruído causado pela variação de iluminação e a diversidade de características e padrões apresentada pelas pessoas presentes em tais imagens. De modo a contornar tais dificuldades, buscou-se aumentar artificialmente o tamanho do banco de treinamento por meio de técnica denominada *data augmentation* (ZOPH et al., 2019), segundo a qual as imagens do banco são replicadas e em seguida submetidas aleatoriamente a operações geométricas (*e.g.*, translação e rotação) e à aplicação de diferentes filtros espaciais. Além disso, com o objetivo de reduzir a diversidade de padrões e simplificar o treinamento dos *frameworks* de detecção, experimentou-se a conversão de todas as imagens do banco de treinamento para escala de cinzas. No entanto, mesmo após tais mudanças não foi possível aprimorar de maneira significativa a qualidade dos detectores finais obtidos, o que indica a necessidade da aquisição e da marcação de mais imagens. Dessa forma, considerou-se mais adequado para a base BUS Challenge 2018 a inclusão de detecções obtidas manualmente, de modo que a performance de rastreadores por detecção

avaliados sobre suas sequências de teste não fosse prejudicada pela qualidade das detecções fornecidas como entrada para os mesmos.