UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA (CT)
DEPARTAMENTO DE ENGENHARIA DE TRANSPORTES (DET)
ENGENHARIA CIVIL

CORNÉLIO ALBUQUERQUE DE SOUSA

FERRAMENTA DE EXTRAÇÃO DE TRAJETÓRIAS DE USUÁRIOS DO SISTEMA DE TRANSPORTE EM VÍDEOS DE MONITORAMENTO DO TRÁFEGO USANDO TÉCNICAS DE VISÃO COMPUTACIONAL

FORTALEZA
2023

CORNÉLIO ALBUQUERQUE DE SOUSA

FERRAMENTA DE EXTRAÇÃO DE TRAJETÓRIAS DE USUÁRIOS DO SISTEMA DE TRANSPORTE EM VÍDEOS DE MONITORAMENTO DO TRÁFEGO USANDO TÉCNICAS DE VISÃO COMPUTACIONAL

Trabalho de conclusão de curso apresentado ao Curso de Graduação em Engenharia Civil do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia Civil.

Orientador: Prof. Dr. Manoel Mendonça de Castro Neto.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S696f Sousa, Cornélio Albuquerque de.
Ferramenta de extração de trajetórias de usuários do sistema de transporte em vídeos de monitoramento do tráfego usando técnicas de visão computacional / Cornélio Albuquerque de Sousa. – 2023.
81 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Civil, Fortaleza, 2023.
Orientação: Prof. Dr. Manoel Mendonça de Castro Neto.

1. Visão computacional. 2. Rastreio. 3. Trajetórias. 4. Pedestres. 5. Veículos. I. Título.
CDD 620

CORNÉLIO ALBUQUERQUE DE SOUSA

FERRAMENTA DE EXTRAÇÃO DE TRAJETÓRIAS DE USUÁRIOS DO SISTEMA DE TRANSPORTE EM VÍDEOS DE MONITORAMENTO DO TRÁFEGO USANDO TÉCNICAS DE VISÃO COMPUTACIONAL

Trabalho de conclusão de curso apresentado ao Curso de Graduação em Engenharia Civil do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia Civil.

Aprovada em: 11/12/2023.

BANCA EXAMINADORA

_____

Prof. Dr. Manoel Mendonça de Castro Neto (Orientador)

Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Flávio José Craveiro Cunto

Universidade Federal do Ceará (UFC)

_____

Prof. Dr. José Antônio Fernandes de Macêdo

Universidade Estadual do Ceará (UFC)

A Deus.

Aos meus pais, Lucila e Antônio, e ao meu irmão Ricardo.

AGRADECIMENTOS

Aos meus pais, Lucila e Antônio, pela dádiva da vida e por todo o seu suporte e amor desde o momento em que eu fui concebido.

Ao meu irmão, Ricardo, por ser uma referência que me guia e por seu amor e cuidado comigo desde sempre.

Aos amigos que fiz ao longo da graduação, pelas alegrias compartilhadas, pelas ajudas oferecidas e por estarem presentes em momentos importantes da minha vida.

Ao meu orientador, Prof. Dr. Manoel Mendonça de Castro Neto, pela oportunidade de ser seu orientando, pelos conhecimentos compartilhados e pelo suporte ao longo de todo o processo.

Ao Prof. Dr. Weslay Vieira de Araújo, pelo seu suporte desde a minha participação na OBMEP até o meu ingresso na universidade.

Aos meus colegas do grupo Visão Computacional Tráfego, pelo compartilhamento e construção de conhecimentos, alegrias compartilhadas e suporte mútuo.

Aos professores que tive ao longo da graduação, pelo compartilhamento dos conhecimentos que possuem.

RESUMO

Os estudos do tráfego na escala microscópica focam nos movimentos individuais dos usuários do sistema de transporte. Como exemplos desse tipo de estudo, têm-se as modelagens comportamentais e as avaliações da segurança viária. Em vista disso, dados de trajetória de veículos e pedestres se mostram extremamente úteis para o desenvolvimento desses estudos, porém a sua aquisição é um desafio, frente à enorme quantidade de observações que necessitam ser coletadas a cada instante de tempo. No entanto, com o sucesso crescente do campo da Inteligência Artificial, em específico da Visão Computacional, surgiram novos e mais eficientes modelos e algoritmos capazes de automatizar a extração de trajetórias de objetos em vídeos. Isto posto, a presente monografia busca construir uma ferramenta de extração de trajetórias de usuários do sistema de transporte em vídeos de monitoramento do tráfego a partir do uso de modelos e algoritmos de visão computacional bem estabelecidos na literatura e disponibilizados gratuitamente. Os usuários aqui referidos são os pedestres, ciclistas, automóveis, caminhões e ônibus. A ferramenta construída foi baseada no modelo de detecção de objetos *YOLOv7* (WANG *et al.*, 2022) e no algoritmo de rastreio de objetos *StrongSORT* (DU *et al.*, 2023). Após a validação da ferramenta em tarefas de contagem veicular classificatória, coleta dos instantes de entrada e de saída de veículos na faixa de pedestres, contagem de travessias de pedestres e coleta do atraso e do tempo de travessia dos pedestres, foram obtidos resultados promissores nas tarefas referentes à veículos, mas notou-se maior espaço para melhoria da ferramenta em vista dos resultados obtidos nas tarefas relacionadas às travessias de pedestres.

Palavras-chave: visão computacional; rastreio; trajetórias; pedestres; veículos.

ABSTRACT

The traffic studies at the microscopic scale focus on the individual movements of transportation system users. Examples of such studies include behavioral modeling and road safety assessments. In this setting, trajectory data of vehicles and pedestrians are extremely useful for the development of these studies, but their acquisition poses a challenge due to the vast amount of observations that need to be collected at each moment in time. However, with the growing success of the field of Artificial Intelligence, specifically Computer Vision, new and more efficient models and algorithms have emerged capable of automating the extraction of object trajectories in videos. Given this context, this undergraduate thesis aims to construct a tool for extracting trajectories of transportation system users from traffic monitoring videos using well-established, open-source computer vision models and algorithms available in the literature. The users referred to here are pedestrians, cyclists, cars, trucks, and buses. The constructed tool was based on the YOLOv7 object detection model (WANG *et al.*, 2022) and the StrongSORT object tracking algorithm (DU *et al.*, 2023). After validating the tool in the tasks of classificatory vehicle counting, determination of the instants when vehicles enter and exit the pedestrian crosswalk, pedestrian crossing count and estimation of pedestrian delay and crossing time, promising outcomes emerged from the tasks related to vehicles, however, was acknowledged greater room for improvement in the tool's quality given the results obtained in the tasks related to pedestrian crossings.

Keywords: computer vision; tracking; trajectories; pedestrians; vehicles.

SUMÁRIO

1 INTRODUÇÃO

Os estudos do tráfego na escala microscópica são aqueles onde os veículos [e pedestres] são analisados individualmente, sendo que a posição e a velocidade de cada um definem o estado do sistema (BOGO; GRAMANI; KAVISKI, 2015). Podemos citar como exemplos desse tipo de estudo as modelagens comportamentais e alguns métodos de avaliações da segurança viária. Em vista disso, o conhecimento da localização espacial e temporal dos usuários é importante nesta escala de estudo, como, por exemplo, para validação de modelos de microssimulação e avaliação dos riscos de colisão entre usuários.

Entretanto, a aquisição dos dados de localização dos usuários se mostra problemática, como é o exemplo dos estudos de segurança viária com base em conflitos – proximidade espaço-temporal entre usuários caracterizando um risco de colisão, onde metodologias tradicionais de coleta de dados a partir de observações humanas em campo estavam contribuindo no impedindo de aplicações extensivas desse tipo de estudo (SAUNIER; SAYED, 2007). Por outro lado, técnicas de visão computacional vêm sendo aplicadas em diversas áreas de estudo, inclusive na de operação do sistema de transporte, com o objetivo de rastrear objetos em vídeos, obtendo suas trajetórias ao longo dos frames de forma automática (ALVER *et al.*, 2021; CASTRO JUNIOR; CASTRO NETO; CUNTO, 2021; SAUNIER; SAYED, 2007; ZHANG *et al.*, 2020).

Tendo isso em vista e dado que sistemas de monitoramento do tráfego são cada vez mais comuns nas grandes cidades, como é o caso do sistema mantido em Fortaleza-CE pelo Controle de Tráfego em Área de Fortaleza (CTAFOR), a aplicação de ferramentas de visão computacional para coleta de dados de trajetórias pode gerar uma grande fonte de dados de qualidade para aplicações diversas, como é caso de estudos microscópicos do tráfego.

1.1 Problemática e justificativa

Os métodos tradicionais de coleta de dados do tráfego a partir de observadores humanos são arcaicos e apresentam uma série de problemáticas, como os custos com os profissionais de coleta e possíveis treinamentos, subjetividade dos observadores, impossibilidade humana de processar múltiplas situações ocorrendo em um intervalo de tempo comum e inviabilidade de coletas de longa duração.

Em vista disso, foram surgindo novos métodos de coleta para suprir as necessidades da aquisição de dados de tráfego, como, por exemplo, usando sensores mecânicos. Mas

também, com os avanços alcançados no campo da visão computacional, surgiram diversos modelos e algoritmos capazes de automatizar tarefas ligadas ao processamento de cenas do mundo real, como detecção e rastreio de objetos, identificação de ações e reidentificação de pessoas. Dado que muitos dos algoritmos e modelos de visão computacional são passíveis de calibração e/ou retreinamento para aplicação em diferentes problemas-alvo, bem como muitos são disponibilizados em forma de código aberto, se torna possível a aplicação desses na automatização de coleta de dados e como ferramentas auxiliares em estudos do tráfego a partir de gravações do trânsito, como é o caso dos estudos desenvolvidos por Sun *et al.* (2021) e Zhang *et al.* (2020).

Por outro lado, trazendo o foco especificamente para a esfera microscópica do tráfego, mesmo já havendo disponibilidade ferramental para construção de sistemas automatizados de coleta de dados a partir de vídeos, a oferta e acesso destes de forma livre ainda é pequena, havendo sobretudo estudos estrangeiros cuja generalização para o contexto brasileiro não é direta.

1.2 Objetivos

O objetivo geral desta monografia é construir uma ferramenta automatizada de extração de trajetórias de usuários do sistema de transporte em vídeos de monitoramento do tráfego usando técnicas de visão computacional. Os usuários, neste trabalho, são os pedestres, ciclistas, automóveis, ônibus e caminhões. Ademais, as técnicas de visão computacional escolhidas para atacar o problema são referentes ao paradigma de rastreio por detecção dos sistemas de rastreio de objetos. Desse modo, para o alcance do objetivo geral, tem-se os seguintes objetivos específicos:

a) treinar um modelo de detecção de objetos em imagens na tarefa de detecção de usuários do sistema de transporte usando vídeos de monitoramento do tráfego;
b) rastrear automaticamente os usuários do sistema de transporte para obtenção de suas trajetórias usando um algoritmo de rastreio por detecção;
c) validar a aplicabilidade da ferramenta de extração de trajetórias em tarefas de contagem classificatória de veículos, contagem classificatória de conversões, contagem de travessias, coleta dos instantes de entrada e de saída de veículos na faixa de pedestres e coleta dos atrasos e dos tempos de travessia.

2 REVISÃO DA LITERATURA

Esta seção apresenta os fundamentos dos dois principais componentes necessários para construção da ferramenta-alvo do objetivo geral, sendo estes os métodos de rastreio de múltiplos objetos e os modelos de detecção de objetos. Ademais, apresenta trabalhos e pesquisas que se assemelham a pesquisa desta monografia.

2.1 Detecção de objetos

Atualmente, os principais sistemas modernos de detecção de objetos são de estágio único (*one-stage detectors*) ou duplo (*two-stage detectors*). Os modelos de estágio único apresentam maior simplicidade e velocidade, porém acabam ficando atrás dos modelos de estágio duplo no quesito acurácia (LIN *et al.*, 2018). Ainda segundo Lin *et al.* (2018), modelos de estágio único são aqueles aplicados sobre uma densa amostra de localizações candidatas de objetos, enquanto os de estágio duplo são aplicados sobre amostras esparsas, obtidas a partir de métodos de proposição de regiões.

Dentre os modelos de estágio único se pode citar a família *You Only Look Once* (*YOLO*) (BOCHKOVSKIY; WANG; LIAO, 2020; REDMON *et al.*, 2016; REDMON; FARHADI, 2016; REDMON; FARHADI, 2018) e o modelo *RetinaNet* (LIN *et al.*, 2018). Quanto aos de estágio duplo, o principal representante são os modelos da família *Regions with CNN features* (*R-CNN*) (GIRSHICK *et al.*, 2014; GIRSHICK, 2015; REN *et al.*, 2016).

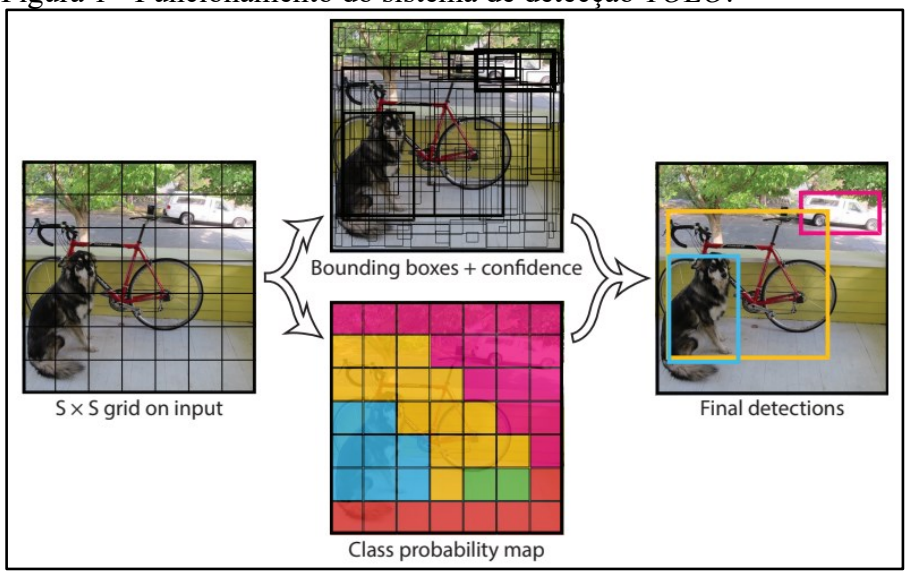
2.1.1 Sistemas de estágio único

2.1.1.1 You Only Look Once (YOLO)

A primeira versão dos detectores de objetos *YOLO* (REDMON *et al.*, 2015) consiste em uma rede neural convolucional profunda (*deep convolutional neural net*) com camadas densas (*fully connected layers*) no seu fim. A porção convolucional da rede neural é responsável pelo mapeamento de características em imagens (produção de *features maps*) e a porção de camadas densas assume o papel de interpretação dessas características através de um processo de regressão. A saída final do modelo é uma série de coordenadas de caixas delimitadoras (*bounding boxes*), de valores que buscam refletir a confiança do modelo quanto a presença de algum objeto dentro de cada caixa delimitadora (*confidence scores*) e de probabilidades

condicionais que representam a confiança do modelo quanto a qual classe o objeto dentro de alguma caixa delimitadora pertence (*conditional class probabilities*). O funcionamento do modelo é exemplificado na Figura 1 abaixo.

Figura 1 - Funcionamento do sistema de detecção *YOLO*.



Fonte: Redmon *et al.* (2016, p. 02).

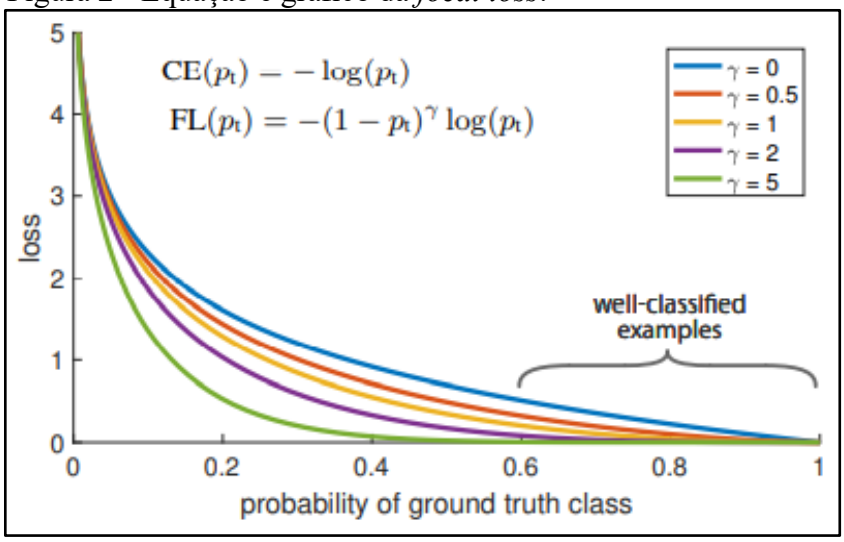
Em 2016 foi lançada a segunda versão da arquitetura *YOLO* (REDMON; FARHADI, 2016). Dentre as mudanças, destaca-se aqui a substituição das camadas densas por camadas convolucionais, utilização de caixas delimitadoras base (*anchor boxes*) para predição das caixas delimitadoras dos objetos e o treinamento multiescala para adição de robustez quanto às mudanças de dimensão das imagens. Nos anos que se seguiram a família *YOLO* recebeu diversas novas versões à medida que novas formas de ampliar a eficiência de redes convolucionais foram sendo descobertas, como *You Only Learn One Representation* (*YOLOR*) (WANG *et al.*, 2021) e *YOLOv7* (WANG *et al.*, 2022).

2.1.1.2 RetinaNet

O modelo *RetinaNet* foi construído durante os experimentos de Lin *et al.* (2018) quanto a chamada *focal loss*, uma heurística de modificação da *cross entropy loss* para endereçar o problema de desbalanceamento de classes presente no treinamento de modelos de detecção, principalmente entre plano de fundo (*background*) e o plano frontal (*foreground*, onde se encontra os objetos a serem detectados) das imagens. Ainda segundo Lin *et al.* (2018), a ideia geral por trás da *focal loss* é a diminuição do peso de importância dos exemplos bem

classificados, de modo que a contribuição desses exemplos para a função custo seja reduzido, aumentando assim o foco nos exemplos de difícil classificação e classificados erroneamente. A Figura 2 apresenta a função custo proposta para diferentes valores do parâmetro γ.

Figura 2 - Equação e gráfico da *focal loss*.



Fonte: Lin *et al.* (2018, p. 01).

O modelo *RetinaNet* é composto de um módulo de extração de características e de sub-redes para classificação de objetos e regressão de caixas delimitadoras a partir das características obtidas. A extração de características é realizada por uma *Feature Pyramid Network* (*FPN*) modificada, gerando mapas de características multiescala na forma de uma pirâmide de características. Para cada escala de mapas de características (nível da pirâmide) há duas *Fully Convolutional Networks* (*FCN*), uma para tarefa classificação e outra para tarefa de regressão (LIN *et al.*, 2018).

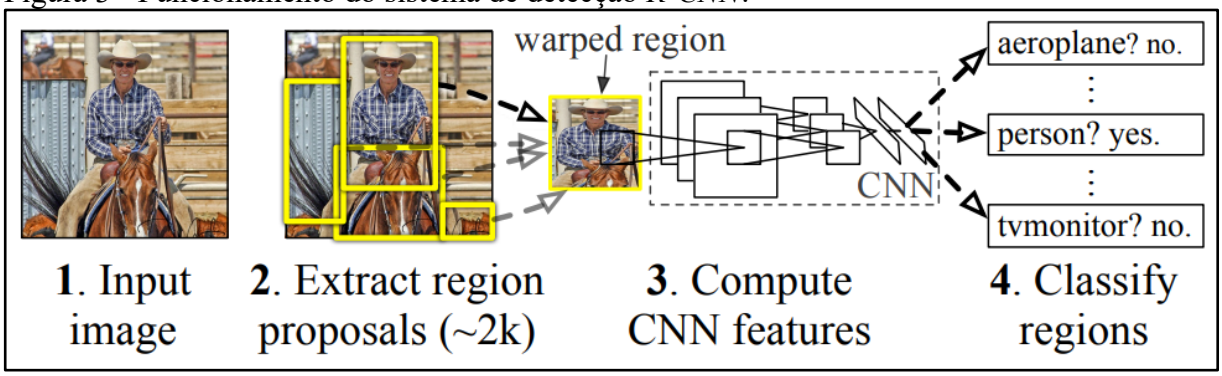
2.1.2 Sistemas de estágio duplo

2.1.2.1 Regions with CNN features (R-CNN)

A alcunha *R-CNN* representa uma família de sistemas de detecção de objetos em imagens digitais de estágio duplo, onde primeiramente é realizado a proposição de regiões que possam conter objetos para só então ser feita a detecção de objetos nessas regiões. A Figura 3 ilustra o funcionamento da primeira versão dos modelos *R-CNN*, lançada em 2014. Este sistema é composto de três módulos principais: o primeiro módulo usa o método *selective search* para gerar proposições de regiões que possam conter objetos quaisquer (independente de classes); o

segundo consiste de uma rede neural convolucional para extração de características de cada uma das regiões propostas; e o terceiro módulo é constituído de um conjunto de *Support Vector Machines* (*SVMs*) para predição de classes para cada região a partir das características geradas pelo segundo módulo (GIRSHICK *et al.*, 2014). Ademais, Girshick *et al.* (2014) apresentou resultados práticos que mostram que um processo de pré-treino supervisionado com uso de um grande *dataset* auxiliar, seguido de um treinamento mais refinado com uso de um *dataset* de domínio específico para o problema alvo, pode elevar consideravelmente a capacidade de aprendizagem da *RNC* em situações em que a quantidade de dados é escassa.

Figura 3 - Funcionamento do sistema de detecção *R-CNN*.



Fonte: Girshick *et al.* (2014, p. 01).

Em 2015 foi publicado o artigo referente a segunda versão da família *R-CNN*, a qual foi intitulada *Fast R-CNN*, contendo diversas modificações para melhoria da velocidade de treinamento e teste do sistema e da acurácia dos resultados (GIRSHICK, 2015). Ainda segundo Girshick (2015), o método *Fast R-CNN* trouxe um algoritmo que integra as etapas de treinamento da Rede Neural Convolucional para extração de características e das Máquinas de Vetores Suporte para classificação de objetos, o qual, juntamente com um método de compartilhamento de computação, simplifica e acelera o processo de treinamento e teste.

Outras versões e melhorias foram surgindo ao longo do tempo, mas vale destacar a versão *Faster R-CNN* publicada por Ren *et al.* (2016). A *Faster R-CNN* passou a utilizar uma Rede Neural Convolucional para proposição de regiões, intitulada *Region Proposal Network* (*RPN*), o que trouxe grande aumento de performance para o sistema e possibilitou a integração de todos os seus módulos, trazendo o modelo para mais próximo da zona de detecção em tempo real (REN *et al.*, 2016).

2.2 Rastreio de múltiplos objetos

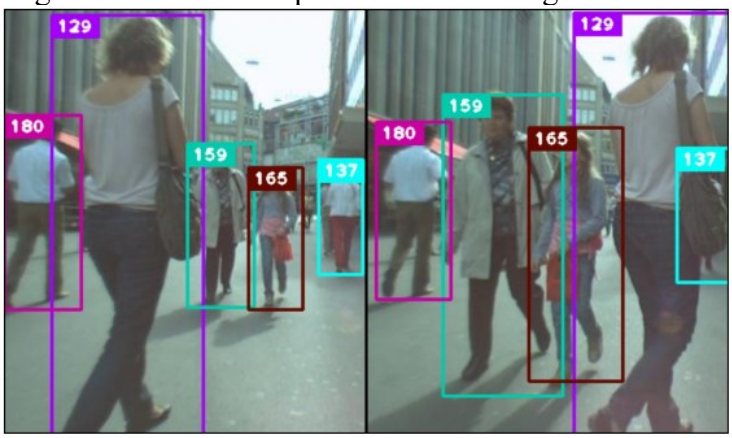
O rastreio de múltiplos objetos, tradução livre de *Multiple Object Tracking* (*MOT*), também conhecido como rastreio de múltiplos alvos - *Multiple Targets Tracking* (*MTT*), é um problema desafiador e de grande importância no campo da visão computacional devido às suas inúmeras aplicações práticas, podendo ser particionado de forma simplista nas tarefas de localização de múltiplos objetos, manutenção de suas identidades e retorno de suas trajetórias dado um vídeo como fonte bruta de dados (LUO *et al.*, 2022).

Ainda segundo Luo *et al.* (2022), a maioria das soluções desenvolvidas para resolver o problema de rastreio de múltiplos objetos podem ser divididas em dois grupos, a depender do método de inicialização empregado, sendo estes o rastreio por detecção (*detection-based tracking* ou *tracking-by-detection*) e o rastreio livre de detecção (*detection-free tracking*), sendo o primeiro paradigma o mais comum, uma vez que o surgimento e desaparecimento de objetos é tratado automaticamente, diferente da abordagem sem uso de detectores que necessita de inicialização manual quando objetos aparecem. Ademais, os estudos de Bewley *et al.* (2016) supõem que a qualidade dos algoritmos de rastreio por detecção depende grandemente da performance da detecção e podem se beneficiar fortemente do recente desenvolvimento alcançado na área de detecção de objetos.

2.2.1 Métodos de rastreio SORT

Simple Online and Realtime Tracking (*SORT*) é um algoritmo de rastreio de múltiplos objetos que utiliza de detecções fornecidas por um detector de objetos para estimar as trajetórias dos objetos-alvo, ou seja, é um algoritmo de rastreio por detecção. Sua abordagem foca em simplicidade e eficiência, tirando proveito dos avanços nas técnicas de detecção de objetos que permitiram a produção de dados de detecção com maior qualidade (BEWLEY *et al.*, 2016). A Figura 4 abaixo ilustra o resultado desse algoritmo de rastreio em dois quadros de um vídeo.

Figura 4 - Rastreio de pedestres usando algoritmo *SORT*.



Fonte: Wojke, Bewley e Paulus (2017, p. 01).

Segundo Bewley *et al*. (2016), o método *SORT* é do tipo *online*, o que significa que o algoritmo apenas utiliza informações do frame atual (frame que está sendo processado pelo algoritmo) e dos frames passados para construção das trajetórias dos objetos, processando o vídeo em forma de *stream*. Para cada novo objeto que surge no vídeo, o método *SORT* gera um "alvo" que será responsável por tentar seguir o objeto ao longo de sua trajetória. O surgimento de um objeto nada mais é que uma detecção no frame atual que não foi associada a nenhum alvo ativo, necessitando, portanto, a criação de um novo alvo para ser associado à ela. Um alvo é desativado quando fica uma certa quantidade consecutiva de frames sem ser associado a nenhuma nova detecção, logo, cada alvo armazena internamente o número de frames que se passaram desde a sua última associação, valor considerado como sua "idade". Os alvos têm a capacidade de estimar a localização dos objetos que estão rastreando em quadros futuros, gerando detecções (coordenadas de caixas delimitadoras) que buscam antecipar as próximas posições dos objetos com base nas informações de trajetória armazenadas até o momento. Para cada novo frame do vídeo, o método *SORT* recebe como dados de entrada as detecções de objetos no respectivo frame e tenta associá-las aos alvos existentes de modo a minimizar a distância entre estas detecções e as detecções estimadas pelos alvos ativos. Ademais, o método *SORT* implementa uma limitação na associação entre uma detecção e um alvo com base na distância entre estes, onde distâncias maiores que um certo limiar não são permitidas.

A primeira versão do algoritmo *SORT* não emprega técnicas de reidentificação de objetos, de modo a evitar o aumento da complexidade do método, buscando manter boa eficiência e processamento em tempo real (BEWLEY *et al.*, 2016). Contudo, em 2017, foi publicado o artigo *Simple Online and Realtime Tracking with a Deep Association Metric* (WOJKE; BEWLEY; PAULUS, 2017), o qual apresentou o algoritmo *DeepSORT*. Esse algoritmo, por meio de uma série de modificações, teve a intenção de fortalecer a capacidade

de reidentificação de objetos da versão anterior. Dentre estas modificações, destaca-se aqui a utilização de uma métrica para quantificar a semelhança de aparência entre objetos e o uso da distância de Mahalanobis no lugar da distância *IoU* (distância que usa como base a métrica *Intersection over Union*, também conhecida como Índice Jaccard ou simplesmente *IoU*, a qual quantifica a sobreposição de dois conjuntos) para mensurar a proximidade entre duas detecções. Dessa forma, o algoritmo *DeepSORT* passou a associar detecções à alvos buscando minimizar tanto a distância entre detecções, dessa vez a distância de Mahalanobis, quanto a discrepância entre a aparência de objetos, utilizando um fator para ponderar a importância de cada critério. Porém, a etapa de associação de detecções à alvos passou a ser executada no algoritmo *DeepSORT* em forma de cascata, de modo que as detecções são associadas aos alvos de forma incremental, começando com alvos de menor idade (menor número de frames desde a sua última associação com uma detecção) e gradualmente chegando aos alvos de maior idade. Ademais, a aparência dos objetos é descrita por um vetor de características obtido usando uma rede neural convolucional treinada em um conjunto de dados de reidentificação de pessoas. A diferença na aparência entre dois objetos é medida pela distância cosseno entre seus vetores de características correspondentes (WOJKE *et al.*, 2017).

Por fim, Du *et al.* (2023) conduziram uma série de melhorias ao algoritmo *DeepSORT*, produzindo o que eles chamaram de *StrongSORT*. Dentre estas melhorias, destaca-se aqui a mudança do modelo de detecção de objetos, modificações no filtro Kalman com a intenção de torná-lo robusto contra detecções de baixa qualidade (baixo valor de confiança) e aplicação de compensação de movimento de câmera (DU *et al.*, 2023).

2.2.2 Avaliação de métodos de rastreio de múltiplos objetos

Os esforços dedicados pela comunidade de visão computacional para criar *datasets* e métricas de avaliação para servir como referência em suas respectivas áreas de estudo mostraram-se úteis para o avanço do estado da arte, mesmo tendo em vista a possível existência de problemas com estas referências (LEAL-TAIXÉ *et al.*, 2015). Tomando como exemplo as tarefas de classificação e detecção de objetos em imagens, tem-se o *PASCAL Visual Object Classes (VOC) Challenge* (EVERINGHAM *et al.*, 2012), que fornece conjuntos de imagens anotadas, um aplicativo padronizado para avaliação dos resultados dos algoritmos, bem como competições e *workshops* anuais. Quanto ao *MOT*, podemos citar como exemplos o *PETS Challenge* (FERRYMAN; ELLIS, 2010), o *MOTChallenge benchmark* (LEAL-TAIXÉ *et al.*, 2015) e vários outros esforços na criação de padrões e referências para avaliação de soluções

de rastreio (BERNARDIN; STIEFELHAGEN, 2008; DENDORFER *et al.*, 2020; SMITH *et al.* 2005).

2.2.2.1 Métricas CLEAR MOT

Bernardin e Stiefelhagen (2008), com o intuito de contribuir na problemática da falta de procedimentos e métricas gerais para avaliação da performance de soluções *MOT*, propuseram as métricas *CLEAR MOT*, as quais foram desenvolvidas na expectativa de que métricas úteis sejam poucas em número, mas ainda expressivas, terem poucos parâmetros livres, expressarem de forma clara e intuitiva as qualidades do rastreador avaliado e serem gerais o suficiente para serem aplicáveis em diversos tipos de rastreio. Desse modo, as métricas *Multiple Object Tracking Precision* (*MOTP*) e *Multiple Object Tracking Accuracy* (*MOTA*) foram desenvolvidas tendo em mente dois critérios principais: serem capazes de julgar a precisão do rastreador em determinar a localização exata dos objetos e capazes de refletir a sua habilidade de rastrear as configurações dos objetos ao longo do tempo (rastrear corretamente a trajetória dos objetos, produzindo exatamente uma trajetória por objeto).

De forma resumida, o procedimento necessário para quantificação das métricas *CLEAR MOT* é representado pelo passo-a-passo a seguir, o qual é executado para cada ponto de tempo t da sequência (cada frame, em caso de vídeos), dado os conjuntos $o_t = \{o_0, o_1, \ldots, o_n\}$ e $h_t = \{h_0, h_1, \ldots, h_n\}$, que representam, respectivamente, os objetos visíveis (os *ground truths*) e as hipóteses do rastreador (predições do sistema de rastreio) para o instante t; o conjunto M_{t-1}, que representa os pares objeto-hipótese vigentes até o instante anterior, inicialmente vazio; a função $d(o_i, h_j)$, que calcula a distância (ou a discrepância) entre o objeto o_i e a hipótese h_j; e T, que representa a distância máxima que um par objeto-hipótese pode assumir:

a) para cada par objeto-hipótese (o_i, h_j) em M_{t-1}, verificar se este continua válido no instante atual, ou seja, se o objeto o_i ainda existe em o_t, se a hipótese h_j ainda existe em h_t e se a distância objeto-hipótese entre os atuais o_i e h_j não ultrapassa o limiar de associação T, ou seja, $d(o_i, h_j) \leq T$. Em caso afirmativo, manter o par (o_i, h_j) no instante atual;
b) para os objetos restantes em o_t que ainda não foram associados com alguma hipótese em h_t, buscar pareamentos através de associações bijetivas, de modo a minimizar a distância total dos pares e de modo a não ter pares que excedam o

limiar T (problema de otimização de custo de associação). Para cada correspondência (o_k, h_l) resultante da resolução do problema de otimização, caso esta contradiz uma associação vigente (o_k, h_g) presente em M_{t-1}, contar como associação incompatível (representa uma mudança de identidade) e substituir (o_k, h_g) por (o_k, h_l) em M_{t-1}, caso não, adicioná-la à M_{t-1}. Dessa forma, considerar mme_t como o número de associações incompatíveis no instante t;

c) ao fim dos dois passos anteriores, está completo as associações objeto-hipóteses do instante atual. Considerar c_t como o número de pares associados no instante t e d_t as distâncias objeto-hipótese de cada associação. Ademais, M_{t-1} agora corresponde a M_t;

d) contar todos os objetos em o_t que não foram associados a uma hipótese como falhas de rastreio (m_t) e contar todas as hipóteses em h_t que não foram associados com um objeto como falsos positivos (fp_t). Ademais, considerar g_t como o número de objetos presentes no instante t (número total de objetos em o_t);

e) repetir os passos a) à d) para o próximo instante.

Uma vez realizado esse procedimento para toda a sequência, as métricas *MOTP* e *MOTA* podem ser calculadas utilizando as equações 1 e 2 abaixo.

$$MOTP = \frac{\sum_{t=1}^{L} \sum_{j=1}^{c_t} d_{t,j}}{\sum_{t=1}^{L} c_t} \tag{1}$$

$$MOTA = 1 - \frac{\sum_{t=1}^{L} (mme_t + m_t + fp_t)}{\sum_{t=1}^{L} g_t} \tag{2}$$

Onde L representa o número total de instantes na sequência sendo avaliada.

Em vista da Equação 1, nota-se que a métrica *MOTP* quantifica a capacidade média do rastreador em estimar precisamente as posições dos objetos, considerando apenas os segmentos de trajetória de objetos que foram associados à hipóteses de trajetória estimadas pelo rastreador, desconsiderando falsos negativos e falsos positivos e independente da capacidade do rastreador em manter a identidade dos objetos. Segundo a implementação original da métrica *MOTP*, quanto menor o seu valor, melhor o desempenho do sistema de rastreio, sendo que a

unidade da métrica fica a critério da função que computa a distância entre os pares objeto-hipótese. Ademais, uma variação da métrica *MOTP* surge a partir da utilização de uma função de similaridade em vez da função de dissimilaridade, de modo que quanto maior o valor da métrica *MOTP*, melhor a qualidade de localização do rastreio (MILAN *et al.*, 2016).

Quanto a métrica *MOTA*, definida na Equação 2, esta busca resumir em um único valor a capacidade do método de rastreio em evitar falhas de detecção (m_t), falsos positivos (fp_t) e mudanças de identidade (mme_t). Um ponto interessante de ser ressaltado é que o valor máximo da métrica *MOTA* é 1, o qual informa que o rastreador não apresenta falhas de detecção, falsos positivos e nem mudanças de identidade ao longo de toda a sequência, porém, não possui valor mínimo, podendo alcançar valores menores que 0, o que ocorre quando o sistema de rastreio apresenta inúmeros falsos positivos e mudanças de identidade ao longo da sequência. Outro ponto de atenção quanto a métrica *MOTA* é o fato de ela não levar em conta a fusão de duas ou mais trajetórias reais em uma única trajetória predita, falha esta nomeada de fusão (*merge*) nos estudos de Ristani *et al.* (2016).

2.2.2.2 Métricas de Identificação

Ristani *et al.* (2016) classifica as métricas *CLEAR MOT* como baseadas em eventos (*event-based*), alegando que estes tipos de métricas ajudam a identificar a origem de alguns erros, sendo, portanto, informativas para os envolvidos no projeto do sistema de rastreio. Porém, ainda segundo Ristani *et al.* (2016), do ponto de vista dos usuários da aplicação, a preservação da identidade é crucial:

> Rastreio de Múltiplos Alvos têm sido tradicionalmente definido como seguir continuamente múltiplos objetos de interesse. Por causa disso, métricas de performance existentes, como as métricas *CLEAR MOT*, reportam quão frequentemente um rastreador produz diferentes tipos de decisões errôneas. Nós defendemos que alguns usuários do sistema de rastreio podem estar mais interessados em quão bem eles conseguem determinar a todo tempo quem está onde. (RISTANI *et al.*, 2016, p. 01, tradução nossa).

Dessa forma, Ristani *et al.* (2016) propõe duas medidas baseadas nas identidades dos objetos (*identity-based measures*) para avaliar quão bem as identidades preditas conformam com as verdadeiras identidades dos objetos, desconsiderando onde e por que os erros ocorrem: *Identification Precision* (*IDP*) e *Identification Recall* (*IDR*).

Tendo isso em vista, para mensurar a capacidade de um sistema de rastreio em manter a identidade de um objeto, primeiro é necessário definir qual das diversas identidades

associadas a trajetória de um objeto deve ser considerada a "correta", de modo que qualquer ponto da trajetória que não condizer com esta identidade é um ponto em que o rastreador está em erro. Ristani *et al.* (2016) optaram por um método de associação entre trajetórias computadas (trajetórias preditas pelo sistema de rastreio, as identidades) e trajetórias reais (trajetórias anotadas dos objetos) de modo a minimizar globalmente o número de falhas de identificação. Uma vez determinado os pares trajetória-identidade, as métricas podem ser calculadas usando as equações 3 a 8 a seguir.

$$IDFN = \sum_{\tau \in AT} \sum_{t \in T_\tau} m(\tau, \gamma_m(\tau), t, \Delta) \tag{3}$$

$$IDFP = \sum_{\gamma \in AC} \sum_{t \in T_\gamma} m(\tau_m(\gamma), \gamma, t, \Delta) \tag{4}$$

$$IDTP = \sum_{\tau \in AT} len(\tau) - IDFN = \sum_{\gamma \in AC} len(\gamma) - IDFP \tag{5}$$

$$IDP = \frac{IDTP}{IDTP + IDFP} \tag{6}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{7}$$

$$IDF_1 = \frac{2\,IDTP}{2\,IDTP + IDFP + IDFN} \tag{8}$$

Onde τ representa a trajetória real de um objeto (*ground truth*),AT é o conjunto das trajetórias reais que foram associadas a trajetórias computadas, t é um instante de tempo que representa um ponto da trajetória,T_τ é o conjunto de instantes de tempo que formam a trajetória τ, $\gamma_m(\cdot)$ é uma função que mapeia trajetórias reais às suas trajetórias computadas associadas, $m(\cdot)$ é uma função booleana que retorna 1 se um determinado ponto da trajetória está em erro de rastreio (identificação errônea) ou 0 caso contrário, Δ é um limiar usado pela função $m(\cdot)$ para validar um instante t da associação objeto-identidade com base na proximidade do objeto real com a sua localização prevista pelo rastreio, γ representa uma trajetória computada pelo sistema de rastreio, AC é o conjunto das trajetórias computadas que foram associadas a trajetórias reais, T_γ é o conjunto de instantes de tempo que formam a trajetória computada γ,

$\tau_m(\cdot)$ é a inversa de $\gamma_m(\cdot)$ e $len(\cdot)$ é uma função que retorna o número total de instantes de tempo (duração temporal) de uma trajetória qualquer.

Dessa forma, a métrica *Identification Precision* (*IDP*) é a razão entre o número de detecções que corretamente localizam e identificam os objetos presentes no vídeo, chamadas de *Identification True Positive* (*IDTP*), e o número total de pontos/detecções presentes nas trajetórias preditas pertencentes à AC. Logo, a métrica *IDP* indica quantos por cento das detecções preditas pelo sistema de rastreio acertam tanto a localização quanto a identidade dos objetos. Já a métrica *Identification Recall* (*IDR*) é a razão entre o número de detecções que predisseram corretamente a localização e a identidade dos objetos existentes (*IDTP*) e o número total de pontos/detecções presentes nas trajetórias reais (*ground truth detections*) pertencentes à AT. A métrica *IDR* também está contida no intervalo $[0,1]$, porém esta indica quantos por cento de todas as aparições de objetos o sistema de rastreio consegue localizar e identificar corretamente. As métricas IDP e IDR juntas permitem avaliar como o sistema de rastreio balanceia a sua precisão e sua sensibilidade, enquanto a métrica IDF_1 fornece um único valor que busca exprimir ambas as métricas através de uma média harmônica.

2.2.2.3 Métrica HOTA

Luiten *et al.* (2020) fizeram um estudo amplo acerca da avaliação de métodos *MOT* e propuseram a métrica *Higher Order Tracking Accuracy* (*HOTA*), a qual resume os três principais aspectos de um sistema de rastreio (detecção, localização e associação) em um único valor. Com o intuito de que a métrica *HOTA* fosse expressiva tanto para usuários finais dos sistemas de rastreio quanto para os desenvolvedores, Luiten *et al.* (2020) projetaram-na de modo a ser decomposta em 5 submétricas que avaliam os diferentes tipos de erro presentes no rastreio de múltiplos objetos, os quais são expressos pela revocação da detecção, acurácia da detecção, revocação da associação, precisão da associação e acurácia de localização. Desse modo é possível ter uma única métrica que permite facilmente ranquear métodos de rastreio, quanto submétricas que dão indícios de como o sistema de rastreio lida com os diferentes tipos de erros. Ademais, Luiten *et al.* (2020) conduziram experimentos para avaliar como as métricas *HOTA*, *MOTA* (BERNARDIN; STIEFELHAGEN, 2007) e *IDF1* (RISTANI *et al.*, 2016) refletem a percepção humana acerca da qualidade do resultado do rastreio, encontrando resultados que indicam que a métrica *HOTA* é mais condizente que as outras duas.

Para o cálculo da métrica *HOTA*, primeiro são calculados os valores de $HOTA_\alpha$, sendo α o limiar de similaridade que valida o pareamento de uma detecção predita e uma

detecção verdadeira (*ground truth detection*), de modo que este par é apenas permitido se o valor de similaridade entre as detecções for maior ou igual a α. Um exemplo de métrica de similaridade aplicável é a *Intersection over Union* (*IoU*), que quantifica a sobreposição entre as detecções. Dessa forma, dada uma função de similaridade cuja imagem é o intervalo [0, 1], $HOTA_\alpha$ é calculado para valores de α iniciando em 0,05 e indo até 0,95 em passos de 0,05. O valor final da métrica *HOTA* é a média simples dos valores de $HOTA_\alpha$, como mostra a Equação 9 abaixo. Esta média tem o intuito de contabilizar a acurácia de localização do sistema de rastreio, visto que a métrica *HOTA* é calculada usando uma ampla gama de valores α, onde valores pequenos são menos restritivos quanto a similaridade (ou proximidade) das detecções na hora do pareamento e valores altos de α apenas permitem pareamento entre detecções que são muito semelhantes.

$$HOTA = \frac{1}{19} \sum_{\alpha \in \{0{,}05;\ 0{,}1;\ \ldots;\ 0{,}95\}} HOTA_\alpha \tag{9}$$

Por sua vez, para o cálculo de $HOTA_\alpha$, primeiramente é realizado um pareamento entre detecções preditas e detecções verdadeiras para cada frame de modo que o valor final de $HOTA_\alpha$ seja maximizado. Os pares formados são considerados como verdadeiros positivos (*TPs*), as detecções preditas que não foram pareadas com nenhuma detecção verdadeira são consideradas falsos positivos (*FPs*) e as detecções verdadeiras que não foram pareadas a nenhuma detecção predita são consideradas falsos negativos (*FNs*). As contagens *TPs*, *FPs* e *FNs* estão relacionadas a acurácia de detecção do sistema de rastreio, já para mensurar a sua acurácia de associação (capacidade de manter as identidades dos objetos ao longo de suas trajetórias), Luiten *et al.* (2020) propuseram as contagens *True Positive Associations* (*TPAs*), *False Positive Associations* (*FPAs*) e *False Negative Associations* (*FNAs*). Para um dado *TP*, nomeado de *c*, e sabendo que cada detecção, seja predita ou verdadeira, possui um indicador de identidade (valor numérico que representa sua identidade, comumente chamado de *ID*), o conjunto de associações verdadeiros positivos $TPA(c)$ corresponde ao conjunto de *TPs* onde as detecções preditas possuem o mesmo *ID* que a detecção predita de *c* e as detecções verdadeiras possuem o mesmo *ID* que a detecção verdadeira de *c*, conforme Equação 10.

$$TPA(c) = \{k\},\ k \in \{TP \mid prID(k) = prID(c)\ \&\ gtID(k) = gtID(c)\} \tag{10}$$

Já o conjunto de associações falsos positivos $FPA(c)$ são os *TPs* que possuem detecções verdadeiras com *ID* diferente da detecção verdadeira de *c*, mas que possuem detecções preditas com *ID* igual a detecção predita de *c*, juntamente com os *FPs* que possuem o mesmo *ID* que a detecção predita de *c*, conforme Equação 11.

$$\begin{aligned} &FPA(c) = \{k\} \\ &k \in \{TP \mid prID(k) = prID(c)\ \&\ gtID(k) \neq gtID(c)\} \\ &k \in \{FP \mid prID(k) = prID(c)\} \end{aligned} \tag{11}$$

Por fim, o conjunto de associações falsos negativos $FNA(c)$ são os *TPs* que possuem detecções verdadeiras com *ID* igual a detecção verdadeira de *c*, mas que possuem detecções preditas com *ID* diferente da detecção predita de *c*, juntamente com os *FNs* que possuem o mesmo *ID* que a detecção verdadeira de *c*, conforme Equação 12.

$$\begin{aligned} &FNA(c) = \{k\} \\ &k \in \{TP \mid prID(k) \neq prID(c)\ \&\ gtID(k) = gtID(c)\} \\ &k \in \{FP \mid gtID(k) = gtID(c)\} \end{aligned} \tag{12}$$

Dessa forma, o valor de $HOTA_\alpha$ é calculado usando a Equação 13 abaixo.

$$HOTA_\alpha = \sqrt{\frac{\sum_{c \in \{TP\}} \frac{|TPA(c)|}{|TPA(c)| + |FPA(c)| + |FNA(c)|}}{|TP| + |FP| + |FN|}} \tag{13}$$

Luiten *et al.* (2020) comentam que a "métrica *HOTA* mede o quão bem as trajetórias das detecções associadas se alinham e calcula a média disto para todas as detecções associadas enquanto também penaliza detecções sem associação".

2.2.2.4 MOTChallenge benchmark

O *MOTChallenge benchmark* foi desenvolvido com o objetivo de "pavimentar um caminho em direção a uma estrutura de avaliação unificada para quantificações mais significativas do rastreio de múltiplos alvos" (LEAL-TAIXÉ *et al.*, 2015). Leal-Taixé *et al.* (2015) expõem:

> Avaliar e comparar métodos de rastreio de múltiplos alvos não é trivial por inúmeras razões [...]. Primeiro, diferente de outras tarefas, como eliminação de ruído de imagens, o resultado base, isto é, a solução perfeita que se almeja, é difícil de ser definido claramente. Alvos parcialmente visíveis, oclusos ou recortados, reflexos em espelhos ou janelas e objetos que se assemelham grandemente a alvos, todos impõe ambiguidades intrínsecas, de modo que até humanos podem não alcançar consenso quanto a uma solução ideal. Segundo, a existência de diferentes métricas de avaliação com parâmetros não fixos e definições ambíguas frequentemente levam a resultados quantitativos inconsistentes. Por fim, a falta de dados de treino e teste pré-definidos dificulta uma justa comparação entre diferentes métodos. (LEAL-TAIXÉ *et al*., 2015, p. 01, tradução nossa).

Ainda segundo Leal-Taixé *et al.* (2015), o *MOTChallenge benchmark* é composto por três componentes principais: um *dataset* disponíveis publicamente, um método de avaliação centralizado e uma infraestrutura para *crowdsourcing* de novos dados, métodos de avaliação e formas de anotação. Desde o seu surgimento em 2015, o *MOTChallenge* recebe melhorias e ampliações, como a adição de novos conjuntos de dados (DENDORFER *et al.*, 2020), novas métricas de avaliação (LUITEN *et al.*, 2020) e realização de desafios abertos à comunidade.

2.3 Estudos correlatos

Sayed e Saunier (2007) propuseram um método automatizado de detecção de conflitos veiculares a partir de vídeos produzidos por câmeras estacionárias com ênfase em interseções. A ferramenta é composta de dois módulos, um para rastreio de veículos e outro para detecção de conflitos a partir dos dados de rastreio obtidos. Foi utilizado uma implementação do rastreador de características Kanade-Lucas-Tomasi (*KLT Feature Tracker*) para a construção do módulo de rastreio (SAUNIER; SAYED, 2007). Ismail *et al.* (2009) expandiram o método para detecção de conflitos entre pedestres e veículos e Hussein *et al.* (2015) expandiram novamente para realização de análises da segurança de pedestres em interseções semaforizadas de Nova Iorque através da identificação automática de conflitos, violações por parte dos pedestres, parâmetros *GAIT* (frequência e tamanho dos passos), entre outros.

Zhang *et al.* (2020) utilizaram uma Rede Neural Recorrente do tipo *Long Short-Term Memory* (*LSTM*) para predição da intenção de travessia de pedestres em uma interseção

semaforizada a partir de variáveis de localização, direção de movimento, gênero e pertencimento à agrupamento de pedestres. Dentro os dados utilizados na predição da intenção de travessia, os de localização dos pedestres foram obtidos a partir do uso de técnicas de visão computacional para detecção e rastreio de objetos, *You Only Look Once* versão 3 (*YOLOv3*) (REDMON; FARHADI, 2018) e *Simple Online and Realtime Tracking with a Deep Association Metric* (*DeepSORT*) (WOJKE; BEWLEY; PAULUS, 2017), respectivamente. Ainda segundo Zhang *et al.* (2020), o sistema de detecção de intenção de travessia poderia ser utilizado na implementação de sistemas de alerta de motoristas quanto a comportamentos de travessia inesperados.

Alver *et al.* (2021) utilizaram técnicas de detecção e rastreio de objetos durante a realização de estudos sobre o comportamento de travessia de pedestres em Izmir, Turquia. Para tal, foram utilizados os modelos de detecção de objetos *YOLOv3* e segmentação de instâncias *YOLACT* para detecção de pedestres, bagagens de mão e veículos a partir de vídeos produzidos por três câmeras estacionárias. Ademais, o algoritmo *Simple Online and Realtime Tracking* foi utilizado para aproximar as trajetórias dos pedestres e veículos a partir das detecções obtidas com o modelo *YOLOv3*. Para realização desses estudos, também foram coletados dados de gênero e de pertencimento à agrupamento de pedestres, sendo que os dados de gênero foram obtidos manualmente a partir de observações visuais dos vídeos gravados e os dados de agrupamento foram estimados a partir do uso de um limiar de proximidade entre os centroides das caixas de detecção de pedestres produzidas pelo modelo de detecção de objetos.

3 MATERIAIS E MÉTODOS

A presente monografia utilizou o algoritmo de rastreio por detecção *StrongSORT* (DU *et al.*, 2023) juntamente com o modelo detecção de objetos *YOLOv7* (WANG *et al.*, 2022) como os principais componentes na construção da ferramenta de extração de trajetórias. A escolha do algoritmo *StrongSORT* se deu em virtude de três motivos principais:

a) devido às vantagens dos algoritmos de rastreio por detecção (*tracking-by-detection*) em comparação aos algoritmos livres de detecção (*detection-free tracking*), visto que os primeiros permitem um tratamento automático do surgimento e desaparecimento de objetos (LUO *et al.*, 2022) e são passíveis de terem seus resultados melhorados a partir da utilização de modelos de detecções mais performáticos (BEWLEY *et al.*, 2016);
b) devido às métricas *MOT* de avaliação alcançadas por este em relação a outras abordagens similares (DU *et al.*, 2023);
c) devido a seu código-fonte escrito na linguagem de programação Python e disponibilizado em acesso livre, o que facilita a modificação do algoritmo para melhor adequá-lo ao problema-alvo desta monografia.

O detector de objetos utilizado na implementação oficial do algoritmo *StrongSORT* é o modelo *YOLOX*, o qual foi escolhido para substituir o modelo *Faster R-CNN* usado no algoritmo predecessor, buscando melhorar o balanceamento entre o tempo de execução e a acurácia do algoritmo (DU *et al.*, 2023). Nesta monografia, por sua vez, optou-se pela utilização do modelo de detecção de objetos *YOLOv7*. Os principais motivos que levaram à esta escolha foram:

a) é um modelo de detecção de estágio único, o que facilita o seu retreinamento para diferentes problemas-alvo (REDMON *et al.*, 2016);
b) é um modelo de detecção de objetos leve e eficiente, considerado como de tempo real, possuindo métricas de qualidade próximas aos modelos do estado da arte e executável em computadores locais com GPU única (WANG *et al.*, 2022);
c) é uma das versões mais recentes dos modelos da família YOLO até a presente data;
d) possui código-fonte escrito na linguagem de programação Python e disponibilizado em acesso livre, o que facilita a realização de modificações para melhor adequá-lo ao problema-alvo desta monografia.

A seguir são detalhados os materiais e métodos que foram utilizados para o alcance de cada objetivo específico e, consequentemente, do objetivo geral desta monografia.

3.1 Retreinamento do modelo de detecção

Uma vez que o algoritmo de rastreio utilizado é da categoria *tracking-by-detection*, a qualidade do detector de objetos é um fator crucial na performance da ferramenta de coleta de trajetórias. Com o fito de ampliar ainda mais a performance do detector de objetos para o problema-alvo, foi realizado o retreinamento (refinamento) de um modelo *YOLOv7* pré-treinado utilizando um *dataset* de detecção de veículos e pedestres construído a partir de imagens extraídas de vídeos de monitoramento do tráfego fornecidos pelo CTAFOR.

3.1.1 Dataset para retreinamento do modelo YOLOv7

Os modelos *YOLOv7* oficiais são pré-treinados com o *dataset MS COCO*, sendo capazes de detectar objetos de 80 classes diferentes (WANG *et al.*, 2022). Dentre estas, as classes "*pessoa*", "*bicicleta*", "*carro*", "*motocicleta*", "*ônibus*" e "*caminhão*" estão presentes e são de especial interesse para o problema-alvo desta monografia. Tomando como base estas classes, o conjunto de dados produzido para retreinamento do modelo *YOLOv7* abrange as classes "*ônibus*", "*carro*", "*ciclista*", "*motociclista*", "*pedestre*", "*picape*", "*caminhão*" e "*van*". Percebe-se que algumas classes foram renomeadas, visando um melhor alinhamento com o objetivo de rastrear os usuários do sistema de transporte e não os veículos. Ademais, a classe "*carro*" foi julgada como sendo mais genérica e abrangente que as outras, optando-se por fragmentá-la nas classes "*carro*", "*picape*" e "*van*", visto as diferenças de aparência entre estes tipos de veículos e a frequência de aparecimento relativamente alta nos vídeos fonte de onde foram extraídas as imagens para formação do *dataset*, permitindo a coleta de uma quantidade significativa de instâncias dessas classes de objetos para retreinamento do modelo.

Para construção do *dataset*, foram extraídas imagens de 5 vídeos de monitoramento fornecidos pelo CTAFOR a uma taxa de 1 frame a cada 2 segundos. Cada vídeo registra imagens do trânsito em uma interseção diferente de Fortaleza-CE a partir de um ângulo alto, totalizando 16 horas de gravações. Os vídeos são do período matutino do dia, variando das 7 até as 12 horas, portanto todos possuem iluminação natural. A Tabela 1 abaixo contém mais informações sobre os vídeos selecionados.

Tabela 1 - Vídeos que compõem o *dataset* de retreinamento.

Interseção	**Dia da gravação**	**Horário da gravação**	**Duração da gravação**
Des. Moreira com Antônio Sales	09/abril/2019	7 às 12h	5h
L. Carneiro com Borges de Melo	08/junho/2021	9 às 11h	2h
Br. Studart com Pontes Vieira	09/abril/2019	7 às 12h	5h
Br. Studart com Abolição	10/junho/2021	9 às 11h	2h
Br. Studart com Antônio Sales	08/junho/2021	9 às 11h	2h

Fonte: autoria própria.

Um problema identificado na utilização desses vídeos de monitoramento para construção do *dataset* de retreinamento é a pouca variabilidade do plano de fundo das imagens, uma vez que são gravações estacionárias. Dessa forma, durante o treinamento, o modelo receberá poucas informações sobre o que não é um objeto, o que pode ocasionar falhas na aplicação em novos cenários, como um alta ocorrência de falsos positivos, onde o modelo confunde o plano de fundo da imagem como um objeto a ser detectado. Para mitigar esta problemática, foram coletadas imagens de ruas e interseções brasileiras usando o Google Imagens, com o intuito de diversificar os planos de fundo do *dataset* de retreinamento e servir de complemento para a aprendizagem do modelo quanto à diferenciação entre objetos e planos de fundo. Os veículos e pedestres presentes nessas imagens foram removidos, deixando-se apenas o plano de fundo das imagens. A remoção dos objetos se deu pela substituição de todos os pixels dentro da caixa delimitadora do objeto por pixels gerados aleatoriamente segundo uma distribuição uniforme. O objetivo dessa remoção foi aliviar a quebra de domínio introduzida pela utilização de imagens que não são do mesmo contexto que as imagens das gravações de monitoramento, buscando evitar maiores instabilidades e divergências de aprendizagem durante o treinamento, visto que nem todas as imagens coletadas do Google Imagens possuem perspectivas semelhantes as imagens das gravações do CTAFOR.

As rotulações foram conduzidas utilizando o aplicativo “labelImg”, disponível no repositório “HumanSignal/labelImg” na plataforma *GitHub*. A escolha desse aplicativo se deu pela sua disponibilidade em acesso gratuito, facilidade de uso e rótulos de saída no formato próprio para treinamento dos modelos da família *YOLO*. Ademais, algumas análises básicas foram realizadas para avaliar o *dataset* construído, como a distribuição da quantidade de objetos

por classe, distribuição das dimensões das caixas delimitadoras por classe, número de imagens por interseção semaforizada e número de objetos por imagem e por classe.

Por fim, o *dataset* foi dividido em dois grupos, um para treino e outro para validação. A divisão das imagens foi feita de forma aleatória e estratificada, de modo que a porção de validação tenha, para cada uma das classes do *dataset*, pelo menos 15% e no máximo 20% das imagens que tenham rótulos daquela classe, buscando alcançar uma distribuição de classes semelhantes em ambos os grupos de imagens. Uma vez que o objetivo primário do retreinamento é aprimorar o modelo de detecção para o problema-alvo desta monografia, não foram selecionadas imagens para compor o grupo de teste do modelo, visto que uma correta metrificação da qualidade final de detecção não é foco deste trabalho. Dessa forma, as métricas e gráficos obtidos com o grupo de validação foram utilizados para tecer breves análises sobre a qualidade do modelo de detecção ao fim do treinamento, conforme detalha a seção 3.1.2 a seguir.

3.1.2 Retreinamento do modelo YOLOv7

Existem diferentes escalas de modelos *YOLOv7*, desde os computacionalmente mais leves até os mais pesados, onde os tempos de treinamento e inferência e as métricas de avaliação crescem conforme a complexidade do modelo aumenta (WANG *et al.*, 2022). Cada escala de modelo requer uma certa quantidade de memória *RAM* livre para sua execução, a depender do número de parâmetros do modelo (ver última coluna da Tabela 2), da dimensão de entrada das imagens (ver segunda coluna da Tabela 2) e da quantidade de imagens sendo processadas simultaneamente (valor conhecido como *batch size*).

Em vista desse quesito de disponibilidade de memória, o modelo *YOLOv7-W6* foi escolhido dentre as escalas acessíveis, conforme detalha a Tabela 2, uma vez que o computador utilizado no retreinamento do modelo possui uma placa de vídeo dedicada *GeForce RTX 3080* com 10 *GB* de memória *RAM* e buscou-se um valor de *batch size* de no mínimo 4. Foi mantido a maioria dos hiperparâmetros padrões do modelo, sendo que os principais parâmetros configurados foram o *batch size*, o número de épocas de treinamento e a dimensões de entrada das imagens. Para o valor do *batch size*, foi escolhido o maior valor possível a partir do valor mínimo quatro. Quanto ao número de épocas de treinamento, foi selecionado um valor arbitrário e analisado se as métricas de validação estabilizaram ao fim do treinamento, retomando-o por mais épocas caso contrário. Para a dimensão de entrada das imagens, manteve-se a dimensão utilizada no pré-treino do modelo *YOLOv7-W6*.

Tabela 2 - Modelos *YOLOv7* pré-treinados.

Modelo	**Dim.**	**AP teste**	**AP 50 teste**	**AP 75 teste**	**FPS**	**Núm. Parâmetros**
YOLOv7	640	51.4%	69.7%	55.9%	161	36.9 M
YOLOv7-X	640	53.1%	71.2%	57.8%	114	71.3 M
YOLOv7-W6	**1280**	**54.9%**	**72.6%**	**60.1%**	**84**	**70.4 M**
YOLOv7-E6	1280	56.0%	73.5%	61.2%	56	97.2 M
YOLOv7-D6	1280	56.6%	74.0%	61.8%	44	154.7 M
YOLOv7-E6E	1280	56.8%	74.4%	62.1%	36	151.7 M

Fonte: Wang *et al.* (2022, p. 07).

Para avaliar a qualidade do modelo de detecção durante e após o refinamento, as seguintes métricas foram calculadas utilizando o grupo de validação do *dataset*:

a) matriz de confusão, para se ter uma ideia geral de como as predições do modelo diferem das rotulações manuais, como identificar possíveis dificuldades na diferenciação entre classes, não detecção de objetos existentes (falsos negativos) e detecção de objetos que não existem (falsos positivos);
b) gráficos de precisão contra revocação por classe, para analisar como o modelo faz o balanceamento entre sua precisão e sua sensibilidade;
c) gráficos de precisão contra confiança e revocação contra confiança por classe, para servir de auxílio na determinação do limiar de confiança do modelo;
d) *mean average precision* (*mAP*) geral e por classe, para servir como métrica geral e auxiliar na escolha da melhor época de treinamento.

Idealmente é necessário a utilização de um terceiro grupo de dados, chamado grupo de teste, para metrificar a qualidade do modelo de detecção treinado. Porém, visto que o modelo de detecção não é o objetivo final desta monografia, bem como foi posteriormente realizada uma etapa de validação da ferramenta de extração de trajetória, optou-se por utilizar novamente os dados de validação para tal. Ressalta-se que os dados de validação não foram utilizados para treinamento direto do modelo, mas sim apenas no acompanhamento e na seleção da melhor época do retreino.

3.2 Rastreio de objetos

A implementação original do algoritmo *StrongSORT* utiliza dados de detecção e aparência (vetores de características) pré-computados, ou seja, o algoritmo de rastreio não incorpora os modelos de detecção de objetos e de extração de características de forma unificada, necessitando rodar esses modelos com antecedência e fornecer os seus resultados como dados de entrada para o algoritmo, o que dificulta a sua utilização. Em vista disso, optou-se pela utilização de uma bifurcação do código original desenvolvida por terceiros, na qual o algoritmo de rastreio é unificado aos modelos de detecção de objetos e de extração de vetores de características, respectivamente os modelos *YOLOv7* (WANG *et al.*, 2022) e *OSNet* (ZHOU *et al.*, 2019), em substituição aos modelos *YOLOX* e *BoT* utilizados na implementação original do algoritmo (DU *et al.*, 2023).

Ademais, algumas modificações foram realizadas no algoritmo de rastreio com o fito de ampliar a qualidade das trajetórias construídas, sem dar grande atenção à manutenção de sua eficiência computacional e habilidade de processamento em tempo real. Dentre estas modificações, as principais foram:

a) limitar a associação de detecções em trajetórias com base na distância euclidiana: a associação entre as novas detecções e os alvos de rastreio é limitada com base em dois limiares, um relacionado a distância de Mahalanobis (custo de movimento) e outro relacionado a discrepância entre os vetores de características (custo de aparência). Como a distância de Mahalanobis não é tão intuitiva quanto a distância euclidiana do ponto de vista espacial (coordenadas da imagem), foi acrescentado dois novos parâmetros configuráveis que limitam a distância, em pixels, e a velocidade máxima, em pixels por frame, entre duas detecções consecutivas em uma trajetória, utilizando-se da distância euclidiana entre os centróides das detecções;
b) possibilidade de substituir a distância de Mahalanobis pela distância *IoU*: foi acrescentado um parâmetro que permite configurar o algoritmo *StrongSORT* para utilizar a distância *IoU* em vez da distância de Mahalanobis na computação da distância entre duas detecções. Apesar de a distância de Mahalanobis levar em consideração a distribuição das detecções que compõem a trajetória, a distância *IoU* é mais simples e pode se tornar mais robusta quando se trabalha com vídeos com alta taxa de quadros por segundo;
c) utilização de um modelo *OSNet* de tamanho maior: o modelo *OSNet* é passível de ser reduzido utilizando multiplicadores que alteram a largura do modelo e a

dimensão de entrada das imagens, permitindo uma troca entre tamanho do modelo, número de operações e performance (ZHOU *et al.*, 2019). O modelo *OSNet* integrado ao algoritmo *StrongSORT* utilizado é uma versão reduzida no modelo base, dessa forma foi realizada a substituição desta versão reduzida pela versão normal, resultando na troca de um modelo com 0,2 milhões de parâmetros por um modelo com 2,2 milhões de parâmetros e melhores métricas de performance (ZHOU *et al.*, 2019).

Por fim, o algoritmo de rastreio possui vários parâmetros configuráveis que servem para calibrá-lo para diferentes situações de rastreio. Alguns destes parâmetros foram julgados como não intuitivos de serem determinados, necessitando a utilização de um método sistemático para seleção de seus valores. Estes parâmetros são:

a) *matching cascade*: diferentemente do algoritmo *DeepSORT*, o *StrongSORT* não utiliza a associação em cascata, resolvendo o problema de otimização de associações detecção-alvo uma vez por frame. Porém, a implementação do algoritmo disponibiliza o parâmetro binário *matching cascade* que permite a ativação do método em cascata;
b) *appearance lambda*: a associação entre novas detecções e alvos de rastreio ativos é resolvida através de um método que busca minimizar um custo de associação composto por duas parcelas, uma referida como custo de movimento e a outra como custo de aparência. O parâmetro *appearance lambda* serve para selecionar o peso que o custo de aparência possui na média ponderada para cálculo do custo final de associação, aceitando valores no intervalo [0, 1];
c) *feature momentum*: no algoritmo *StrongSORT*, cada detecção gerada pelo modelo de detecção de objetos passa por um modelo de extração de vetores de características, gerando um vetor que representa a aparência do objeto. Ademais, cada alvo de rastreio armazena internamente um vetor de características que representa a aparência do objeto sendo rastreado. Quando uma detecção é associada a um alvo, o vetor de características do alvo é atualizado, de modo que o seu novo vetor é uma média ponderada entre o antigo e o vetor de características que acompanha a detecção. O parâmetro *feature momentum* configura o peso que o antigo vetor de características possui nessa média, aceitando valores no intervalo [0, 1];

d) *appearance gate*: parâmetro que configura o limiar de aparência, o qual limita a associação entre detecção e alvo com base no valor da distância cosseno entre o vetor de características da detecção e o vetor de características do alvo. O parâmetro *appearance gate* aceitando valores no intervalo [0, 2], uma vez que esta é a imagem da distância cosseno;
e) *motion only position*: o padrão do algoritmo *StrongSORT* é calcular a distância de Mahalanobis usando todas as coordenadas das caixas delimitadoras das detecções (coordenadas do centróide, a razão entre a largura e altura da caixa e a altura da caixa), ou seja, no espaço quadridimensional. O parâmetro binário *motion only position* serve para configurar o algoritmo de modo a usar apenas as coordenadas do centroide das detecções no cálculo da distância de Mahalanobis;
f) *IoU distance cost*: parâmetro binário para substituir a distância de Mahalanobis pela distância *IoU* no cálculo da matriz de custo de movimento.

Desse modo, foi realizada uma análise em grade para seleção dos valores dos referidos parâmetros. Para cada um, uma série de valores candidatos foi selecionada de forma arbitrária. Uma vez feita a seleção dos valores candidatos, todas as possíveis combinações de valores dos parâmetros foram avaliadas, utilizando como base de avaliação um vídeo de 40 segundos de duração a uma taxa de 10 quadros por segundo, totalizando 401 quadros. O vídeo é um trecho de uma gravação de monitoramento da interseção da avenida Santos Dumont com a avenida Desembargador Moreira, a qual não teve imagens extraídas para o retreinamento do modelo de detecção (vide Tabela 1) e foi utilizada na etapa seguinte de validação da ferramenta de extração de trajetórias, conforme detalha a seção 3.3 desta monografia. O trecho foi selecionado arbitrariamente, porém de modo a abranger todas as classes de usuários alvo.

Os resultados do rastreio para as diferentes combinações de parâmetros foram então ranqueados com base numa média ponderada das métricas *HOTA* e *IDF1*. Para cálculo dessas métricas de rastreio, foram utilizados rótulos previamente definidos, resultantes de anotações manuais, como o padrão de comparação. Esta rotulação também foi conduzida com o auxílio do aplicativo “labelImg”, utilizado previamente na construção do *dataset* de retreinamento. Ademais, as métricas foram computadas com o uso do repositório “JonathonLuiten/TrackEval” disponibilizado na plataforma *GitHub*. A combinação que obteve a maior média foi então escolhida como a configuração final dos parâmetros. Durante os esforços de calibração, a velocidade de processamento da ferramenta também foi avaliada, de modo a se obter uma

estimativa da sua velocidade média de processamento levando em conta as diferentes combinações de parâmetros experimentadas.

3.3 Validação da ferramenta de extração de trajetórias

As métricas de avaliação de algoritmos *MOT* mais comumente usadas, como *HOTA*, *IDF1* e *MOTA*, buscam avaliar os algoritmos de um ponto de vista mais genérico, de modo a possibilitar o ranqueamento de diferentes algoritmos de rastreio usando *datasets* de *benchmarks* e/ou fornecer informações para os desenvolvedores avaliarem os diferentes aspectos de seus algoritmos. No entanto, quando se trata de questões de validação de aplicações de rastreio em contextos específicos, como a tarefa de rastreio de usuários do sistema de transporte da presente monografia, essas métricas revelam-se pouco intuitivas e fornecem poucas informações sobre a capacidade de resolução dos problemas-alvo.

Em vista disso, a qualidade da ferramenta de extração de trajetórias construída foi validada a partir de um ponto de vista mais prático. Foram desenvolvidas rotinas automatizadas que recebem como dados de entrada as trajetórias coletadas pela ferramenta e informações de geometria da região avaliada e retornam os seguintes resultados:

a) contagem classificatória de veículos;
b) contagem classificatória de conversões;
c) instantes de entrada e de saída dos veículos nas faixas de pedestres;
d) contagem de travessias nas faixas de pedestres;
e) atraso e tempo de travessia dos pedestres.

Para validação da ferramenta, foi utilizado um vídeo de monitoramento da interseção da avenida Santos Dumont com a avenida Desembargador Moreira do dia 09 de junho de 2021, onde a pista sentido Praça Portugal da avenida Desembargador Moreira e a avenida Santos Dumont foram analisadas. O vídeo tem 60 minutos de duração, registrando o trânsito das 10 às 11 horas da manhã a uma taxa de 10 quadros por segundo. A gravação foi escolhida de forma arbitrária, mas com intenção de abranger todas as 8 classes de objetos do modelo de detecção e possuir um fluxo veicular e de pedestres considerável.

As variáveis listadas acima foram coletadas manualmente a partir de avaliações visuais do vídeo. A coleta manual foi realizada com o auxílio do aplicativo *Road User Behaviour Analysis* (*RUBA*) (AGERHOLM *et al.*, 2017). Os valores coletados manualmente

representam os valores esperados, os quais serviram como referência para análise dos valores retornados pelas rotinas automatizadas.

As contagens de passagens veiculares e de travessias também auxiliaram na construção de associações binárias entre os “usuários reais”, resultantes da coleta manual, e os “usuários preditos” pela ferramenta de extração de trajetórias. Uma vez construída essas associações, foi analisada a distribuição dos erros para os instantes de entrada e de saída dos veículos nas faixas de pedestres e para os tempos de atraso e de travessia dos pedestres. Quanto às trajetórias sem correspondências (falsos negativos e falsos positivos), foram feitas análises visuais qualitativas acerca dos erros e das falhas cometidas pela ferramenta de extração de trajetórias. Ademais, foram comparados os tempos médios que os veículos levam para percorrer a faixa de pedestres, o atraso médio de travessia e a duração média da travessia obtidos pela coleta manual e pela coleta automatizada.

4 RESULTADOS E DISCUSSÃO

4.1 Retreinamento do modelo de detecção

4.1.1 Dataset para retreinamento

Das imagens extraídas dos vídeos listados na Tabela 1, apenas algumas foram rotuladas. A quantidade de imagens rotuladas por vídeo está informada na Tabela 3 abaixo, enquanto, na Figura 5, estão ilustradas imagens exemplos de cada um desses vídeos.

Tabela 3 - Número de imagens rotuladas por interseção.

Interseção	**Número de imagens rotuladas**
Des. Moreira com Antônio Sales	325
L. Carneiro com Borges de Melo	370
Br. Studart com Pontes Vieira	355
Br. Studart com Abolição	410
Br. Studart com Antônio Sales	315
Total	1775

Fonte: autoria própria.

Figura 5 - Vídeos utilizados na construção do *dataset* de retreinamento.

Des. Moreira com Antônio Sales

L. Carneiro com Borges de Melo

Br. Studart com Pontes Vieira

Br. Studart com Abolição

Br. Studart com Antônio Sales

Fonte: CTAFOR.

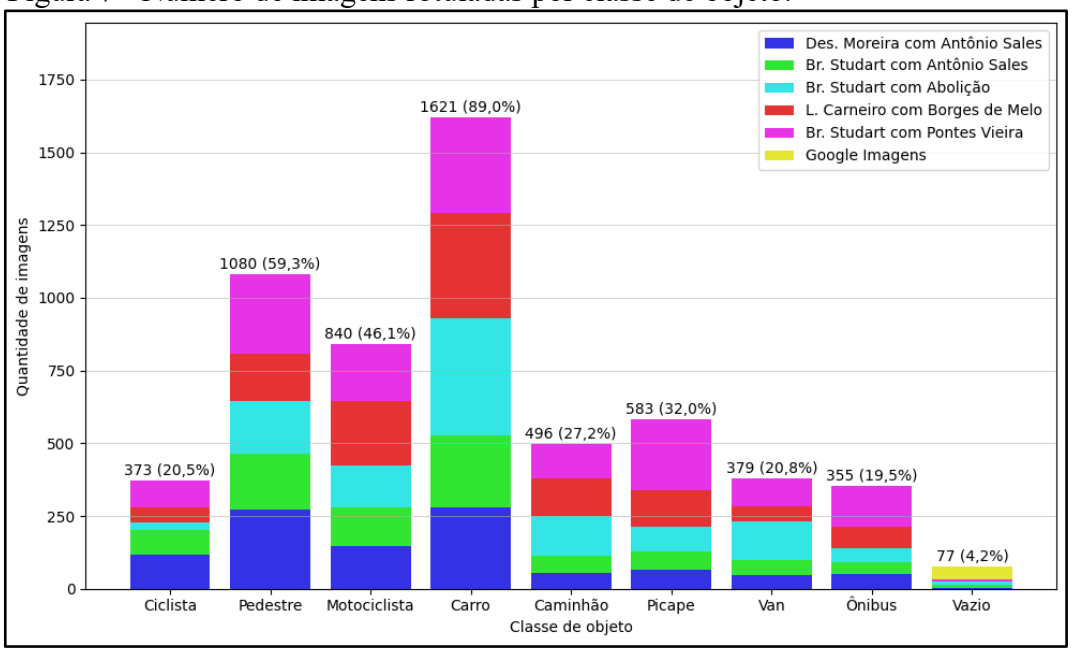
Quanto às imagens amostradas pelo Google Imagens, foram selecionadas um total de 46, onde todas tiveram os veículos e pedestres removidos. Na Figura 6 abaixo está representada uma dessas imagens, antes e depois da remoção de objetos.

Figura 6 - Imagem coletada pelo Google Imagens.

Fontes: Autoescola Online (2018), fotografia digital à esquerda; Autoria própria, fotografia digital à direita.

Dessa forma, um total de 1821 imagens compõem o *dataset* para retreinamento do modelo de detecção. A Figura 7 ilustra a quantidade de imagens rotuladas por classe de objeto, onde as porcentagens são em referência a quantidade total de imagens rotuladas. A classe "vazia" refere-se às imagens sem incidência de veículos ou pedestres, ou seja, não possuem objetos rotulados. Nota-se que todas as imagens amostradas do Google Imagens pertencem a esta "classe", uma vez que apenas o plano de fundo foi mantido.

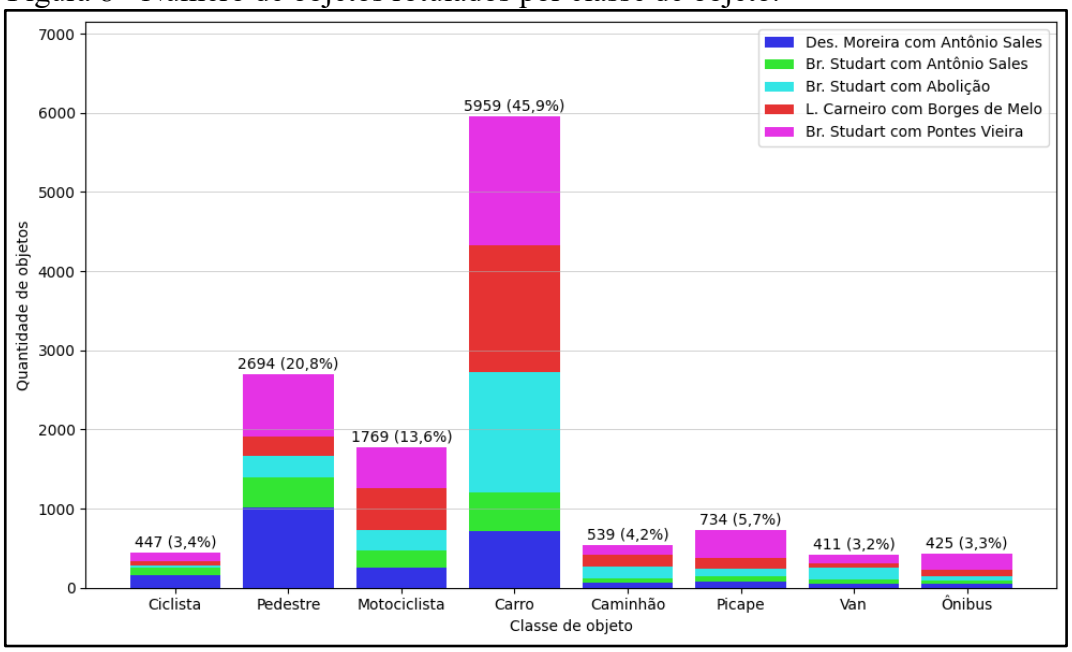
Figura 7 - Número de imagens rotuladas por classe de objeto.



Fonte: autoria própria.

É perceptível que as classes "ciclista", "ônibus", "picape", "van" e "caminhão" estão presentes em menos imagens que as demais, uma vez que essas classes, de modo intuitivo, são menos frequentes em relação aos usuários mais comuns (pedestres, carros de passeio e motociclistas). Passando para uma análise da quantidade de objetos rotulados, conforme ilustra a Figura 8, este desbalanceamento se acentua. As porcentagens exibidas são referentes ao número total de objetos rotulados nas 1821 imagens.

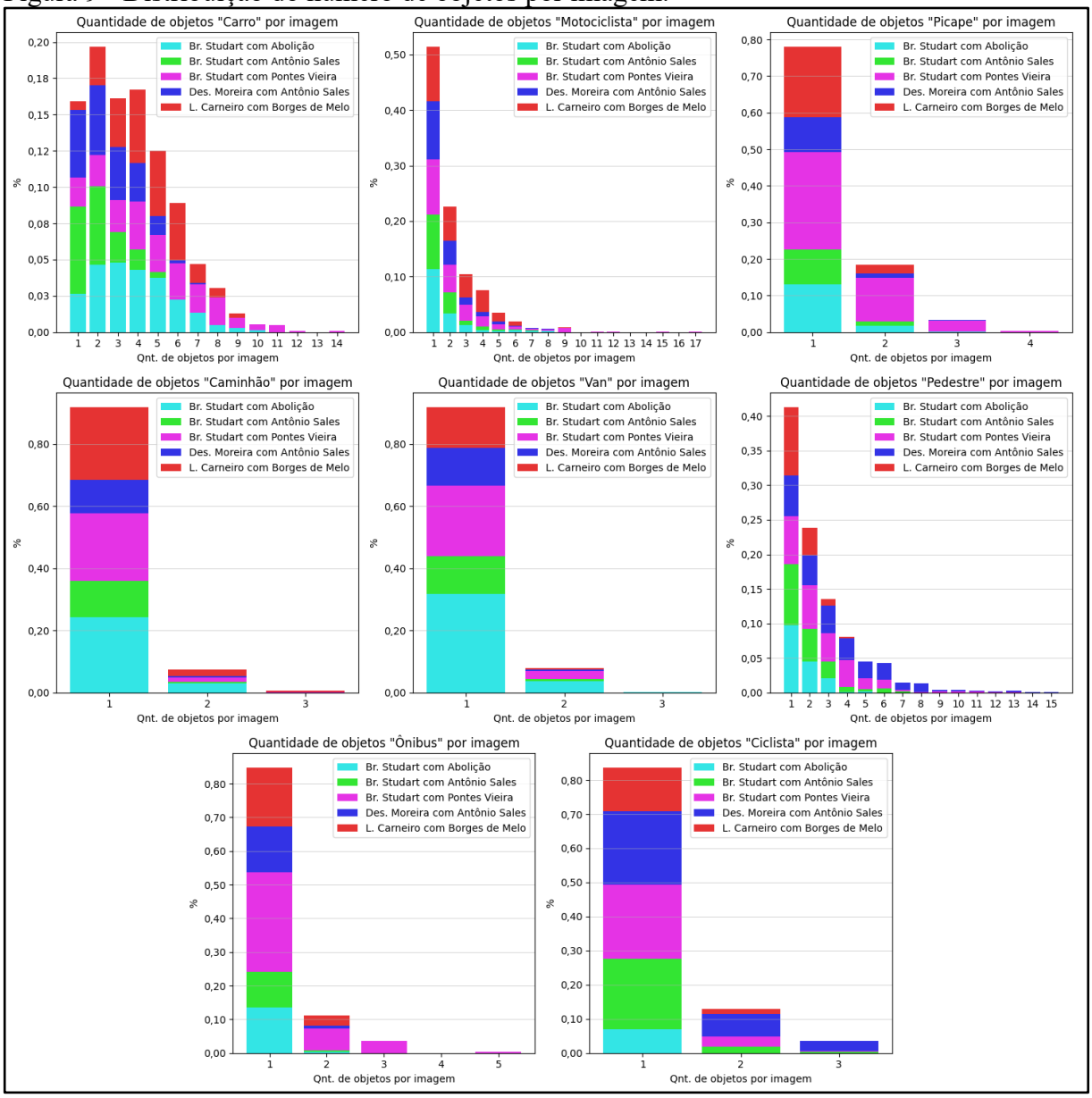
Figura 8 - Número de objetos rotulados por classe de objeto.



Fonte: autoria própria.

O aumento do desbalanceamento entre as classes é justificado pela quantidade elevada de objetos “pedestre”, “motociclista” e “carro” por imagem anotada, uma vez que, por exemplo, é mais comum a presença de 10 carros de passeio em uma única imagem do que a mesma quantidade de caminhões em uma única imagem. Os gráficos da Figura 9 a seguir trazem a distribuição do número de objetos por imagem para cada classe de usuário. Percebe-se que as classes citadas acima chegam a superar a marca de 13 objetos por imagem, enquanto as demais estão limitadas em 5 objetos por imagem ou menos.

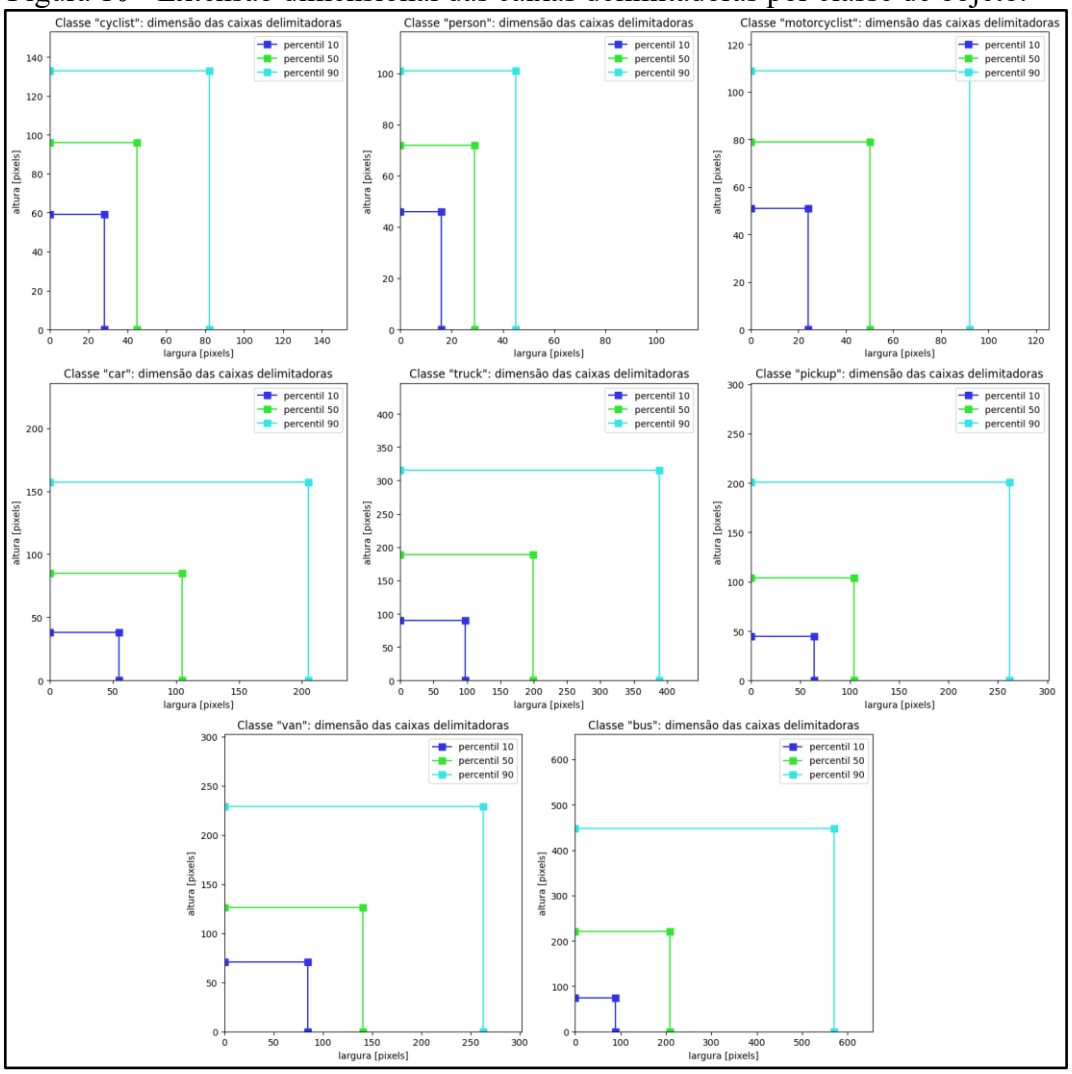
Figura 9 - Distribuição do número de objetos por imagem.



Fonte: autoria própria.

Ainda quanto da caracterização do *dataset* para retreinamento, a Figura 10 abaixo traz uma representação da extensão dimensional dos rótulos presentes no *dataset,* mostrando a largura e altura das caixas delimitadoras dos objetos para os percentis 10, 50 e 90.

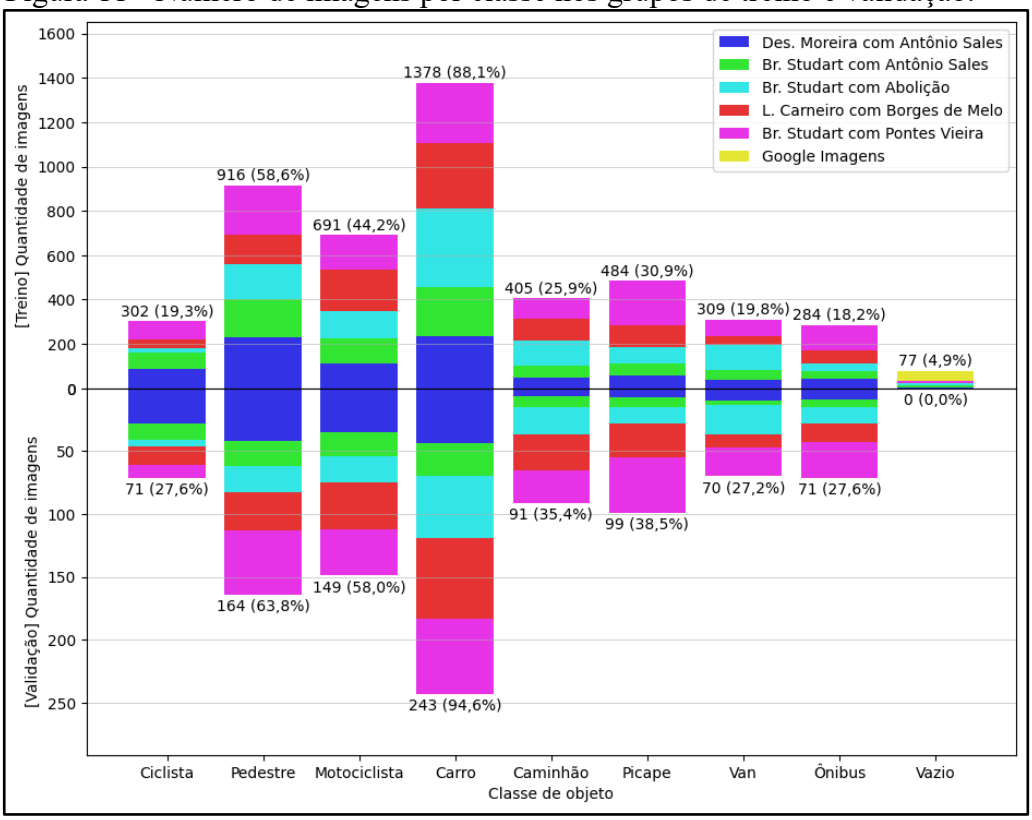
Figura 10 - Extensão dimensional das caixas delimitadoras por classe de objeto.



Fonte: autoria própria.

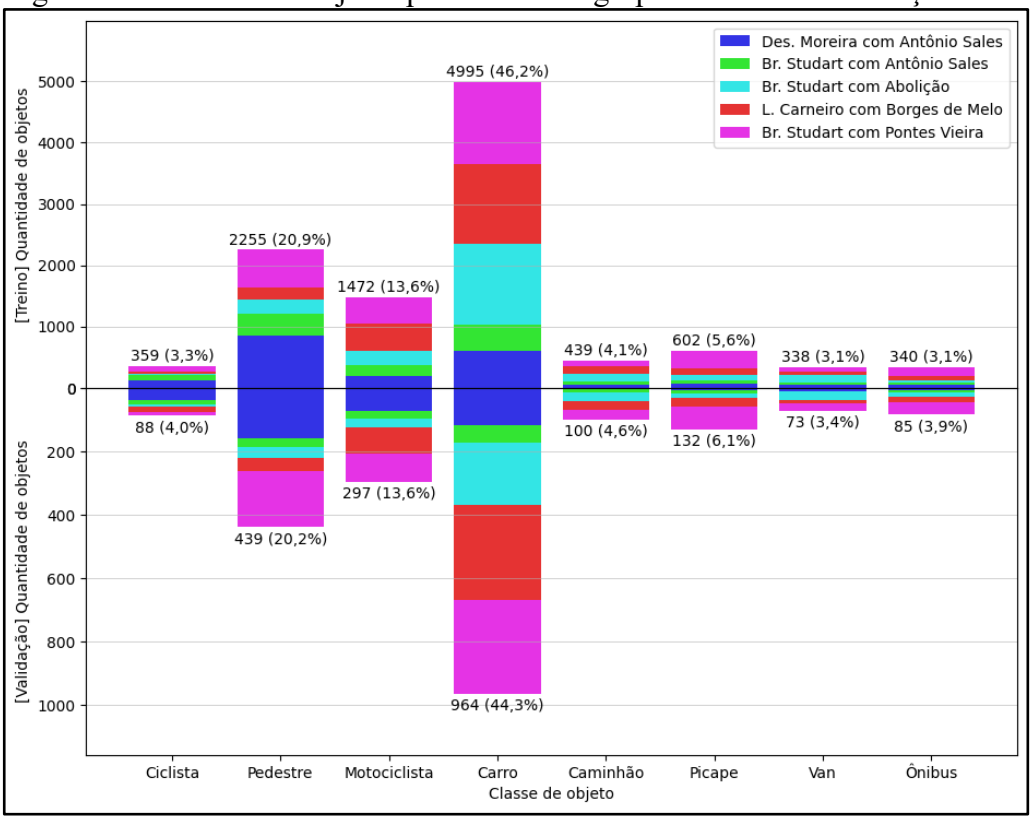
Finalizada a análise do *dataset* como um todo, foi realizada a separação deste entre os grupos de treino e validação. A separação foi de modo aleatório estratificado, buscando manter uma distribuição de objetos semelhantes em ambos os grupos, com o grupo de validação recebendo até 20% das imagens com rótulos de cada classe. Ademais, todas as imagens "vazias", ou seja, com nenhum objeto rotulado, foram redirecionadas para o grupo de treino. A quantidade de imagens no grupo de treinamento e validação são de 1564 e 257, respectivamente. A seguir, nas figuras 11 e 12, são ilustradas as distribuições do número de imagens e de objetos, respectivamente, para os grupos de treinamento e validação.

Figura 11 - Número de imagens por classe nos grupos de treino e validação.



Fonte: autoria própria.

Figura 12 - Número de objetos por classe nos grupos de treino e validação.



Fonte: autoria própria.

4.1.2 Retreinamento do modelo YOLOv7

O repositório oficial dos modelos *YOLOv7* possui pouca manutenção de código, o que dificulta a sua utilização, visto a presença de falhas que foram encontradas, mas que não foram corrigidas. Em vista disso, foi feita uma bifurcação (ou *fork*, termo mais conhecido para este procedimento) do repositório, onde foram implementadas algumas correções de erros essenciais para a conclusão do retreinamento do modelo pré-treinado *YOLOv7-W6*. Esta bifurcação está disponível no repositório "nelioasousa/yolov7" na plataforma *GitHub*.

O processo de treinamento dos modelos *YOLOv7* pode ser configurado a partir de uma série de parâmetros, todos possuindo um valor padrão. Destes parâmetros, apenas alguns foram alterados de modo a diferir dos valores padrões. A Tabela 4 abaixo detalha as modificações mais importantes.

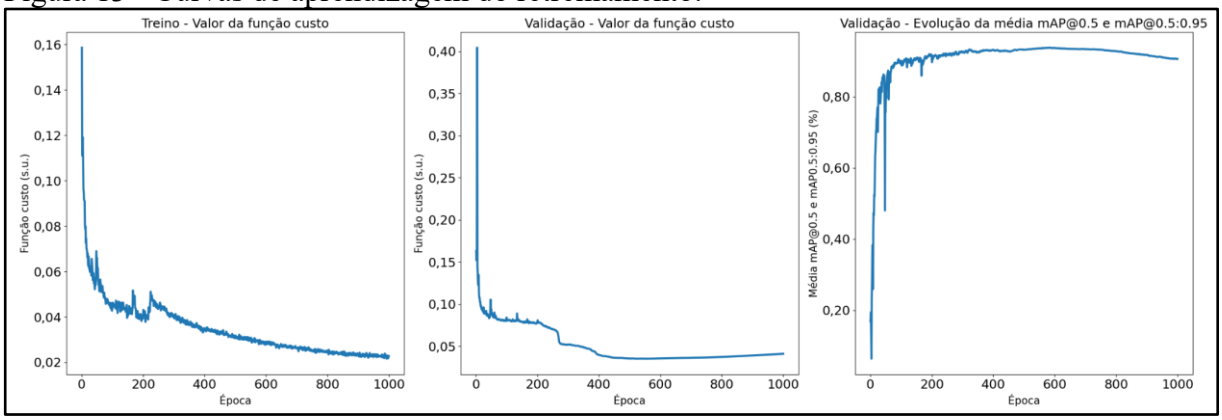
Tabela 4 - Parâmetros de treinamento modificados.

Parâmetro	**Descrição**	**Valor padrão**	**Valor utilizado**
weights	Arquitetura do modelo e valores iniciais de peso.	yolov7.pt	yolov7-w6_training.pt
hyp	Arquivo que determina os valores dos hiperparâmetros de treinamento.	data/hyp.scratch.p5.yaml	data/hyp.scratch.p6.yaml
epochs	Número de épocas de treino.	300	1000
batch-size	Número de imagens utilizadas para realizar um "passo" de treino (quando os pesos do modelo são atualizados).	16	4
workers	Número de processos responsáveis pelo carregamento de imagens na memória para treinamento.	8	4
img-size	Referente a dimensão de entrada das imagens, representando o valor da maior dimensão em pixels.	640	1280

Fonte: autoria própria.

Finalizado o retreinamento, a Figura 13 a seguir ilustra a evolução do modelo de detecção ao longo das épocas de treino, onde os dois primeiros gráficos mostram a evolução do valor da função custo para os dados de treino e validação, respectivamente, enquanto o último gráfico exibe a evolução da média simples entre as métricas *mAP*@0.5 e *mAP*@0.5:0.95 em referência aos dados de validação.

Figura 13 - Curvas de aprendizagem do retreinamento.

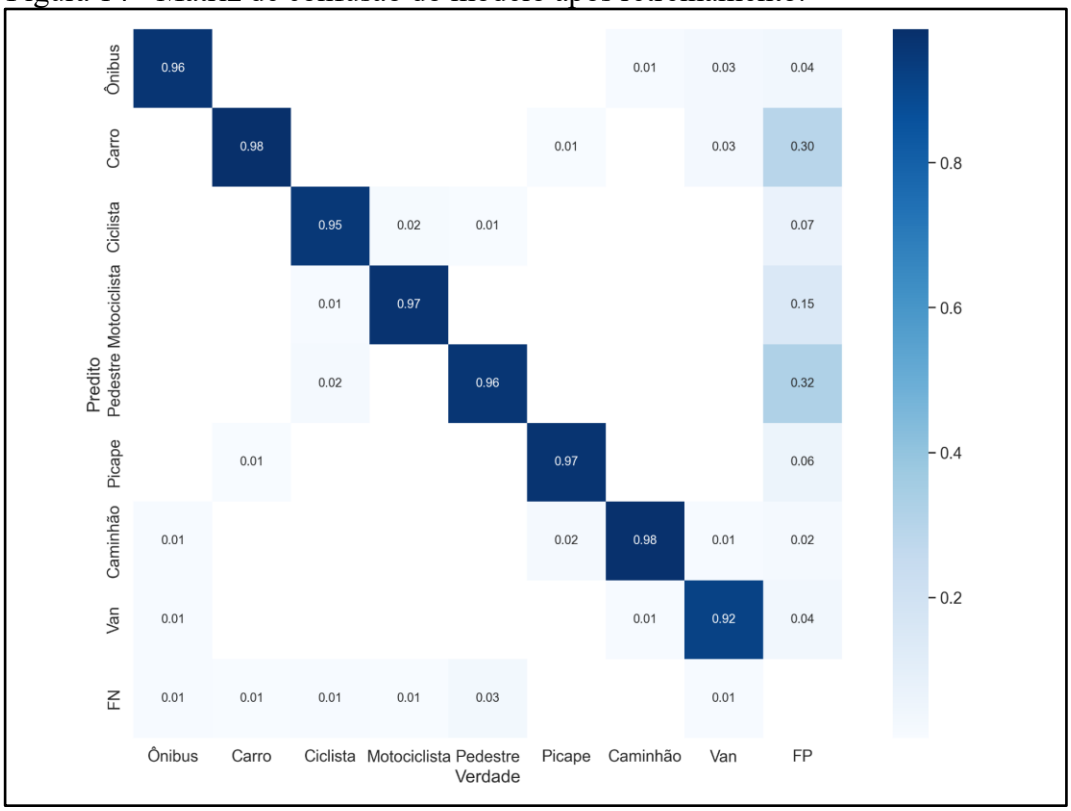


Fonte: autoria própria.

Apesar da curva da função custo de treinamento (primeiro gráfico da Figura 13) indicar que o modelo ainda pode melhor se ajustar aos dados de treino, os gráficos de validação já indicam uma possível ocorrência de sobreajuste do modelo, visto que, por volta da 600ª época, o valor da função custo para os dados de validação começou a aumentar e a média das métricas *mAP* começou a diminuir suavemente. Tendo como base estes indícios, optou-se por não dar continuidade ao retreinamento do modelo. Para seleção da melhor época de treinamento, utilizou-se a média simples das métricas *mAP*@0.5 e *mAP*@0.5:0.95, resultando que os pesos obtidos na época 584 possuem o melhor desempenho nos dados de validação em comparação às demais épocas. Desse modo, esses pesos foram selecionados como o modelo de detecção final utilizado na construção da ferramenta de extração de trajetórias.

Uma vez determinado o melhor conjunto de pesos para o modelo de detecção, foram realizadas avaliações da capacidade de detecção utilizando o conjunto de dados de validação. Abaixo, na Figura 14, segue a matriz de confusão do modelo de detecção após o retreinamento.

Figura 14 - Matriz de confusão do modelo após retreinamento.



Fonte: autoria própria.

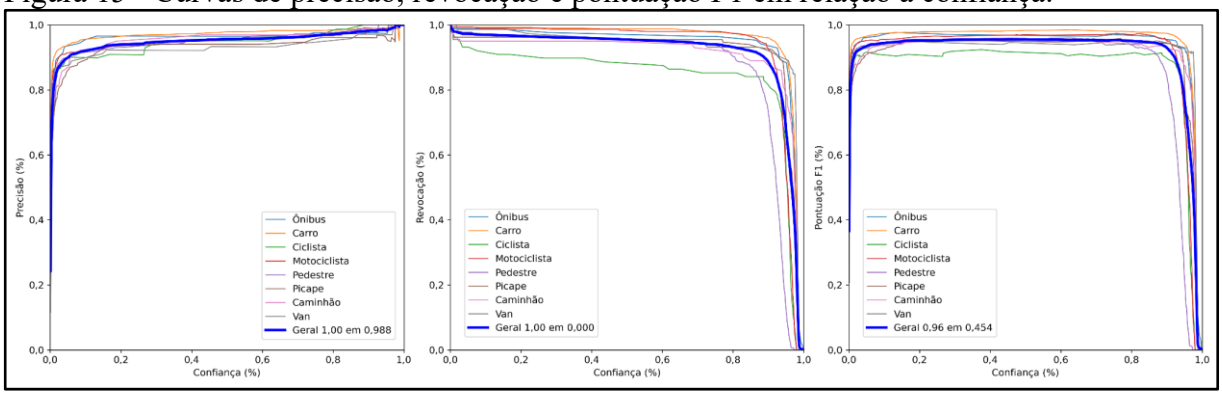
A matriz possui uma diagonal principal forte, indicando uma boa precisão e revocação. Porém, nota-se que a última coluna, referente aos falsos positivos, se destaca dentre os erros cometidos pelo modelo, indicando uma pequena fraqueza deste em distinguir alguns objetos do plano de fundo das imagens, com foco nas classes “carro”, “motociclista” e “pedestre”. Ademais, o modelo, de forma esporádica, confunde a classe “ônibus” com as classes “van” e “caminhão”, confunde a classe “carro” com a classe “picape”, confunde a classe “ciclista” com as classes “pedestre” e “motociclista”, confunde a classe “motociclista” com a classe “ciclista”, confunde a classe “pedestre” com a classe “ciclista”, confunde a classe “picape” com as classes “caminhão” e “carro”, confunde a classe “caminhão” com as classes “van” e “ônibus” e, por fim, confunde a classe “van” com as classes “caminhão”, “carro” e “ônibus”.

Quanto à ênfase das falhas do tipo falso positivo, uma hipótese de solução seria aumentar o número de imagens sem incidência de nenhum objeto e diversificar o plano de fundo das imagens, usando gravações de outras interseções ou das mesmas interseções, mas com diferenças de perspectiva. Já em relação às confusões entre classes, uma hipótese de solução seria aumentar o número de dados rotulados e buscar um melhor balanceamento do número de rótulos por classe. Porém, algumas dessas confusões são de certa forma compreensíveis, visto

a semelhança entre algumas classes. Um ponto positivo é o fato de o algoritmo de rastreio ser independente da classe do objeto, adicionando robustez contra o problema de troca de classe na construção das trajetórias.

Na Figura 15 a seguir estão representados, respectivamente, os gráficos de precisão, revocação e pontuação F1 (*F1-score*) em função do valor de limiar de confiança das detecções.

Figura 15 - Curvas de precisão, revocação e pontuação F1 em relação a confiança.

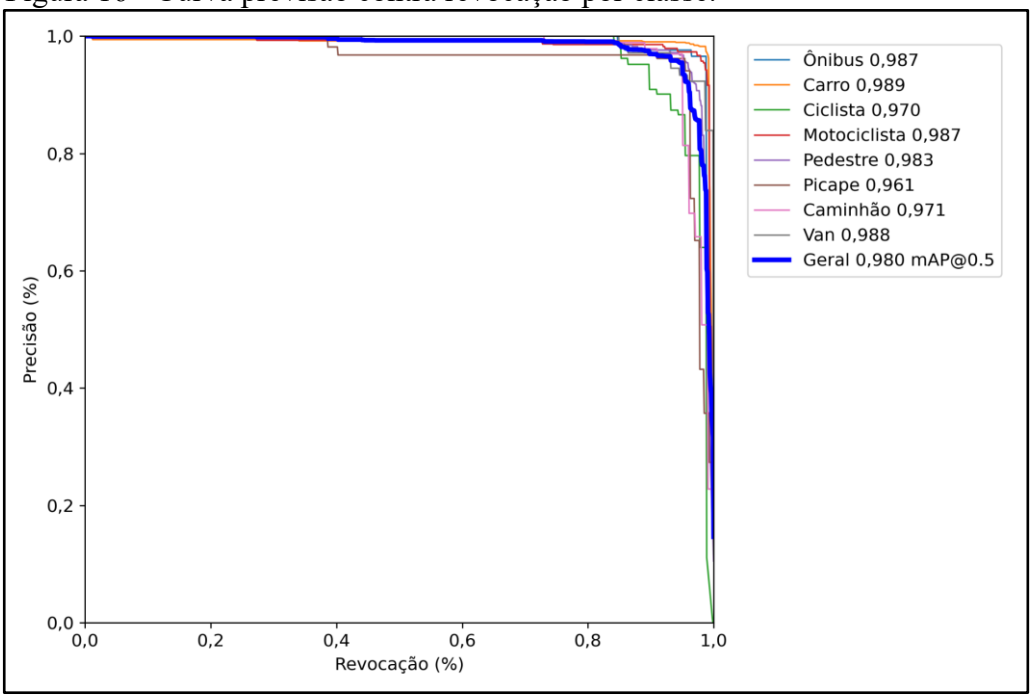


Fonte: autoria própria.

Tais curvas são uma boa referência para seleção do limiar de confiança do modelo, de modo que detecções preditas serão realmente consideradas como objetos se o seu valor de confiança for superior ao valor do limiar, ou seja, apenas quanto o modelo "estiver minimamente confiante" de que a sua detecção realmente é um objeto. Uma vez que o algoritmo de rastreio possui a capacidade de reidentificação de objetos, foi dada preferência a precisão em relação a revocação do modelo, selecionando 75% como o limiar de confiança. Segundo o gráfico da pontuação F1, que é a média harmônica da precisão e da revocação, adotar 45,4% como limiar de confiança resultaria no maior valor de pontuação F1 (96%) com base nos dados de validação, porém percebe-se que para um limiar de 75% esta pontuação não sofre grande alteração, de modo que este valor de limiar também faz um bom balanceamento entre precisão e revocação. Ademais, o valor relativamente alto do limiar de confiança pode fornecer uma certa capacidade de filtragem das detecções falso positivas.

Por fim, na Figura 16 abaixo, está representado o gráfico de precisão contra a revocação, tanto por classe quanto geral, bem como as métricas *AP* por classe e *mAP@0.5* geral. Apesar dos problemas apontados pela matriz de confusão, os valores das métricas *AP* são muito promissores para todas as classes, alcançando um *mAP* de 98,0%.

Figura 16 - Curva previsão contra revocação por classe.



Fonte: autoria própria.

4.2 Rastreio de objetos

A versão final do algoritmo de rastreio utilizado nesta monografia está disponível no repositório "nelioasousa/strongsort-yolo" na plataforma *GitHub*. Este repositório também contém a rotina automática para calibração do algoritmo. Quanto ao esforço de calibração do algoritmo de rastreio, os valores candidatos para cada parâmetro analisado estão representados na Tabela 5 abaixo.

Tabela 5 - Valores candidatos para calibragem do algoritmo de rastreio.

Parâmetro	**Valores candidatos**
Matching cascade	[Não, Sim]
Appearance lambda	[0,2; 0,5; 0,7; 0,9; 0,98]
Feature momentum	[0; 0,25; 0,5; 0,75; 0,95]
Appearance gate	[0,15; 0,175; 0,2; 0,225; 0,25; 0,275; 0,3; 0,325; 0,35]
Motion only position	[Não, Sim]
IoU distance cost	[Não, Sim]

Fonte: autoria própria.

Desse modo, um total de 1800 combinações foram comparadas entre si. Para ranquear estas combinações e selecionar a melhor avaliada, foi utilizado uma média ponderada entre as métricas *HOTA* e *IDF1*, com importância de 30% e 70%, respectivamente. Esta escolha de pesos foi devido a preferência pela manutenção das identidades ao longo das trajetórias dos objetos, característica representada pelo valor da métrica *IDF1*, em detrimento da métrica *HOTA* que é relativamente mais complexa e abrange de forma monotônica todos os diferentes tipos de erros de rastreio (LUITEN *et al.*, 2020). Abaixo, na Tabela 6, estão representados os principais parâmetros da melhor combinação, onde os parâmetros em destaque foram os parâmetros calibrados.

Tabela 6 - Valores dos parâmetros após calibragem do algoritmo de rastreio.

Parâmetro	**Valor**
matching-cascade	Não
appearance-lambda	0,98
feature-momentum	0,95
appearance-gate	0,35
motion-only-position	Sim
iou-distance-cost	Sim
conf-thre	0,75
init-period	3
iou-thres	0,6
agnostic-nms	Sim
augment	Sim
iou-gate	0,99
feature-bank-size	60
max-centroid-distance	200
max-age	60

Fonte: autoria própria.

As métricas obtidas por esta combinação estão presentes na Tabela 7 abaixo.

Tabela 7 - Métricas de avaliação do algoritmo de rastreio após calibragem.

Métrica	**Valor**
HOTA	13,25%
LocA	61,04%
$HOTA_{\alpha=0,05}$	80,98%
$LocA_{\alpha=0,05}$	16,73%
IDF1	49,78%
IDP	50,68%
IDR	48,91%

Fonte: autoria própria.

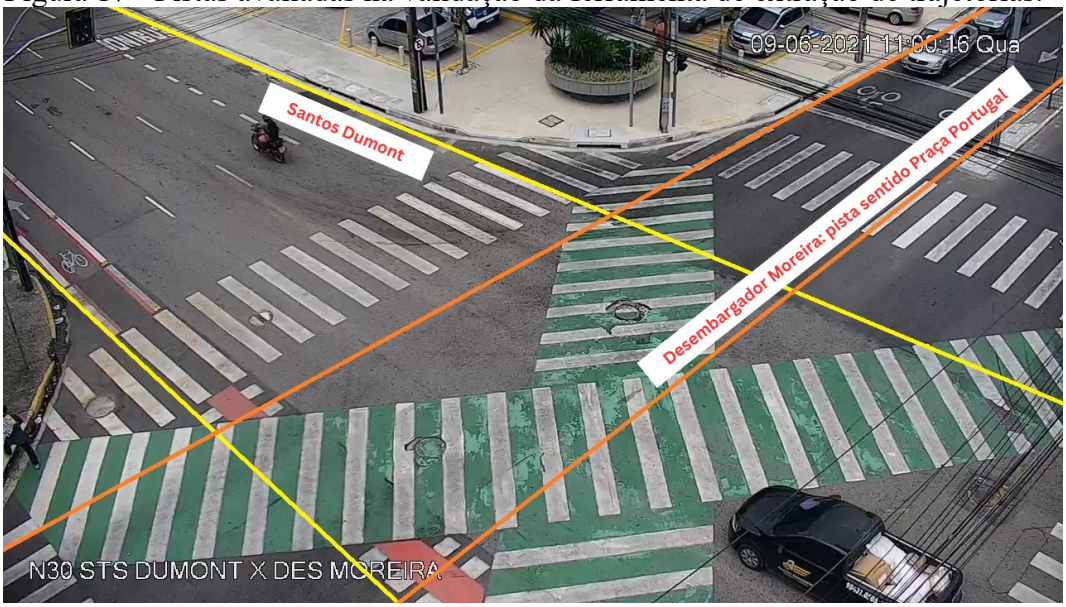
Analisando as métricas $HOTA_{\alpha=0,05}$, $LocA_{\alpha=0,05}$, *HOTA* e *LocA*, percebe-se que a ferramenta de rastreio apresenta dificuldades quando se eleva o limiar de localização α, visto que para o menor valor de $\alpha=0,05$ o valor da métrica $HOTA_{\alpha}$ é relativamente alto, porém resultando em uma sobreposição média (*IoU* médio) de 16,73% entre as anotações manuais e as detecções do modelo. Porém, para métrica final *HOTA* com valor de 13,25% (uma queda de 67,73% em relação à $HOTA_{\alpha=0,05}$), que representa a média das métricas $HOTA_{\alpha}$ para α variando de 0,05 a 0,95 a passos de 0,05, essa semelhança sobe para 61,04% (acréscimo de 44,31%). Já quanto às métricas *IDP*, *IDR* e *IDF1*, todas se aproximam de 50%. Para a métrica *IDP*, isso indica que de todas as identidades associadas aos objetos detectados, um pouco mais da metade receberam a correta identidade. Já quanto a métrica *IDR*, de todas as caixas delimitadoras anotadas manualmente (*ground truths*), pouco menos da metade recebeu a correta identidade. Apesar dos valores das métricas serem medianos, ainda é necessário maiores esforços de validação para julgar a aplicabilidade da ferramenta de coleta de trajetórias.

Durante os esforços de calibração, a velocidade de processamento do algoritmo também foi avaliada, obtendo-se uma média de 83,65 milissegundos por quadro, com um desvio padrão de 1.63 milissegundos e com mínimo e máximo de 82,18 e 119,29 milissegundos, respectivamente. Como a sequência utilizada na calibração não é um vídeo, mas sim um conjunto de imagens, o tempo médio de processamento de vídeos pode ser superior, uma vez que é necessário decodificar os quadros antes de passá-los para o algoritmo de rastreio.

4.3 Validação da ferramenta de extração de trajetórias

O vídeo de monitoramento da interseção entre a avenida Santos Dumont com a avenida Desembargador Moreira utilizado nos esforços de validação da ferramenta está representado na Figura 17 abaixo. Nela estão delimitadas as pistas analisadas, sendo estas a avenida Santos Dumont e a pista sentido Praça Portugal da avenida Desembargador Moreira. Devido ao posicionamento da câmera, a contagem de veículos na avenida Santos Dumont foi feita à jusante da interseção, visto que as conversões de entrada na avenida Desembargador Moreira não são completamente visíveis. Em consequência disto, apenas as conversões de entrada na avenida Santos Dumont foram verificadas, sendo este tipo de conversão a única claramente visível na gravação. Por questões de simplificação da escrita, a partir deste ponto estas conversões serão referidas como conversões à direita. Ademais, apenas as travessias de pedestres nas faixas perpendiculares às pistas foram avaliadas.

Figura 17 - Pistas avaliadas na validação da ferramenta de extração de trajetórias.



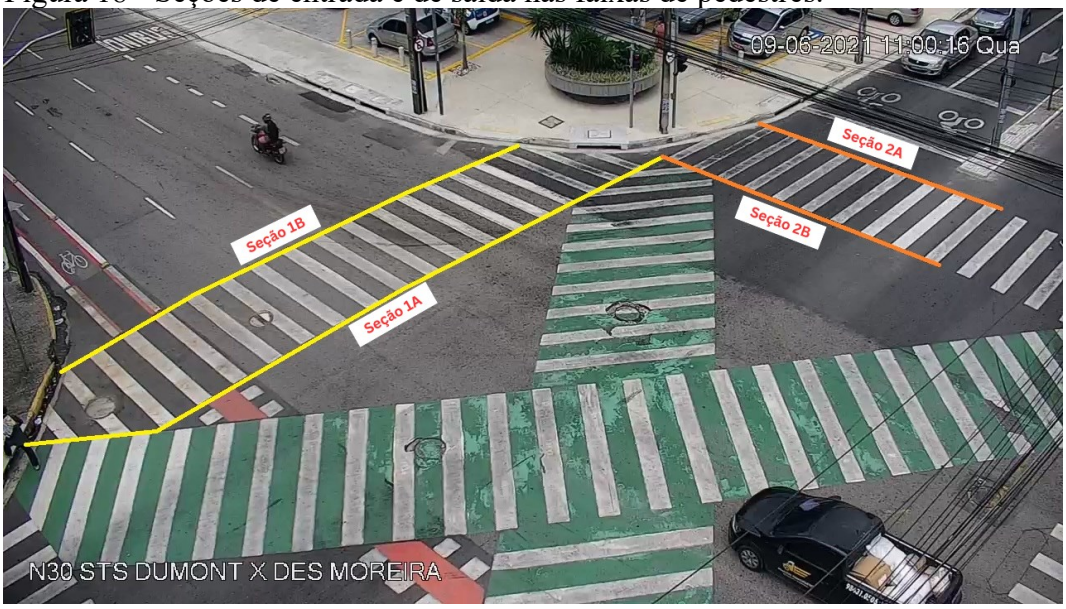
Fonte: autoria própria.

4.3.1 Coleta manual

4.3.1.1 Veículos

A contagem de veículos trafegando nas pistas analisadas se deu em conjunto com a coleta dos instantes de entrada e de saída dos veículos nas faixas de pedestres. A Figura 18 ilustra as seções que delimitam o momento de entrada e saída de veículos para ambas as pistas.

Figura 18 - Seções de entrada e de saída nas faixas de pedestres.



Fonte: autoria própria.

A contagem de veículos abrangeu o tráfego em ambas as direções, sendo que os usuários que trafegavam na contramão foram exclusivamente ciclistas. Para os veículos trafegando no sentido correto, o instante de entrada foi coletado como o instante de tempo em que o pneu dianteiro do veículo estava localizado sobre ou após as seções 1A ou 2A, enquanto o instante de saída segue a mesma referência, porém para as seções 1B e 2B. Para o tráfego na contramão, o procedimento é o mesmo, com exceção de que as seções de entrada são as do tipo B e as seções de saída são as do tipo A.

Para coleta desses instantes, foi utilizado a ferramenta *RUBA* (AGERHOLM *et al.*, 2017), que permite a análise quadro a quadro do vídeo. Uma vez que o vídeo possui uma taxa média de 10 quadros por segundo, a precisão dos instantes é de no máximo 0,1 segundos. A Tabela 8 abaixo traz os resultados de contagem classificatória obtidos na coleta manual, enquanto a Tabela 9 traz o tempo médio que os veículos levam para percorrer a largura da faixa de pedestres, incluindo um intervalo de confiança amostral de 95% segundo a distribuição t de Student.

Tabela 8 - Contagens classificatórias segundo coleta manual (continua).

Tipo de usuário	**Pista**		**Conversões à direita**
	Av. Santos Dumont	**Av. Desembargador Moreira**	
Ônibus	46	22	8

Tabela 8 - Contagens classificatórias segundo coleta manual (conclusão).

Carro	894	685	284
Ciclista	48	6	2
Motociclista	252	191	74
Picape	67	53	22
Caminhão	24	23	9
Van	22	17	9
Total	1353	997	408

Fonte: autoria própria.

Tabela 9 - Tempo médio para percurso da faixa de pedestres segundo coleta manual.

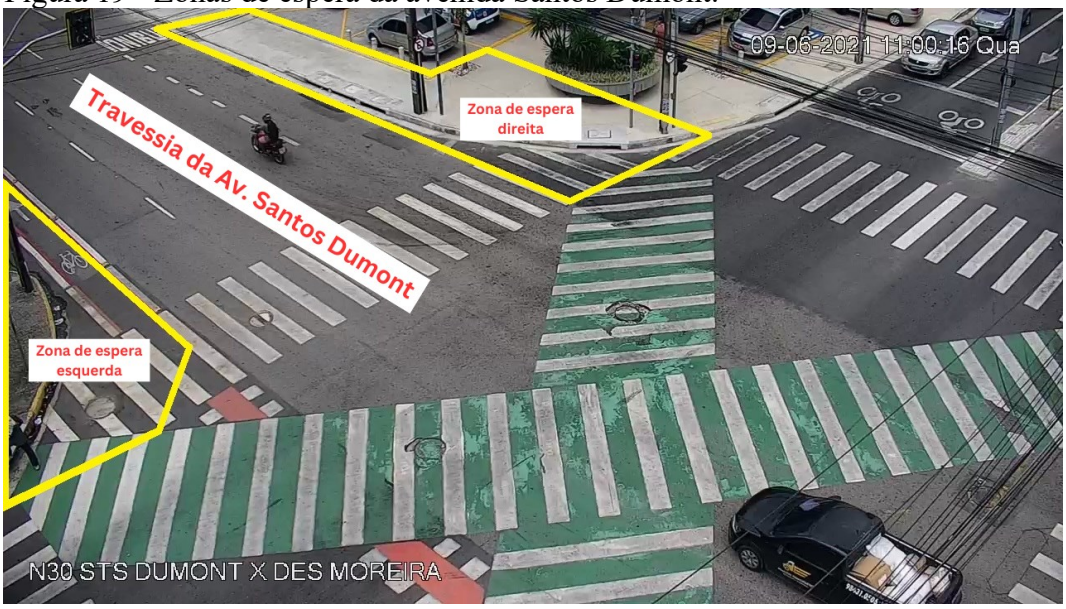
Tipo de usuário	**Tempo médio de travessia**	
	Av. Santos Dumont	**Av. Desembargador Moreira**
Ônibus	0,54 ± 0,06	0,74 ± 0,12
Carro	0,66 ± 0,03	0,72 ± 0,02
Ciclista	2,92 ± 2,53	1,00 ± 0,34
Motociclista	0,61 ± 0,04	0,84 ± 0,19
Picape	0,64 ± 0,06	0,72 ± 0,04
Caminhão	0,68 ± 0,11	0,76 ± 0,09
Van	0,75 ± 0,14	0,84 ± 0,13
Total	0,72 ± 0,09	0,75 ± 0,04

Fonte: autoria própria.

4.3.1.2 Pedestres

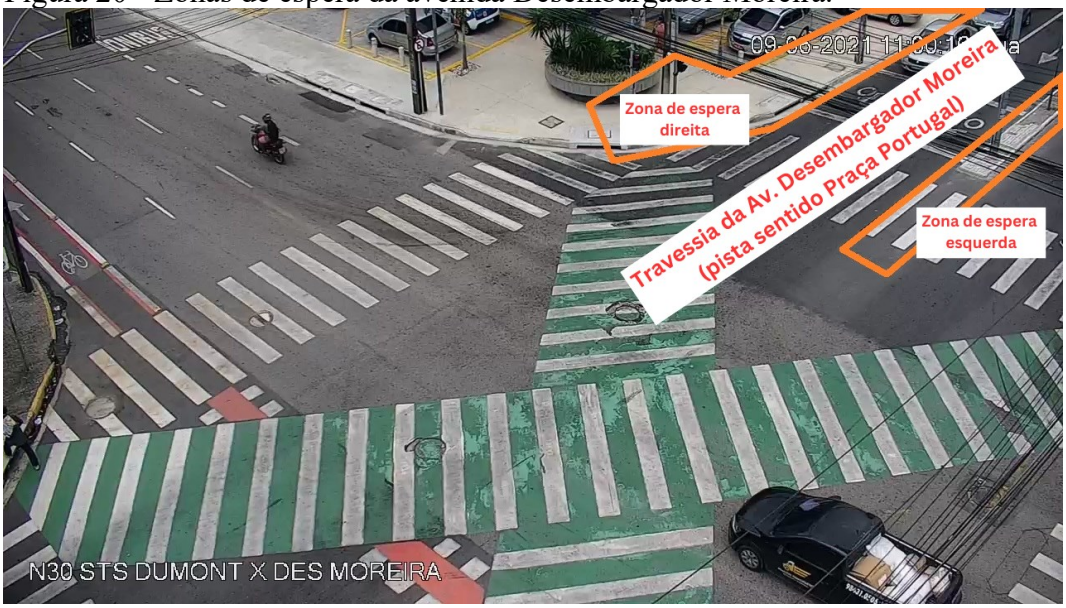
Como dito anteriormente, apenas as travessias de pedestres nas faixas perpendiculares às pistas foram analisadas. As figuras 19 e 20 a seguir ilustram as zonas de espera que foram utilizadas como referência para coleta dessas travessias, onde a nomenclatura da zona (zona de espera esquerda ou direita) está em referência ao sentido do tráfego na pista. Apenas as travessias que iniciaram dentro de uma zona de espera e alcançaram a zona oposta foram coletas, ou seja, travessias que começaram ou terminaram fora das zonas de espera não foram consideradas.

Figura 19 - Zonas de espera da avenida Santos Dumont.



Fonte: autoria própria.

Figura 20 - Zonas de espera da avenida Desembargador Moreira.



Fonte: autoria própria.

Para cada travessia válida, foram coletadas como variáveis a direção de travessia e os instantes em que o pedestre começa a esperar, começa a atravessar e termina de atravessar. A direção de travessia pode ser da esquerda para direita ou da direita para esquerda, a depender de qual a zona de espera inicial. Quanto ao início da espera para travessia, foi coletado o instante em que o pedestre alcança a zona de espera inicial ou o instante em que ele aparentar ter iniciado as avaliações para travessia (parou para esperar, apertou o botão de travessia, começou a avaliar o trânsito na pista ou avaliou o sinal semafórico), caso este último seja perceptível para o

avaliador, selecionando o que ocorrer primeiro. Para o começo da travessia, foi coletado o instante em que o pedestre deixa a zona de espera inicial. Caso este pare para esperar após a saída da zona de espera e antes de atravessar por completo a primeira faixa veicular (excluindo-se a ciclofaixa), é selecionado como instante de início da travessia o momento em que o pedestre volta a se movimentar em direção a zona de espera oposta. Por fim, para o fim da travessia, foi coletado o instante em que o pedestre ingressa na zona de espera oposta à zona de espera inicial. A Tabela 10 traz o resumo dos resultados da coleta manual de travessias, onde os tempos médios são acompanhados de intervalos de confiança amostral de 95% segundo distribuição t de Student.

Tabela 10 - Resumo dos resultados de travessia obtidos na coleta manual.

Descrição	**Av. Santos Dumont**		**Av. Desembargador Moreira**	
Zona de espera inicial	esquerda	direita	esquerda	direita
Contagem	32	30	54	42
Tempo de espera médio	10,36 ± 5,29	16,89 ± 5,99	8,01 ± 3,54	7,56 ± 3,74
Tempo de travessia médio	9,51 ± 1,08	8,65 ± 0,74	5,45 ± 0,20	5,33 ± 0,38

Fonte: autoria própria.

4.3.2 Coleta automatizada

A ferramenta de extração de trajetórias retorna um arquivo CSV (*Comma-Separated Values*) com os dados de rastreio dos objetos, onde cada linha do arquivo corresponde a um ponto de trajetória. As informações que compõem cada linha, ou seja, cada ponto de trajetória, são: número do quadro em que o objeto aparece (comumente chamado de *frame id*), identidade do objeto (representado por um número inteiro comumente chamado de *track id*), classe do objeto e as coordenadas do objeto na imagem (coordenadas do ponto superior esquerdo, largura e altura da caixa delimitadora). A Tabela 11 exemplifica um trecho dos dados retornados pela ferramenta. Com base nesses dados, foram desenvolvidas rotinas automatizadas que buscam aproximar as variáveis-alvo.

Tabela 11 - Exemplo de dados retornados pela ferramenta de extração de trajetórias (continua).

Quadro (Frame)	**Identidade**	**Classe**	**Canto superior esquerdo**		**Largura da caixa delimitadora**	**Altura da caixa delimitadora**
			Coordenada *X*	**Coordenada *Y***		

Tabela 11 - Exemplo de dados retornados pela ferramenta de extração de trajetórias (conclusão).

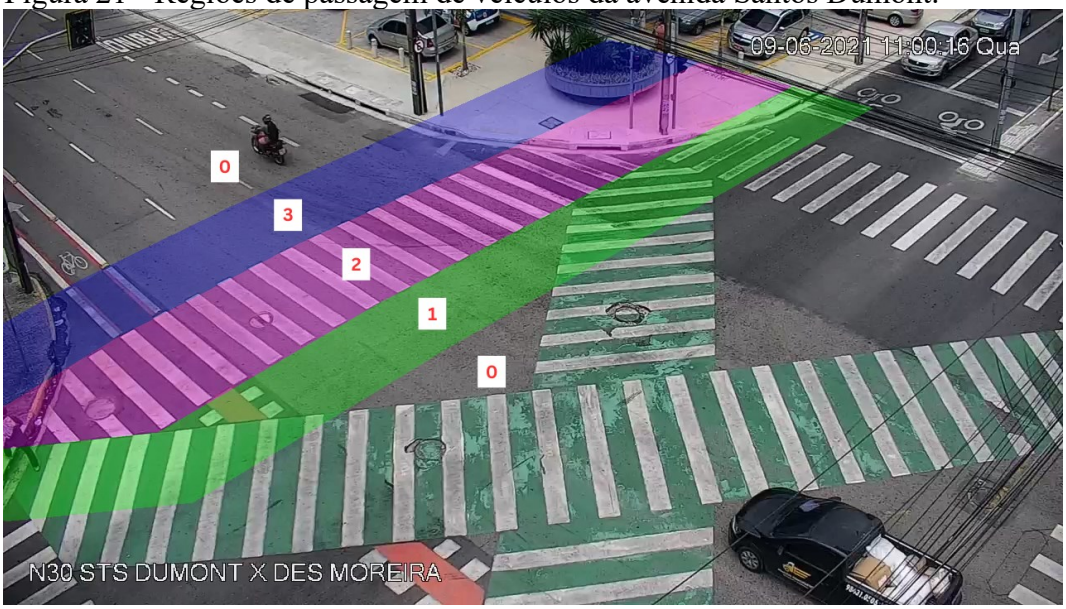
4	8	Ciclista	998 px	68 px	36 px	64 px
4	13	Ônibus	1161 px	0 px	80 px	28 px
5	8	Ciclista	993 px	70 px	40 px	64 px
5	13	Ônibus	1165 px	0 px	76 px	27 px
6	8	Ciclista	988 px	73 px	41 px	63 px
6	13	Ônibus	1167 px	0 px	74 px	28 px

Fonte: autoria própria.

4.3.2.1 Veículos

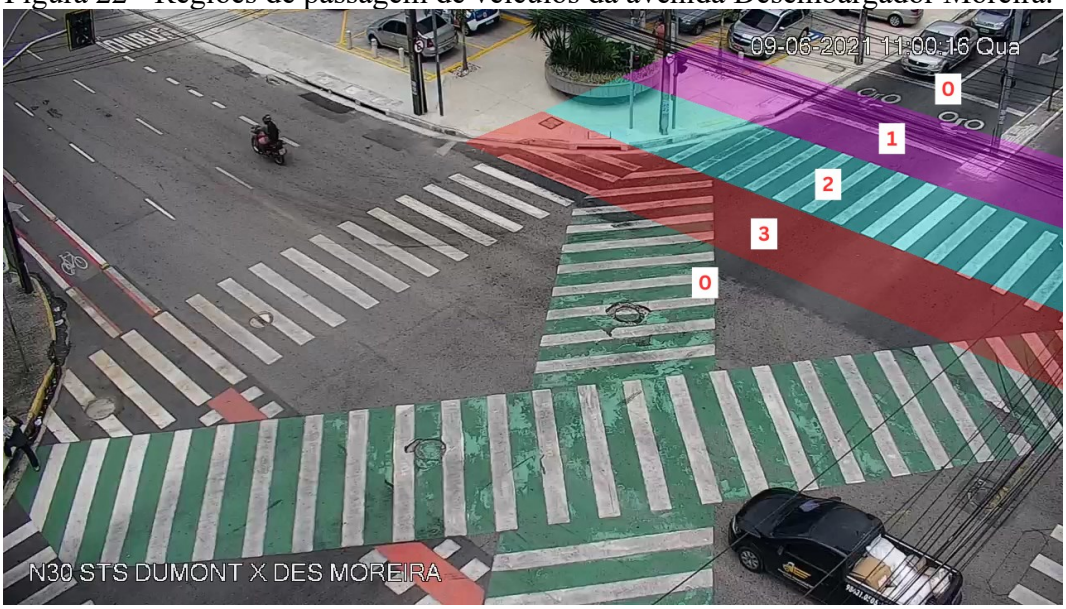
Para a automatização da contagem veicular e dos instantes de entrada e saída de veículos nas faixas de pedestres, foram delimitadas regiões na imagem que serviram para determinação de quando os veículos entram e saem destas regiões. As figuras 21 e 22 ilustram e enumeram estas regiões para ambas as pistas analisadas.

Figura 21 - Regiões de passagem de veículos da avenida Santos Dumont.



Fonte: autoria própria.

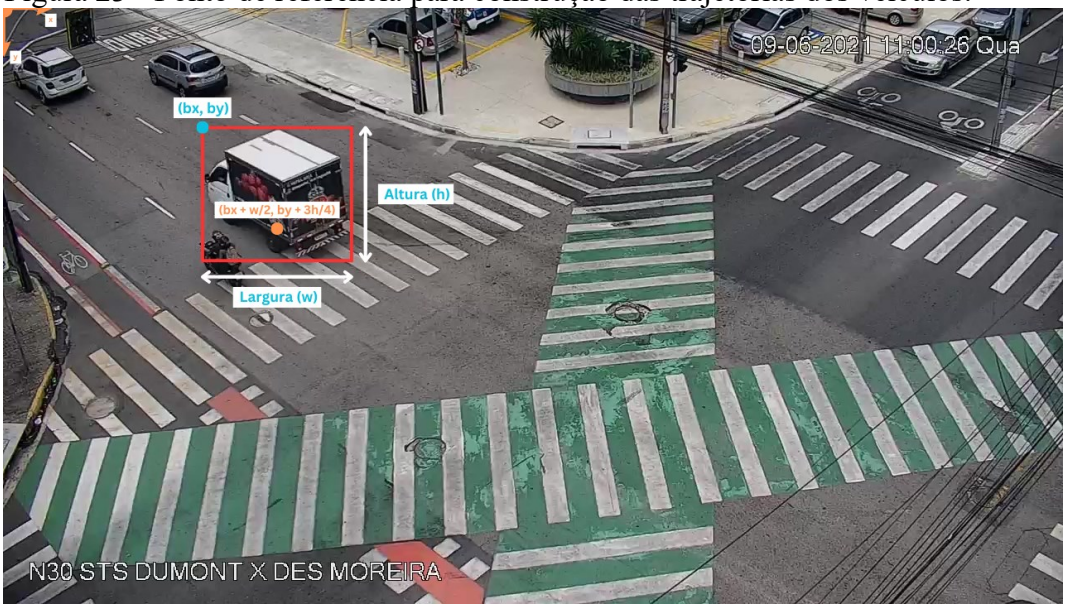
Figura 22 - Regiões de passagem de veículos da avenida Desembargador Moreira.



Fonte: autoria própria.

Os pontos que formam as trajetórias de cada veículo foram computados a partir das informações de coordenada do ponto superior esquerdo e de largura e altura das caixas delimitadoras que são retornadas pela ferramenta. A referência para estes pontos está ilustrada na Figura 23 a seguir, a qual foi escolhida vista a sua generalidade em relação a direção e sentido de movimento do veículo, bem como para se ter um ponto de referência mais próximo do plano da pista, útil principalmente para veículos altos, como caminhões e ônibus.

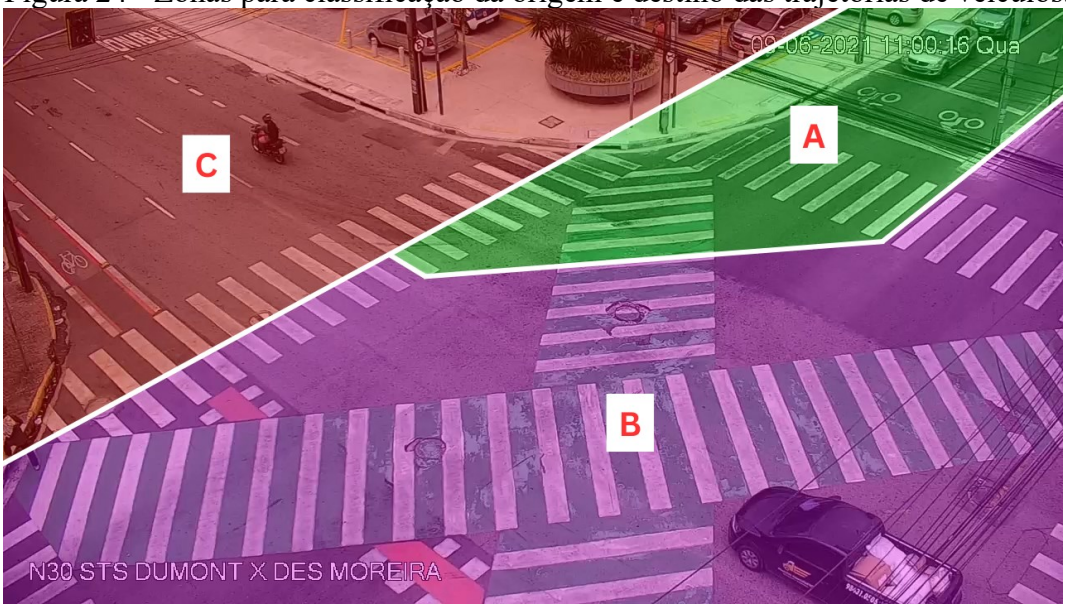
Figura 23 - Ponto de referência para construção das trajetórias dos veículos.



Fonte: autoria própria.

Ademais, cada trajetória teve sua origem e destino classificados de acordo com as zonas ilustradas na Figura 24, com o objetivo de determinar quais trajetórias correspondem a conversões à direita.

Figura 24 - Zonas para classificação da origem e destino das trajetórias de veículos.



Fonte: autoria própria.

Para cada trajetória veicular retornada pela ferramenta, foram analisados os trechos em que esta trajetória percorre as regiões 1, 2 e 3 em sequência, respectivamente nesta ordem para passagens no sentido correto e na ordem inversa (3, 2 e 1) para passagens na contramão. Foram consideradas para contagem das passagens veiculares apenas as trajetórias que passam pelo menos uma vez por estas três regiões, em sequência e na ordem especificada. Ademais, uma mesma trajetória pode apresentar mais de uma passagem veicular em uma mesma seção de análise, uma vez que é possível que diferentes usuários compartilhem uma mesma identidade devido a possíveis trocas de identidade.

Para cada passagem veicular encontrada, foi tomado o número do quadro em que o veículo ingressa na região 2 para determinação do instante de entrada na faixa de pedestres e o número do quadro em que o veículo ingressa na região seguinte (região 3 em caso de passagens no sentido correto ou região 1 em caso de passagens na contramão) para determinação do instante de saída da faixa de pedestres, sendo que o primeiro quadro do vídeo é o quadro de número 1. A partir destes quadros e com base na taxa média de quadros por segundos do vídeo (10 quadros/s), foi aproximado tais instantes de passagem utilizando a Equação 14 abaixo.

$$Instante\ (s) = (nº\ do\ quadro\ -\ 1)\ /\ 10 \quad (14)$$

Para a contagem classificatória do número de conversões à direita, foram contadas as trajetórias veiculares cujo primeiro ponto da trajetória está dentro da zona A e o último está dentro da zona C (vide Figura 24), porém apenas caso a trajetória possua pelo menos uma passagem veicular válida em pelo menos uma das duas seções analisadas (vide figuras 21 e 22).

As tabelas 12 e 13 trazem os resultados obtidos a partir da automatização da rotina descrita acima. As médias reportadas pela Tabela 13 são acompanhadas de intervalos de confiança amostral de 95% segundo a distribuição t de Student.

Tabela 12 - Contagens classificatórias segundo coleta automatizada.

Tipo de usuário	**Pista**		**Conversões à direita**
	Av. Santos Dumont	**Av. Desembargador Moreira**	
Ônibus	46	22	8
Carro	900	693	287
Ciclista	37	4	2
Motociclista	261	185	73
Picape	62	43	20
Caminhão	23	23	8
Van	23	16	9
Total	1352	986	407

Fonte: autoria própria.

Tabela 13 - Tempo médio para percurso da faixa de pedestres segundo coleta automatizada (continua).

Tipo de usuário	**Tempo médio de travessia**	
	Av. Santos Dumont	**Av. Desembargador Moreira**
Ônibus	0,52 ± 0,07	0,69 ± 0,11
Carro	0,66 ± 0,03	0,68 ± 0,02
Ciclista	2,56 ± 2,85	1,12 ± 0,53
Motociclista	0,62 ± 0,04	0,66 ± 0,04
Picape	0,65 ± 0,08	0,69 ± 0,06

Tabela 13 - Tempo médio para percurso da faixa de pedestres segundo coleta automatizada (conclusão).

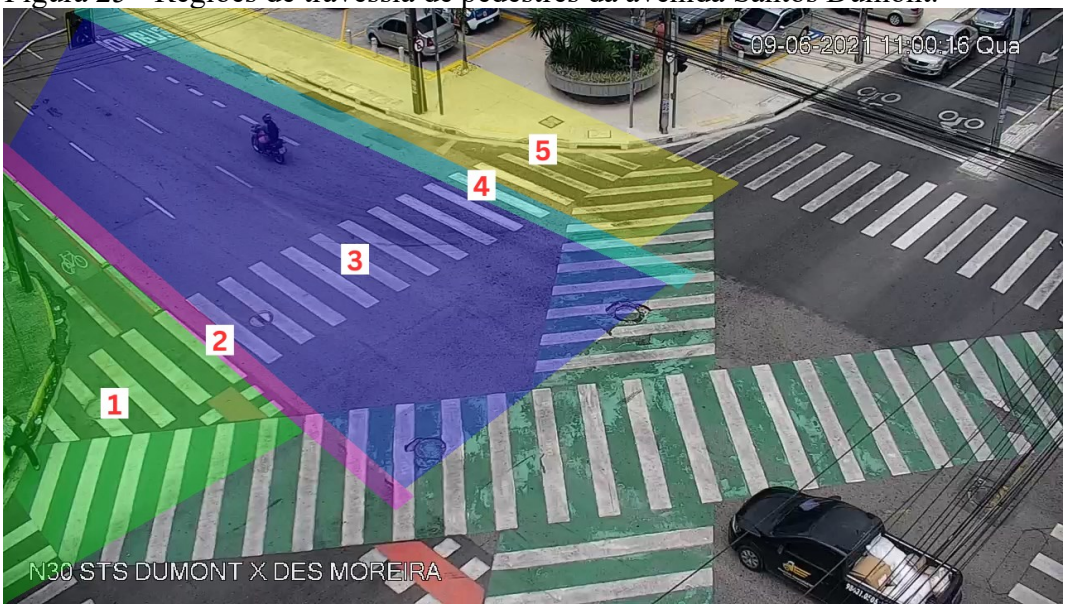
Caminhão	0,67 ± 0,14	0,70 ± 0,09
Van	0,76 ± 0,17	0,84 ± 0,16
Total	0,70 ± 0,08	0,68 ± 0,01

Fonte: autoria própria.

4.3.2.2 Pedestres

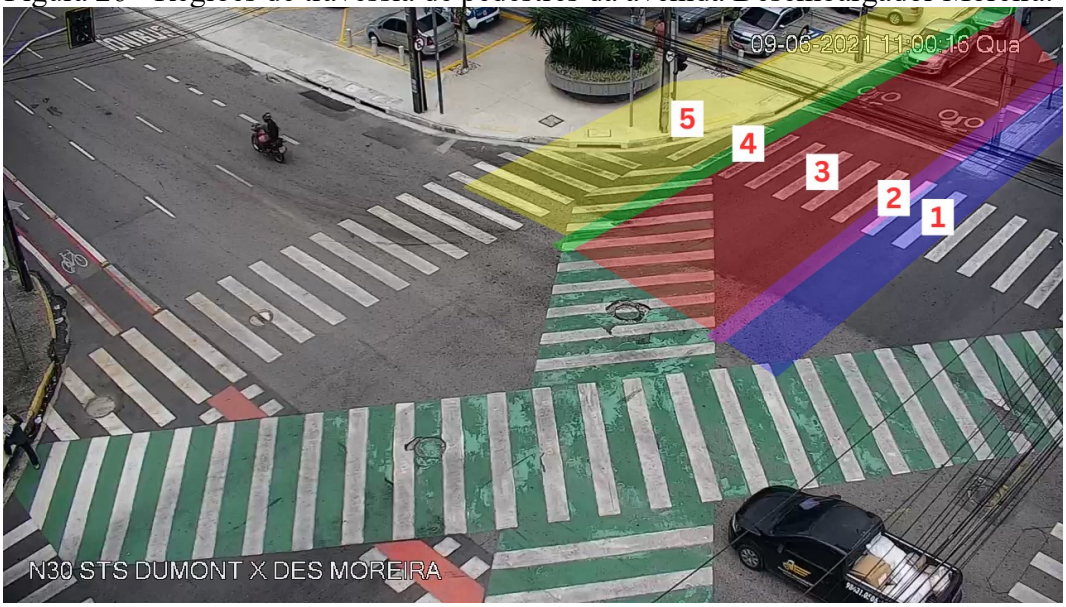
Para a automatização da coleta das variáveis de travessia de pedestres, foi utilizada uma estratégia semelhante à utilizada para os veículos, com a criação de regiões auxiliares que permitem a determinação dos momentos em que o pedestre entra e sai destas regiões. As figuras 25 e 26 ilustram e enumeram estas regiões para as zonas de travessia analisadas.

Figura 25 - Regiões de travessia de pedestres da avenida Santos Dumont.



Fonte: autoria própria.

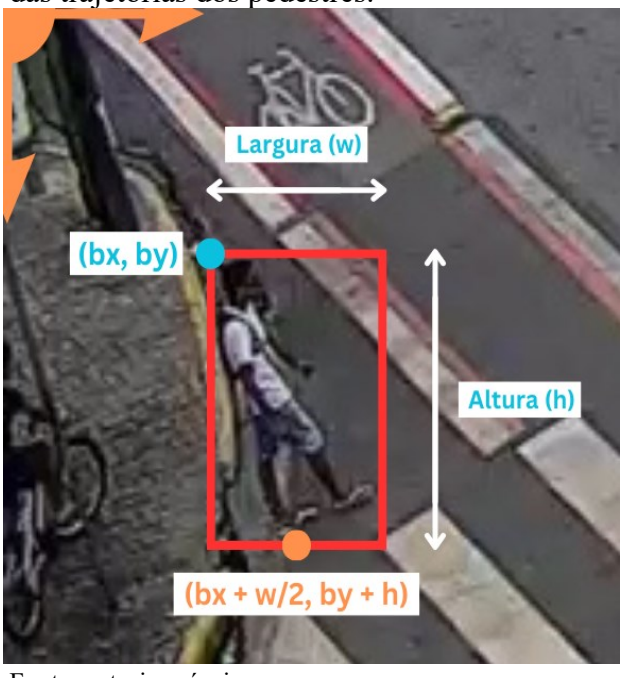
Figura 26 - Regiões de travessia de pedestres da avenida Desembargador Moreira.



Fonte: autoria própria.

Semelhante à construção das trajetórias dos veículos, os pontos que formam as trajetórias de cada pedestre foram computados a partir das informações de coordenada do ponto superior esquerdo e de largura e altura das caixas delimitadoras. A referência para estes pontos está ilustrada da Figura 27 a seguir. Novamente, essa referência foi escolhida de forma a ser independente da direção e do sentido de movimento do pedestre e ser o mais próximo possível da pista de rolamento.

Figura 27 - Ponto de referência para construção das trajetórias dos pedestres.



Fonte: autoria própria.

Para cada trajetória de pedestre retornada pela ferramenta, foram analisados os trechos em que esta trajetória percorre as regiões 1, 2, 3, 4 e 5 em sequência, respectivamente nesta ordem para passagens da esquerda para direita (em relação ao fluxo veicular da pista) e na ordem inversa (5, 4, 3, 2 e 1) para passagens da direita para a esquerda. Para a contagem das travessias de pedestres, apenas trajetórias que passam por estas cinco regiões, em sequência e na ordem especificada foram consideradas. Ademais, semelhante a contagem das passagens de veículos, uma mesma trajetória de pedestre pode apresentar mais de uma travessia em uma mesma seção de análise, uma vez que é possível que diferentes pedestres compartilhem uma mesma identidade devido a possíveis trocas de identidade entre usuários.

Para cada sequência de travessia encontrada, foram coletados o sentido da travessia, o número do quadro em que o pedestre ingressa na primeira região de espera (região 1 para travessias esquerda-direita e região 5 para travessias direita-esquerda), o número do quadro em que o pedestre sai da primeira âncora de travessia (região 2 para travessias esquerda-direita e região 4 para travessias direita-esquerda) e o número do quadro em que o pedestre sai da segunda âncora de travessia (região 4 para travessias esquerda-direita e região 2 para travessias direita-esquerda). Os números de quadros coletados correspondem, respectivamente, ao momento em que o pedestre começa a esperar para atravessar, o momento que ele inicia a travessia da pista e o momento que ele termina de atravessar. A Equação 14 foi utilizada novamente para estimar o instante em segundos de cada momento a partir do número do quadro coletado. A Tabela 14 traz o resumo dos resultados da coleta automatizada de travessias, onde as médias estão acompanhadas de intervalos de confiança amostral de 95% segundo a distribuição t de Student.

Tabela 14 - Resumo dos resultados de travessia obtidos na coleta automatizada.

Descrição	**Av. Santos Dumont**		**Av. Desembargador Moreira**	
Zona de espera inicial	esquerda	direita	esquerda	direita
Contagem	31	29	51	40
Tempo de espera médio (s)	11,42 ± 5,48	18,58 ± 5,52	8,16 ± 3,62	10,61 ± 4,67
Tempo de travessia médio (s)	7,32 ± 1,02	5,91 ± 0,56	4,15 ± 0,16	4,07 ± 0,29

Fonte: autoria própria.

4.3.3 Comparação dos resultados

4.3.3.1 Veículos

A Tabela 15 traz a comparação entre os resultados das contagens classificatórias veicular segundo o método manual e o automatizado para a pista Santos Dumont. Observando os valores das métricas, percebe-se que todas as classes de usuários obtiveram precisão superior à 95%, revocação superior à 91% (com exceção de ciclistas que obteve77,08%) e pontuação F1 superior à 94% (também com exceção de ciclistas que obteve 87,06%).

Tabela 15 - Comparação das contagens classificatórias na avenida Santos Dumont.

Tipo de usuário	Coleta Manual	Coleta Automatizada (TPs + FPs)	Acertos (TPs)	Erros (FPs)	Faltas (FNs)	Precisão (%)	Revocação (%)	Pontuação F1 (%)
Ônibus	46	46	46	0	0	100,00%	100,00%	100,00%
Carro	894	900	894	6	0	99,33%	100,00%	99,67%
Ciclista	48	37	37	0	11	100,00%	77,08%	87,06%
Motociclista	252	261	250	11	2	95,79%	99,21%	97,47%
Picape	67	62	61	1	6	98,39%	91,04%	94,57%
Caminhão	24	23	23	0	1	100,00%	95,83%	97,87%
Van	22	23	22	1	0	95,65%	100,00%	97,78%
Total	1353	1352	1333	19	20	98,59%	98,52%	98,56%

Fonte: autoria própria.

Fazendo uma análise individual dos erros e faltas cometidos pelo método automatizado, tem-se:

a) dos 6 erros cometidos na contagem de carros, todos são devidos a classificações errôneas, onde 1 motociclista e 5 picapes foram confundidos como sendo carros. Estes erros correspondem a 1 das 2 faltas observadas na contagem de motociclistas e a 5 das 6 faltas observadas na contagem de picapes;
b) todos os 11 erros cometidos na contagem de motociclistas foram devido a classificação errônea de ciclistas como sendo motociclistas. Estes erros correspondem a todas as faltas observadas na contagem de ciclistas;
c) o único erro cometido na contagem de picapes foi devido a classificação errônea de um caminhão como sendo uma picape. Este erro corresponde a única falta observada na contagem de caminhões;

d) o único erro cometido na contagem de vans foi devido a classificação errônea de uma picape como sendo uma van. Este erro corresponde a 1 das 6 faltas observadas na contagem de picapes;
e) a falta restante na contagem de motociclistas foi devido ao algoritmo de rastreio não ter conseguido iniciar o rastreio de um motociclista em tempo hábil, de modo que a coleta de sua trajetória se iniciou já sobre a faixa de pedestres.

Já para a pista sentido Praça Portugal da avenida Desembargador Moreira, a Tabela 16 traz a comparação entre os resultados das contagens classificatórias segundo os métodos manual e automatizado. Observando os valores das métricas, percebe-se que todas as classes de usuários obtiveram precisão superior à 93%, revocação superior à 88% (com exceção de ciclistas e picapes que obtiveram 66,67% e 77,36%, respectivamente) e pontuação F1 superior à 90% (também com exceção de ciclistas e picapes que obtiveram 80,00% e 85,42%, respectivamente).

Tabela 16 - Comparação das contagens classificatórias na avenida Desembargador Moreira.

Tipo de usuário	Coleta Manual	Coleta Automatizada (TPs + FPs)	Acertos (TPs)	Erros (FPs)	Faltas (FNs)	Precisão (%)	Revocação (%)	Pontuação F1 (%)
Ônibus	22	22	22	0	0	100,00%	100,00%	100,00%
Carro	685	693	679	14	6	97,98%	99,12%	98,55%
Ciclista	6	4	4	0	2	100,00%	66,67%	80,00%
Motociclista	191	185	184	1	7	99,46%	96,34%	97,87%
Picape	53	43	41	2	12	95,35%	77,36%	85,42%
Caminhão	23	23	22	1	1	95,65%	95,65%	95,65%
Van	17	16	15	1	2	93,75%	88,24%	90,91%
Total	997	986	967	19	30	98,07%	96,99%	97,53%

Fonte: autoria própria.

A análise individual dos erros e faltas do método automatizado mostra:

a) dos 14 erros cometidos na contagem de carros, todos foram devido a classificações errôneas, sendo que 2 vans, 11 picapes e 1 motociclista foram confundidos como sendo carros. Estes erros correspondem a 1 das 6 faltas observadas na contagem de motociclistas, a 11 das 12 faltas observadas na

contagem de picapes e a todas as 2 faltas observadas na contagem de vans;

b) o único erro cometido na contagem de motociclistas foi devido a classificação errônea de um ciclista como sendo um motociclista. Este erro corresponde a 1 das 2 faltas observadas na contagem de ciclistas;
c) dos dois erros cometidos na contagem de picapes, ambos foram devido a classificações errôneas, sendo que 1 carro e 1 caminhão foram confundidos como sendo picapes. Estes erros correspondem a 1 das 5 faltas observadas na contagem de carros e a única falta observada na contagem de caminhões;
d) o único erro cometido na contagem de vans foi devido a classificação errônea de um carro como sendo uma van. Este erro corresponde a 1 das 5 faltas observadas na contagem de carros;
e) o único erro cometido na contagem de caminhões foi devido a classificação errônea de uma picape como sendo um caminhão. Este erro corresponde a 1 das 12 faltas observadas na contagem de picapes;
f) das 4 faltas restantes na contagem de carros, 1 foi devido à oclusão de um carro por um ônibus durante a sua passagem pela faixa de pedestres, 2 foram devido a falha da ferramenta de rastreio em coletar as trajetórias e 1 foi devido a perda da identidade nas proximidades da faixa de pedestres (a alocação de uma nova identidade não é imediata, o que acarretou a perdendo do momento de entrada do carro na faixa de pedestres);
g) a falta restante na contagem de ciclistas foi devido a oclusão de um ciclista por um ônibus durante a sua passagem pela faixa de pedestres;
h) das 5 faltas restantes na contagem de motociclistas, 1 foi devido à oclusão do motociclista por um ônibus durante a sua passagem pela faixa de pedestres, 3 foram devido à ferramenta de rastreio tardar ou não conseguir iniciar o rastreio dos motociclistas (as trajetórias começaram a ser coletadas após os motociclistas já terem passado pela faixa de pedestres) e 2 foram devido a perda da identidade nas proximidades da faixa de pedestre.

Por fim, quanto à contagem de conversões à direita, a Tabela 17 a seguir traz as comparações entre a contagem manual e a contagem automatizada. Observando os valores das métricas, percebe-se que todas as classes de usuários obtiveram precisão igual ou superior à 95%, revocação superior à 85% e pontuação F1 superior à 90%.

Tabela 17 - Comparação das contagens classificatórias de conversões à direita.

Tipo de usuário	Coleta Manual	Coleta Automatizada (TPs + FPs)	Acertos (TPs)	Erros (FPs)	Faltas (FNs)	Precisão (%)	Revocação (%)	Pontuação F1 (%)
Ônibus	8	8	8	0	0	100,00%	100,00%	100,00%
Carro	284	287	283	4	1	98,61%	99,65%	99,12%
Ciclista	2	2	2	0	0	100,00%	100,00%	100,00%
Motociclista	74	73	73	0	1	100,00%	98,65%	99,32%
Picape	22	20	19	1	3	95,00%	86,36%	90,48%
Caminhão	9	8	8	0	1	100,00%	88,89%	94,12%
Van	9	9	9	0	0	100,00%	100,00%	100,00%
Total	408	407	402	5	6	98,77%	98,53%	98,65%

Fonte: autoria própria.

Feita novamente a análise individual dos erros e faltas do método automatizado, tem-se:

a) dos 4 erros cometidos na contagem de conversões à direita de carros, todos são devido a classificações errôneas, onde 1 motociclista e 3 picapes foram confundidos como sendo carros. Estes 4 erros correspondem às faltas na contagem de conversões observadas para motociclistas e picapes;
b) o único erro cometido na contagem de conversões à direita de picapes foi devido a classificação errônea de um caminhão como sendo uma picape. Este erro corresponde à falta observada na contagem de conversões de caminhões;
c) a única falta na contagem de conversões à direita de carros foi devido a troca de identidade entre um pedestre e um carro, de modo que a origem da trajetória predita para o carro fosse na verdade a origem da trajetória de um pedestre.

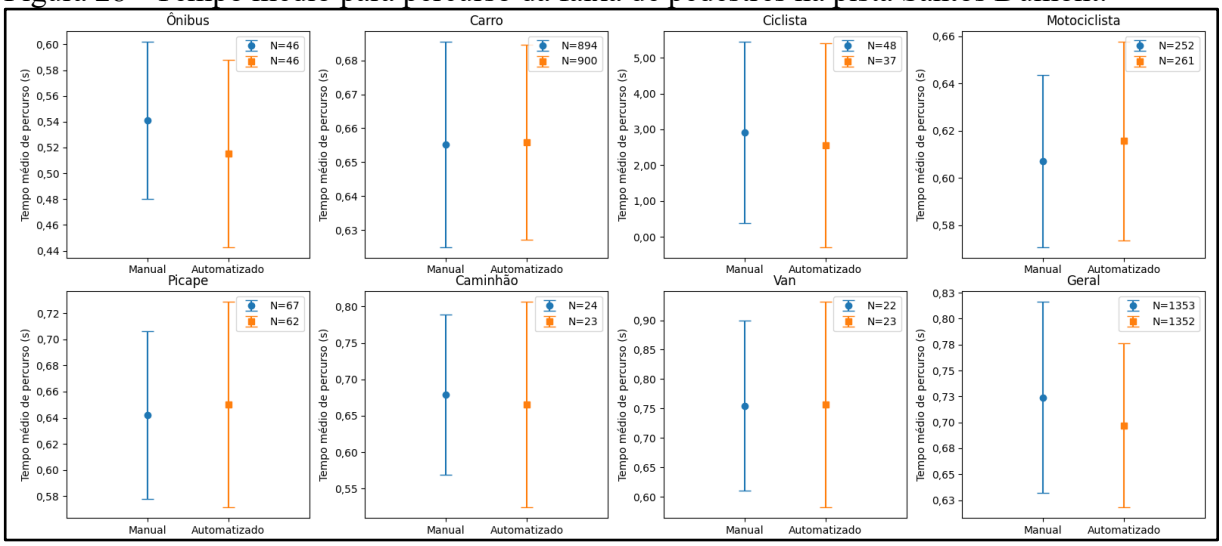
É importante ressaltar que os mesmos erros apontados acima para as conversões à direita também ocorrem nas contagens classificatórias das avenidas Santos Dumont e/ou Desembargador Moreira, visto que trajetórias que representam uma conversão à direita obrigatoriamente passam em pelo menos uma das seções de contagem classificatória analisadas. Dentre esses erros, um importante de ser detalhado é o da classificação errônea de um motociclista como sendo um carro. Este mesmo erro aparece nos três tipos de contagem e foi devido a uma falha do algoritmo de rastreio que permitiu que a identidade de um carro fosse

assumida por um motociclista momentos antes da passagem pela faixa de pedestres. Porém, tal erro poderia ser remediado com o uso de técnicas de pós-processamento das trajetórias, com o fito de dividir trajetórias preditas que possivelmente englobam a trajetória de dois ou mais usuários reais distintos.

Ademais, as confusões entre motociclistas e ciclistas e entre carros, picapes, vans e caminhões observadas nos erros do método automatizado já haviam sido indiciadas pela matriz de confusão do modelo de detecção (vide Figura 14). Tais confusões podem ser atenuadas com novos esforços de treinamento com mais robustez de dados de treino e validação.

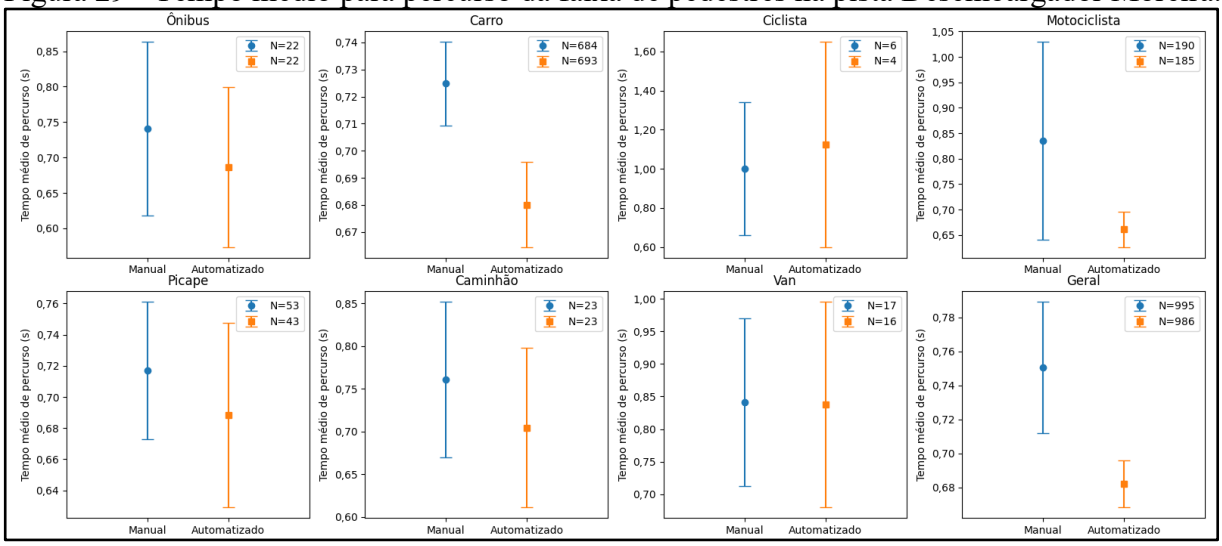
Passando para a análise dos intervalos de tempo que os veículos levam para percorrer a largura da faixa de pedestres, as figuras 28 e 29 abaixo trazem a comparação entre os tempos médios segundo a coleta manual e os tempos médios segundo o método automatizado para ambas as pistas analisadas. Esses tempos médios são acompanhados de intervalos de confiança amostral de 95% de confiança segundo distribuição t de Student.

Figura 28 - Tempo médio para percurso da faixa de pedestres na pista Santos Dumont.



Fonte: autoria própria.

Figura 29 - Tempo médio para percurso da faixa de pedestres na pista Desembargador Moreira.

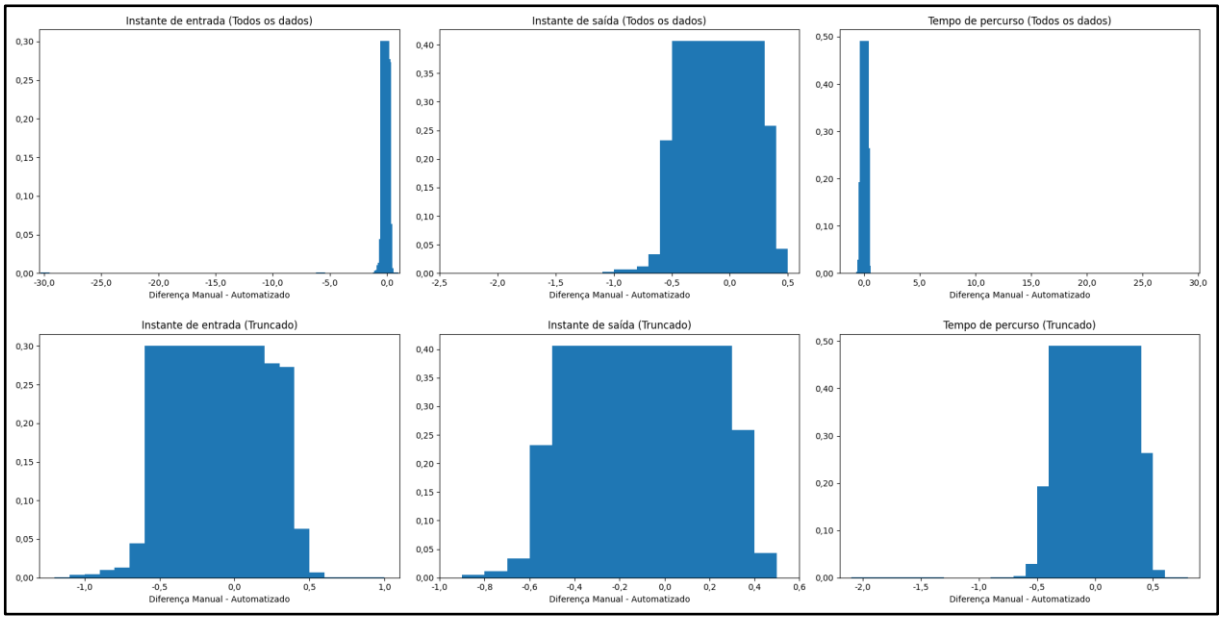


Fonte: autoria própria.

Percebe-se que, para a pista da Santos Dumont, os tempos médios retornados pela coleta manual e pela coleta automatizada são relativamente próximos e os intervalos de confiança possuem uma boa sobreposição, independentemente do tipo de usuário. Entretanto, para a pista da Desembargador Moreira, a depender do tipo de usuário, os tempos médios são mais discrepantes, com destaque para os carros, onde os intervalos de confiança não se sobrepõem, e para os motociclistas, que tem sua média segundo o método automatizado próxima do limite inferior do intervalo de confiança da coleta manual. Uma hipótese plausível para esta discrepância entre os dados obtidos para a pista da Desembargador Moreira é o fato de que a sua faixa de pedestres é ligeiramente menor na imagem em relação a faixa de pedestres da Santos Dumont, uma vez que é uma região mais distante da câmera de gravação, o que torna menos preciso o mapeamento de pontos do “mundo real” para pontos na imagem, já que regiões de tamanho igual no “mundo real” são mapeados para regiões de tamanhos diferentes na imagem, a depender da distância dessas regiões para a câmera.

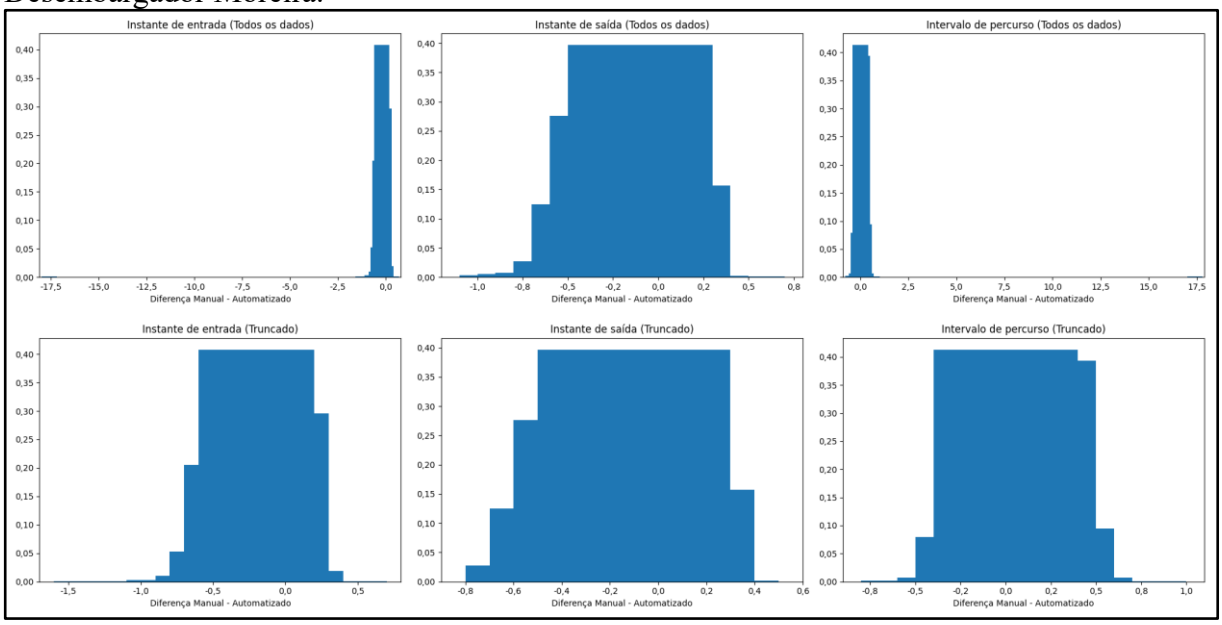
Por fim, as figuras 30 e 31 ilustram a distribuição dos erros (diferenças) entre os instantes de passagem coletados manualmente e os coletados via rotina automatizada para ambas as pistas analisadas.

Figura 30 - Distribuição dos erros dos instantes de entrada e de saída de veículos para a avenida Santos Dumont.



Fonte: autoria própria.

Figura 31 - Distribuição dos erros dos instantes de entrada e de saída de veículos para a avenida Desembargador Moreira.



Fonte: autoria própria.

Percebe-se que, para ambas as pistas, os erros dos instantes de entrada e de saída possuem uma tendência central deslocada para esquerda do erro 0, o que mostra que os instantes retornados pelo método automatizado tendem a ser ligeiramente maiores que os da coleta manual. Entretanto, quanto aos erros do tempo de percurso, percebe-se uma tendência central mais próxima de 0, principalmente para as passagens na avenida Santos Dumont. Tal tendência central para os erros dos instantes de entrada e saída pode ser justificada pelo uso de referências

distintas na coleta manual e na coleta automatizada, onde a coleta manual determinou os instantes de entrada e saída da faixa de pedestres com base no pneu dianteiro do veículo, enquanto a coleta automatizada utilizou como referência um ponto logo abaixo do centroide da caixa delimitadora da detecção (vide Figura 23). Portanto, espera-se que as discrepâncias entre as referências se atenuem quando se analisa o tempo de percurso, o que é visto nos gráficos com tendência central mais próxima do erro zero.

4.3.3.2 Pedestres

A Tabela 18 traz a comparação das contagens de travessias pelo método manual e pelo método automatizado. O método automatizado obteve precisões acima de 97% e revocações acima de 92% em todas as separações analisadas.

Tabela 18 - Comparação das contagens de travessia de pedestres.

Descrição	**Av. Santos Dumont**			**Av. Desembargador Moreira**		
Zona de espera inicial	Esquerda	Direita	Ambas	Esquerda	Direita	Ambas
Coleta Manual	32	30	62	54	42	96
Coleta Automatizada (TPs + FPs)	31	29	60	51	40	91
Acertos (TPs)	31	29	60	51	39	90
Erros (FPs)	0	0	0	0	1	1
Faltas (FNs)	1	1	2	3	3	6
Precisão (%)	100,00%	100,00%	100,00%	100,00%	97,50%	98,90%
Revocação (%)	96,88%	96,67%	96,77%	94,44%	92,86%	93,75%
Pontuação F1 (%)	98,41%	98,31%	98,36%	97,14%	95,12%	96,26%

Fonte: autoria própria.

Após a análise individual dos erros e faltas cometidos pelo método automatizado, tem-se:

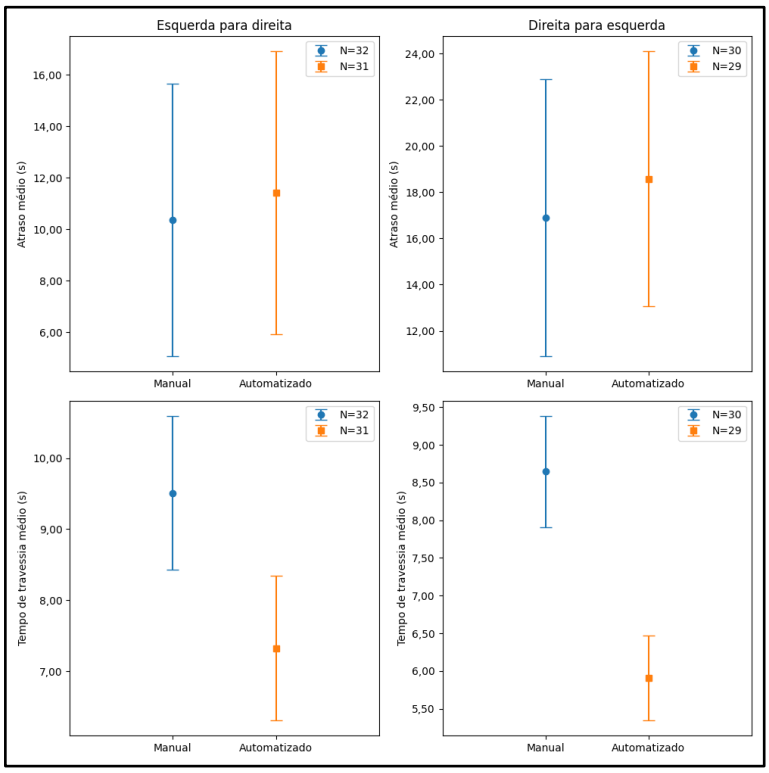
a) o único erro cometido pelo método automático foi devido a uma sucessiva troca de identidade entre um pedestre e dois carros, de modo que a trajetória retornada pela ferramenta de extração na verdade engloba segmentos de trajetórias de três usuários distintos;

b) das duas faltas observadas nas travessias da pista Santos Dumont, uma foi devido a classificação errônea de um pedestre como sendo um ciclista e outra foi devido a oclusão do pedestre por outro pedestre que realizada a travessia ao seu lado;
c) das seis faltas observadas nas travessias da avenida Desembargador Moreira, uma foi devido a falha da ferramenta de rastreio em extrair a trajetória do pedestre e as outras cinco foram devido a oclusão dos pedestres por outros que os acompanhavam na travessia.

Após as avaliações dos erros e das faltas na contagem de travessias, notou-se uma dificuldade da ferramenta de extração de trajetórias em processar situações de aglomeração de pedestres, onde é perceptível trocas de identidade frequentes entre os pedestres próximos e falhas na coleta das trajetórias devido a oclusões parciais. Grande parte das travessias observadas no vídeo de validação são travessias individuais, as quais o pedestre não está próximo ou sendo influenciado por outros pedestres também querendo realizar a travessia. Para estes tipos de travessia, a ferramenta de extração de trajetória se mostrou relativamente robusta, conseguindo capturar boa parte da trajetória dos pedestres dentro e entre as zonas de espera. Por fim, também é notável um número alto de perdas de identidade durante oclusões temporárias, como, por exemplo, ao passar por trás de um poste de luz ou ser ocluído pela passagem de um ônibus.

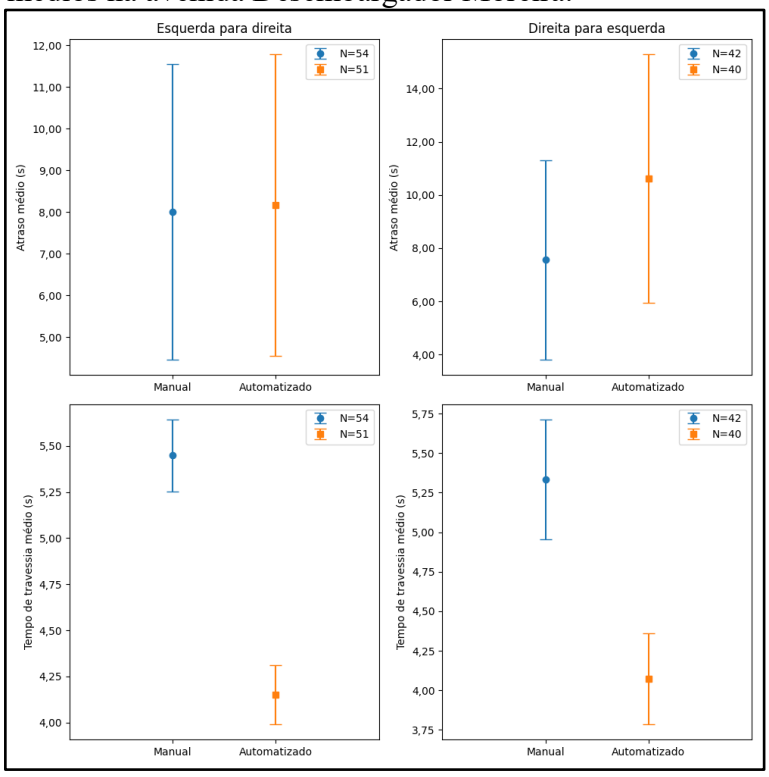
Dando continuidade nas comparações entre os resultados da coleta manual e os da coleta automatizada, as figuras 32 e 33 a seguir comparam o atraso e o tempo de travessia médio obtidos para as pistas Santos Dumont e Desembargador Moreira, respectivamente. Juntamente com os valores médios, são ilustrados intervalos de confiança amostral de 95% segundo a distribuição t de Student.

Figura 32 - Comparação do atraso e tempo de travessia médios na avenida Santos Dumont.



Fonte: autoria própria.

Figura 33 - Comparação do atraso e tempo de travessia médios na avenida Desembargador Moreira.

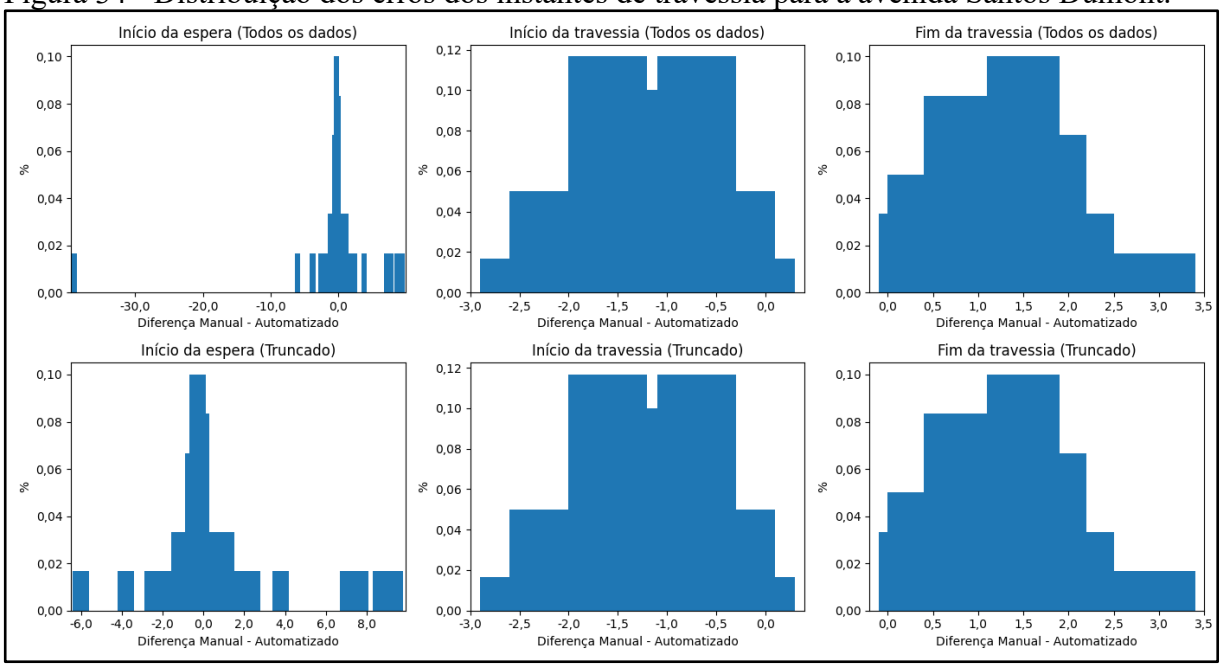


Fonte: autoria própria.

É perceptível em ambas as pistas analisadas que os intervalos de confiança do atraso médio possuem uma boa sobreposição, enquanto os intervalos do tempo de travessia médio não se sobrepõem. Em resumo, o método automatizado superestima ligeiramente o atraso do pedestre e subestima consideravelmente o tempo de travessia. Ademais, as discrepâncias entre os tempos de travessia médio são maiores para os dados da avenida Santos Dumont. Uma fonte de erro presente na coleta automatizada dos instantes de início e fim da travessia, que pode vir a justificar esta discrepância maior nos tempos de travessia da avenida Santos Dumont, reside na delimitação das regiões auxiliares de espera para travessia (vide Figura 25), onde houve a necessidade de estender estas zonas sobre a pista da avenida Santos Dumont, devido a ocorrência frequente de pedestres esperando para atravessar já dentro da pista, como é o caso, por exemplo, de pedestres esperando sobre a ciclofaixa.

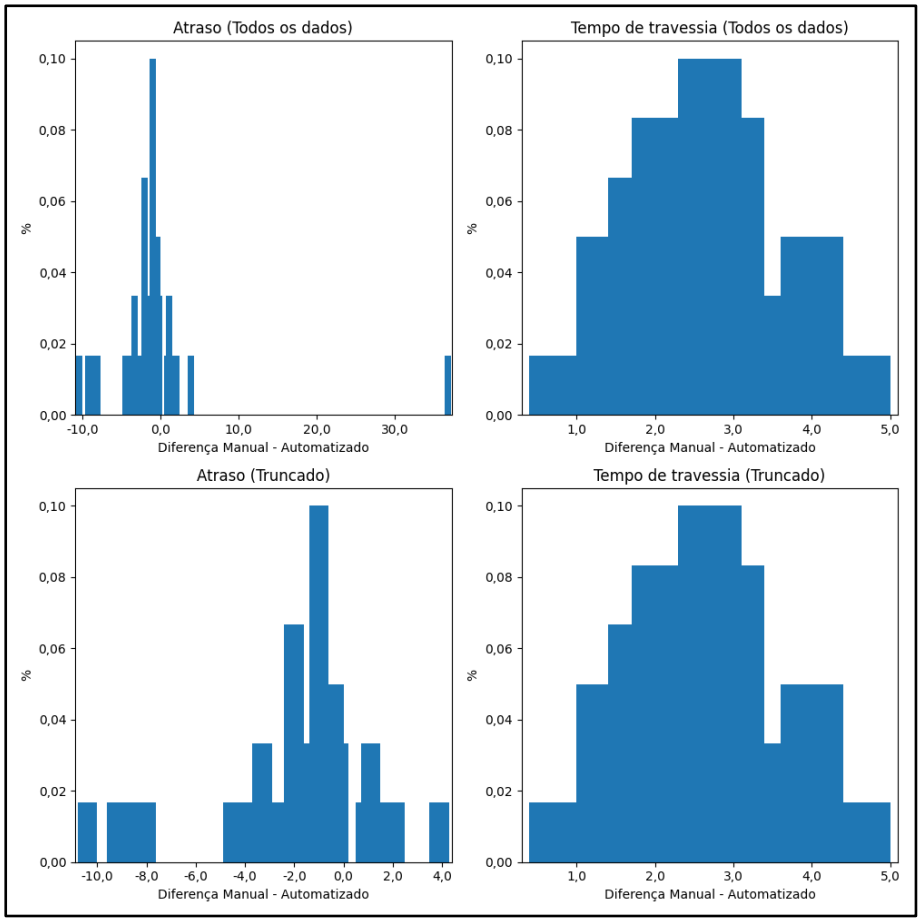
Para as travessias coletadas pelo método automatizado que foram associadas à travessias coletadas manualmente, foram analisadas as distribuições dos erros entre os instantes coletados (início da espera, início da travessia e fim da travessia) e entre os tempos de atraso e de duração da travessia. As figuras 34 e 35 a seguir trazem estas distribuições de erros para as travessias da avenida Santos Dumont.

Figura 34 - Distribuição dos erros dos instantes de travessia para a avenida Santos Dumont.



Fonte: autoria própria.

Figura 35 - Distribuição dos erros de atraso e tempo de travessia para a avenida Santos Dumont.

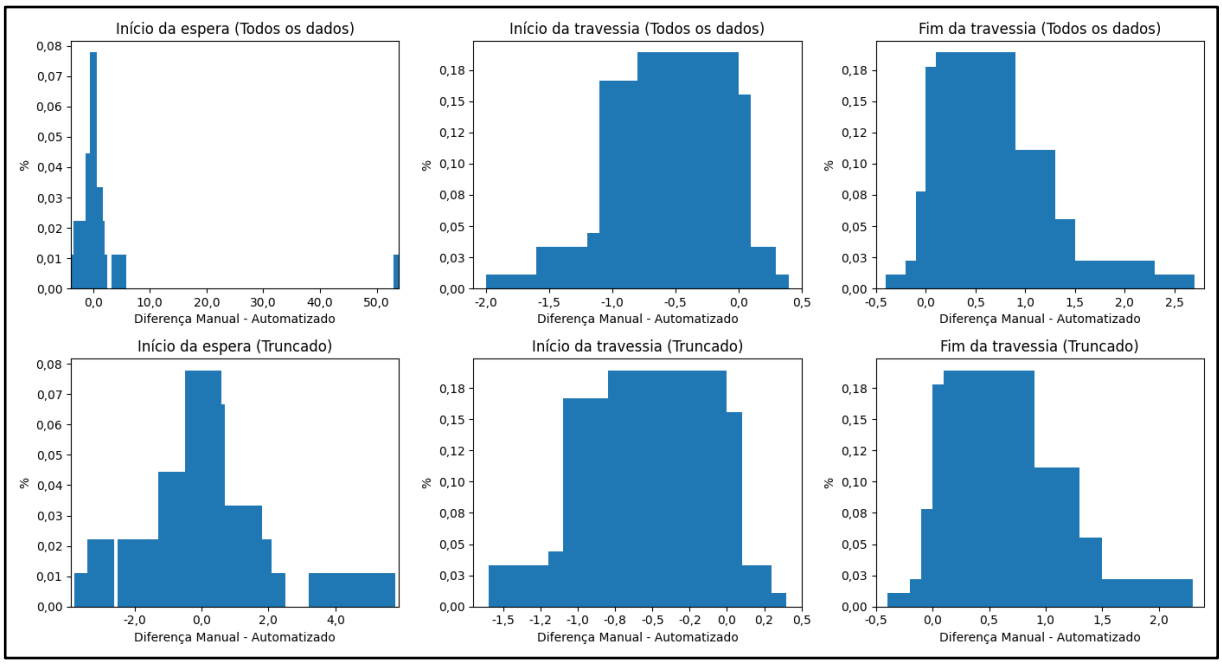


Fonte: autoria própria.

Quanto às distribuições dos erros dos instantes, a que mais se aproxima de uma tendência central sobre o erro zero é a de início da espera, porém apresentando uma dispersão alta em comparação aos demais, o que resulta em instantes de início da espera pouco precisos, mas com tendência central próxima à da coleta manual. Para os instantes de início da travessia e fim da travessia, a tendência central está à esquerda e à direita do erro zero, respectivamente. Dessa forma, o método automatizado tem uma tendência de superestimar o instante que o pedestre começa a travessia e subestimar o instante que ele termina a travessia. Tais características se refletem nas distribuições dos erros do atraso e do tempo de travessia, onde a distribuição do atraso assume a alta dispersão do instante de início da espera e tem sua tendência central deslocada para esquerda devido ao instante de início da travessia, enquanto a distribuição do tempo de espera tem sua tendência central deslocada mais ainda para a direita.

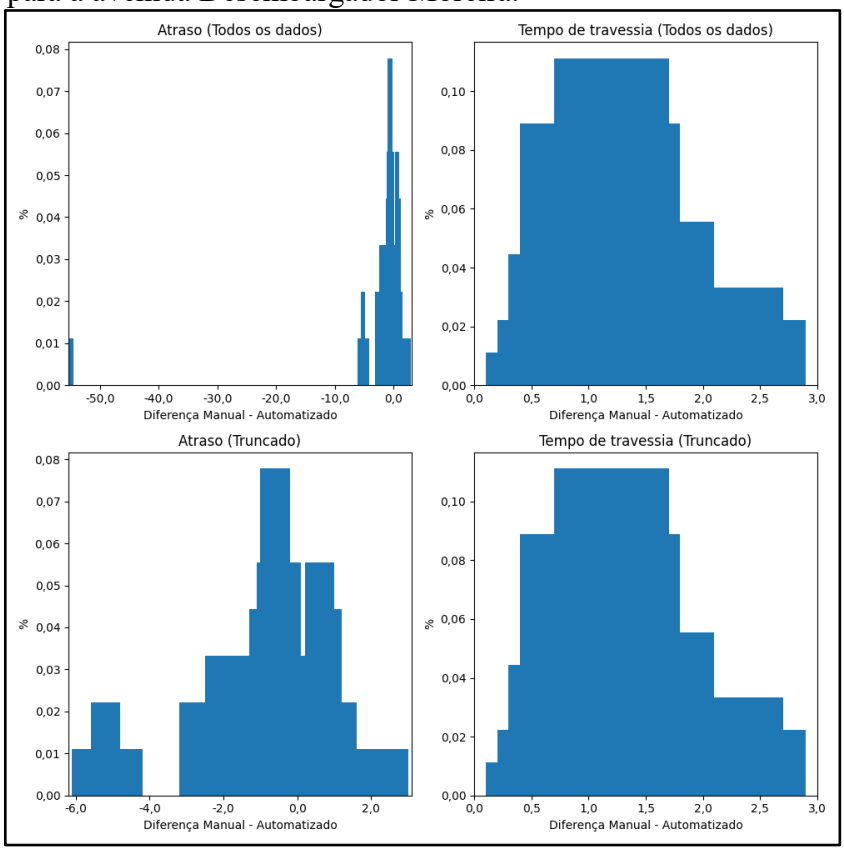
A distribuição de erros para a avenida Desembargador Moreira é similar ao da avenida Santos Dumont, porém as tendências centrais são mais próximas do erro zero e as dispersões são ligeiramente menores, como é possível observar nas figuras 36 e 37 a seguir.

Figura 36 - Distribuição dos erros dos instantes de travessia para a avenida Desembargador Moreira.



Fonte: autoria própria.

Figura 37 - Distribuição dos erros de atraso e tempo de travessia para a avenida Desembargador Moreira.



Fonte: autoria própria.

Outra possível fonte de erro entre os dados de travessia obtidos pelo método manual e os obtidos pelo método automatizado é a maior subjetividade e complexidade da coleta manual de travessias em detrimento da coleta automatizada, uma vez que a primeira engloba, por exemplo, a avaliação das intenções de travessia (pedestres que avaliam a possibilidade de travessia antes de chegar na zona de espera) e a avaliação das desistências de travessia (pedestres que começaram a atravessar, mas desistem e voltam a esperar), enquanto a segunda avalia apenas a posição espacial dos pedestres.

5 CONCLUSÃO

As contagens classificatórias de passagem e conversão veicular obtiveram precisão e revocação totais superiores à 96,9%, mostrando que a ferramenta de extração de trajetórias é capaz de manter uma boa constância de contagem. Quanto à determinação do tempo que os veículos levam para percorrer a faixa de pedestres, obteve-se erros médios na casa do centésimo de segundo para todas as classes de usuários analisados, o que é um resultado promissor tendo em vista que o vídeo utilizado na validação da ferramenta possui uma taxa média de 10 quadros por segundo, o que permite a coleta de instantes com precisão de apenas 0,1 segundos.

Já em relação às contagens de travessias de pedestres, foram obtidos valores de precisão e revocação totais acima de 93% em ambas as pistas analisadas, novamente mostrando uma certa robustez da ferramenta. Porém, tal robustez é apenas verificada em travessias individuais, sendo que em situações de aglomeração de pedestres não se atingiu um desempenho aceitável segundo avaliações qualitativas. Quanto às estimativas do atraso dos pedestres, o método automatizado retornou valores que acompanham a tendência central observada na coleta manual, porém consideravelmente imprecisos, com erros variando de -10 segundos até 4 segundos, desconsiderando alguns pontos fora da curva. Já para as estimativas do tempo de travessia, foram observados erros menores, entre 2 e 5 segundos, mas com uma tendência central abaixo da observada nos dados da coleta manual. No entanto, não é seguro apontar que tais erros são devido à má qualidade das trajetórias coletadas pela ferramenta, visto que o método automático de coleta de travessias é consideravelmente simplista em relação ao método utilizado na coleta manual.

Em vista disso, se nota a necessidade de maiores refinamentos da ferramenta de extração de trajetória, principalmente com foco na diminuição das trocas e perdas de identidades que foram frequentemente observadas nas análises dos resultados, sobretudo entre os pedestres. Novos esforços de treinamento do modelo de detecção também são bem-vindos, buscando diminuir as falhas de detecção e as confusões de classificação observadas. Como sugestões para trabalhos futuros, aponta-se: (1) o retreinamento do modelo de detecção com dados em maior quantidade e melhor balanceados, bem como adicionando mais exemplos de pedestres em situações de aglomeração e maior diversidade de planos de fundo; (2) realização do retreinamento do modelo de reidentificação presente no algoritmo de rastreio, com o fito de melhorar a reidentificação de objetos após oclusão e diminuir as trocas e perdas de identidade; (3) separar o rastreio de pedestres do rastreio de veículos, visto a grande discrepância de movimentos entre esses tipos de usuários e o grande foco que a comunidade de visão

computacional dá para o problema de rastreio de pedestres em detrimento do rastreio de outros tipos de objetos, o que pode trazer grandes melhorias para a coleta das trajetórias dos pedestres e, consequentemente, dos veículos; e (4) desenvolver um método mais robusto de coleta automática de variáveis de travessias de pedestres utilizando as trajetórias extraídas. Os três primeiros itens acima podem ajudar na problemática observada quanto a aglomeração de pedestres, porém sugestões específicas para tal, são: (1) uso/desenvolvimento de algoritmos de rastreio específicos para lidar com o movimento e a interação entre pedestres; (2) uso de mais câmeras que capturam a mesma cena de diferentes perspectivas, de modo a ter dados de outras câmeras em caso de oclusão de pedestres; e (3) uso de vídeos com maior qualidade e que melhor enquadram as regiões de passeio e travessia de pedestres.

REFERÊNCIAS

AGERHOLM, N. *et al*. **Road user behaviour analyses based on video detections**: status and best practice examples from the RUBA software. Proceedings of the 24th ITS World Congress, Montreal, n. 24, p. 1-10, 2017. Disponível em: https://vbn.aau.dk/files/273569946/Road_user_behaviour_analyses_based_on_video_detections_Status_and_best_practice_examples_from_the_RUBA_software.pdf. Acesso em: 13 nov. 2023.

ALVER, Y. *et al*. **Evaluation of pedestrian critical gap and crossing speed at midblock crossing using image processing**. Accident Analysis and Prevention, v. 156, 2021. DOI 10.1016/j.aap.2021.106127. Disponível em: https://doi.org/10.1016/j.aap.2021.106127. Acesso em: 9 nov. 2023.

BERNARDIN, K.; STIEFELHAGEN, R. **Evaluating multiple object tracking performance**: the CLEAR MOT metrics. EURASIP Journal on Image and Video Processing, 2008. DOI 10.1155/2008/246309. Disponível em: https://jivp-eurasipjournals.springeropen.com/counter/pdf/10.1155/2008/246309.pdf. Acesso em: 9 nov. 2023.

BEWLEY, Alex *et al*. **Simple online and realtime tracking**. IEEE International Conference on Image Processing (ICIP), p. 3464-3468, Arizona, 2016. DOI 10.1109/ICIP.2016.7533003. Disponível em: https://arxiv.org/pdf/1602.00763.pdf. Acesso em: 27 jun. 2023.

BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. **YOLOv4**: Optimal Speed and Accuracy of Object Detection. arXiv, 2020. Disponível em: https://arxiv.org/pdf/2004.10934.pdf. Acesso em: 27 dez. 2022.

BOGO, Rudinei Luiz; GRAMANI, Liliana Madalena; KAVISKI, Eloy. **Modelagem computacional do tráfego de veículos pela teoria microscópica**. Revista Brasileira de Ensino de Física, v. 37, 2015. DOI 10.1590/S1806-11173711601. Disponível em: https://www.scielo.br/j/rbef/a/pyhFt4tYLQBL6RZvBbmvTNf/?format=pdf&lang=pt. Acesso em: 9 nov. 2023.

CARDOSO, Ronaldo. [Veículos e pedestres trafegando em uma interseção semaforizada]. 2018. 1 fotografia digital, color. Disponível em: https://www.autoescolaonline.net/wp-content/uploads/2018/08/post-veiculo-linha-reta.jpg. Acesso em: 14 nov. 2023.

CASTRO JUNIOR, F. A. B. de; CASTRO NETO, M. M. de; CUNTO, F. J. C. **Análise do atraso e da brecha aceita dos pedestres em travessias semaforizadas**: um estudo na cidade de Fortaleza utilizando técnicas de visão computacional baseadas em Deep Learning. Congresso de Pesquisa e Ensino em Transportes, 35. ed., 2021. Disponível em: https://www.anpet.org.br/anais35/documentos/2021/Tr%C3%A1fego%20Urbano%20e%20Rodovi%C3%A1rio/Tr%C3%A1fego%20Urbano/2_164_AC.pdf. Acesso em: 14 nov. 2023.

DENDORFER, Patrick *et al*. **MOT20**: A benchmark for multi object tracking in crowded scenes. arXiv, 2020. Disponível em: https://arxiv.org/pdf/2003.09003.pdf. Acesso em: 9 nov. 2023.

DU, Yunhao *et al*. **StrongSORT**: Make DeepSORT Great Again. IEEE Transactions on Multimedia, 2023. DOI 10.1109/TMM.2023.3240881. Disponível em: https://arxiv.org/pdf/2202.13514.pdf. Acesso em: 24 mar. 2023.

EVERINGHAM, Mark *et al*. **The PASCAL Visual Object Classes (VOC) Challenge**. International Journal of Computer Vision, v. 88, n. 2, p. 303-338, 2010. Disponível em: https://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv_voc09.pdf. Acesso em: 9 nov. 2023.

FERRYMAN, J.; ELLIS, A. **PETS2010**: Dataset and Challenge. IEEE International Conference on Advanced Video and Signal Based Surveillance, 7. ed., p. 143-150, Boston, 2010. DOI 10.1109/AVSS.2010.90. Disponível em: https://projet.liris.cnrs.fr/imagine/pub/proceedings/AVSS-2010/data/4264a143.pdf. Acesso em: 9 nov. 2023.

GIRSHICK, Ross *et al*. **Rich feature hierarchies for accurate object detection and semantic segmentation**. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 580-587, Ohio, 2014. DOI 10.1109/CVPR.2014.81. Disponível em: https://arxiv.org/pdf/1311.2524.pdf. Acesso em: 9 nov. 2023.

GIRSHICK, Ross. **Fast R-CNN**. IEEE International Conference on Computer Vision (ICCV), p. 1440-1448, Santiago, 2015. DOI 10.1109/ICCV.2015.169. Disponível em: https://arxiv.org/pdf/1504.08083.pdf. Acesso em: 9 nov. 2023.

HUSSEIN, Mohamed *et al*. **Automated Pedestrian Safety Analysis at a Signalized Intersection in New York City**: Automated Data Extraction for Safety Diagnosis and Behavioral Study. Transportation Research Record, v. 2519, n. 1, p. 17-27, 2019. DOI 10.3141/2519-03. Disponível em: https://www.researchgate.net/publication/296690006_Automated_Pedestrian_Safety_Analysis_at_a_Signalized_Intersection_in_New_York_City. Acesso em: 10 nov. 2023.

ISMAIL, Karim *et al*. **Automated Analysis of Pedestrian-Vehicle Conflicts Using Video Data**. Transportation Research Record, v. 2140, n. 1, p. 44-54, 2009. DOI 10.3141/2140-05. Disponível em: https://www.researchgate.net/publication/228679115_Automated_Analysis_of_Pedestrian-Vehicle_Conflicts_Using_Video_Data. Acesso em: 10 nov. 2023.

LEAL-TAIXÉ, Laura *et al*. **MOTChallenge 2015**: Towards a Benchmark for Multi-Target Tracking. arXiv, 2015. Disponível em: https://arxiv.org/pdf/1504.01942.pdf. Acesso em: 9 nov. 2023.

LIN, Tsung-Yi *et al*. **Focal Loss for Dense Object Detection**. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 42, n. 2, p. 318-327, 2020. DOI 10.1109/TPAMI.2018.2858826. Disponível em: https://arxiv.org/pdf/1708.02002.pdf. Acesso em: 9 nov. 2023.

LUITEN, Jonathon *et al*. **HOTA**: A Higher Order Metric for Evaluating Multi-Object Tracking. International Journal of Computer Vision, v. 129, p. 548–578, 2021. DOI 10.1007/s11263-020-01375-2. Disponível em: https://link.springer.com/content/pdf/10.1007/s11263-020-01375-2.pdf. Acesso em: 14 nov. 2023.

LUO, Wenhan *et al*. **Multiple object tracking**: a literature review. Artificial Intelligence, v. 293, 2021. DOI 10.1016/j.artint.2020.103448. Disponível em: https://arxiv.org/pdf/1409.7618.pdf. Acesso em: 14 nov. 2023.

MILAN, Anton *et al*. **MOT16**: A Benchmark for Multi-Object Tracking. arXiv, 2016. Disponível em: https://arxiv.org/pdf/1603.00831.pdf. Acesso em: 9 nov. 2023.

REDMON, Joseph *et al*. **You Only Look Once**: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 779-788, Nevada, 2016. DOI 10.1109/CVPR.2016.91. Disponível em: https://arxiv.org/pdf/1506.02640.pdf. Acesso em: 21 dez. 2022.

REDMON, Joseph; FARHADI, Ali. **YOLO9000**: Better, Faster, Stronger. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 6517-6525, Havaí, 2017. DOI 10.1109/CVPR.2017.690. Disponível em: https://arxiv.org/pdf/1612.08242.pdf. Acesso em: 21 dez. 2022.

REDMON, Joseph; FARHADI, Ali. **YOLOv3**: An Incremental Improvement. arXiv, 2018. Disponível em: https://arxiv.org/pdf/1804.02767.pdf. Acesso em: 21 dez. 2023.

REN, Shaoqing *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, 2017. DOI 10.1109/TPAMI.2016.2577031. Disponível em: https://arxiv.org/pdf/1506.01497.pdf. Acesso em: 14 nov. 2023.

RISTANI, Ergys *et al*. **Performance Measures and a Data Set for Multi-target, Multi-camera Tracking**. In: HUA, Gang (ed.); JÉGOU, Hervé (ed.). Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II, ed. 1, Amsterdam, Springer Cham, 2016. *E-book*. p. 17-35. DOI 10.1007/978-3-319-48881-3. Disponível em: https://storage.googleapis.com/sgw-extras/zip/2016/978-3-319-48881-3.zip. Acesso em: 14 nov. 2023.

SAUNIER, Nicolas; SAYED, Tarek. **Automated Analysis of Road Safety with Video Data**. Transportation Research Record, v. 2019, n. 1, p. 57-64, 2007. DOI 10.3141/2019-08. Disponível em: https://www.researchgate.net/publication/234004923_Automated_Road_Safety_Analysis_Using_Video_Data. Acesso em: 9 nov. 2023.

SMITH, Kevin *et al*. **Evaluating Multi-Object Tracking**. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), p. 36-36, California, 2005. DOI 10.1109/CVPR.2005.453. Disponível em: https://www.researchgate.net/publication/4207406_Evaluating_Multi-Object_Tracking. Acesso em: 9 nov. 2023.

SUN, Zongyuan *et al*. **Vision-Based Traffic Conflict Detection Using Trajectory Learning and Prediction**. IEEE Access, v. 9, p. 34558-34569, 2021. DOI 10.1109/ACCESS.2021.3061266. Disponível em: https://www.researchgate.net/publication/349515511_Vision-Based_Traffic_Conflict_Detection_Using_Trajectory_Learning_and_Prediction. Acesso em: 9 nov. 2023.

WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. **YOLOv7**: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv, 2022. Disponível em: https://arxiv.org/pdf/2207.02696.pdf. Acesso em: 21 dez. 2022.

WANG, Chien-Yao; YEH, I-Hau; LIAO, Hong-Yuan Mark. **You Only Learn One Representation**: Unified Network for Multiple Tasks. arXiv, 2015. Disponível em: https://arxiv.org/pdf/2105.04206.pdf. Acesso em: 9 nov. 2023.

WOJKE, Nicolai; BEWLEY, Alex; PAULUS, Dietrich. **Simple Online and Realtime Tracking with a Deep Association Metric**. IEEE International Conference on Image Processing (ICIP), p. 3645-3649, Beijing, 2017. DOI 10.1109/ICIP.2017.8296962. Disponível em: https://arxiv.org/pdf/1703.07402.pdf. Acesso em: 14 nov. 2023.

ZHANG, Shile. **Prediction of Pedestrian Crossing Intentions at Intersections Based on Long Short-Term Memory Recurrent Neural Network**. Transportation Research Record, v. 2674, n. 4, p. 57-65, 2020. DOI 10.1177/0361198120912422. Disponível em: https://shilezhang.github.io/files/paper1.pdf. Acesso em: 9 nov. 2023.

ZHOU, Kaiyang *et al*. **Omni-Scale Feature Learning for Person Re-Identification**. IEEE/CVF International Conference on Computer Vision (ICCV), p. 3701-3711, Coreia do Sul, 2019. DOI 10.1109/ICCV.2019.00380. Disponível em: https://arxiv.org/pdf/1905.00953.pdf. Acesso em: 9 nov. 2023.