UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA URBANA

TATIANE FERREIRA OLIVATTO

IDENTIFICAÇÃO AUTOMÁTICA DE RAMPAS DE ACESSIBILIDADE APOIADA POR VISÃO COMPUTACIONAL A PARTIR DE IMAGENS PANORÂMICAS *STREET-LEVEL*

São Carlos - SP
2021

TATIANE FERREIRA OLIVATTO

IDENTIFICAÇÃO AUTOMÁTICA DE RAMPAS DE ACESSIBILIDADE APOIADA POR VISÃO COMPUTACIONAL A PARTIR DE IMAGENS PANORÂMICAS *STREET-LEVEL*

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Urbana da Universidade Federal de São Carlos como parte dos requisitos para a obtenção do título de Mestre em Engenharia Urbana.

Orientador: Prof. Dr. Edson Augusto Melanda

São Carlos - SP
2021

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia Urbana

Folha de Aprovação

Defesa de Dissertação de Mestrado da candidata Tatiane Ferreira Olivatto, realizada em 17/08/2021.

Comissão Julgadora:

Prof. Dr. Edson Augusto Melanda (UFSCar)

Prof. Dr. Elcio Hideiti Shiguemori (INPE)

Prof. Dr. André Luiz Barbosa Nunes da Cunha (USP)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia Urbana.

Agradecimentos

Em primeiro lugar, ao Prof. Dr. Edson Augusto Melanda, pela orientação, dedicação e pelos ensinamentos que transcendem os limites da Universidade. Obrigada por acreditar em mim e por me motivar nos momentos de desânimo. Por toda a paciência, empenho e suporte neste trabalho e em todos aqueles que realizamos em parceria durante o desenvolvimento do mestrado.

Aos colegas do grupo de pesquisa João Mateus Marão Domingues, Bruno Joaquim Lima, Vagner Serikawa, Felipe Facci Inguaggiato e Fábio Noel Stanganini pelas longas horas de discussão e à Larissa Fernanda pelas longas horas de rotulação de imagens.

Aos membros da banca examinadora, Prof. Dr. André Luiz Barbosa Nunes da Cunha e Prof. Dr. Elcio Hideiti Shiguemori, que tão gentilmente aceitaram participar e colaborar com esta dissertação, bem como ao membro da banca de qualificação Prof. Dr. Antonio Maria Garcia Tommaselli pelas sugestões apresentadas na ocasião.

Aos amigos, professores e colaboradores do PPGEU, em especial ao secretário Alex Rogério Silva, por ser sempre prestativo.

À toda minha família, em especial ao meu marido Wilson, pela paciência nos momentos de dificuldade com as ferramentas computacionais, e à minha irmã Talita, por todo suporte emocional e revisão textual.

Por fim, ao apoio financeiro ofertado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - Código de Financiamento 001 (Nº do Processo 8882.426640/2019-01).

"Para as pessoas sem deficiência a tecnologia torna as coisas mais fáceis. Para as pessoas com deficiência, a tecnologia torna as coisas possíveis".

(Radabaugh, 1993)

Resumo

Apesar da previsão legal de acessibilidade nos espaços públicos urbanos, a maioria das calçadas no Brasil não são dotadas de rampas de acessibilidade. Para pessoas com mobilidade reduzida, em especial usuários de cadeiras de rodas, tanto a ausência de rampas de acessibilidade quanto o desconhecimento sobre sua presença ou não em determinado local, restringe a mobilidade urbana independente e, muitas vezes, desestimula a circulação destes agentes, ocasionando exclusão no processo de democratização dos espaços urbanos. A falta de informação relacionada à localização destas infraestruturas se deve, principalmente, a disponibilidade reduzida de bancos de dados que as contemple. Apesar de iniciativas para mapeá-las, muitas metodologias se tornam tecnicamente ou economicamente inviáveis, como é o caso do uso de imagens de satélite e o levantamento de campo. Paralelamente, o uso de imagens ao nível do solo aliadas às técnicas de inteligência artificial, como visão computacional e redes neurais, vêm sendo amplamente empregadas na coleta de dados acerca de infraestruturas urbanas. Buscando uma alternativa para esta questão, o objetivo deste trabalho foi utilizar imagens ao nível do solo para construir um banco de imagens rotuladas e viabilizar a identificação de rampas de acessibilidade em calçadas por meio de detecção de objetos. Para tal, o trabalho teve início pela obtenção de panorâmicas do Google Street View de modo estratégico, a partir de informações censitárias sobre ocorrência de rampas e estratos populacionais. As rampas de acessibilidade identificadas nestas imagens foram rotuladas manualmente para a construção do referido banco de imagens e utilizadas no treinamento e validação de uma rede neural convolucional do detector de objetos *YOLO*v4. A partir daí, foram realizados treinamentos variando-se as técnicas de pré-processamento e os parâmetros de treinamento, com isso, verificou-se que o emprego de *Tiling* e o uso de pesos pré-treinados resultaram numa precisão média de validação da ordem de 65%. Os testes indicaram que a rede detecta objetos com precisão média de 85%, identificando cerca de 77% dos objetos do conjunto de teste. Os pesos convolucionais resultantes permitiram a detecção de rampas com *designs* variados, entretanto, a rede neural apresentou menor desempenho na avaliação de rampas parcialmente oclusas ou em mau estado de conservação. Durante os procedimentos experimentais, observou-se que a adoção do *threshold* de 65% para a precisão de detecção levou a um melhor equilíbrio entre o número de detecções corretas e incorretas, assim como para possibilitar a identificação de objetos proporcionalmente pequenos. O banco de imagens rotuladas elaborado neste trabalho é uma contribuição relevante considerando que não há atualmente um banco similar para rampas de acessibilidade que inclua municípios brasileiros. Além disso, o detector treinado para identificar as rampas nos passeios públicos se mostrou eficaz, potencializando aplicações futuras que possam envolver o mapeamento destas infraestruturas, a inclusão de outras classes e o desenvolvimento de aplicações mais elaboradas.

Palavras-chave: Acessibilidade. Rampa de Acessibilidade. Visão Computacional. Detecção de Objeto. *YOLO*v4. Google Street View. Imagem Panorâmica. *Tiling*.

Abstract

Despite the legal provision for accessibility in urban public spaces, most sidewalks in Brazil are not endorsed with accessibility curb ramps. For people with reduced mobility, especially wheelchair users, both the absence of accessibility ramps and the lack of knowledge about their presence or not in a given location, restricts independent urban mobility and often discourages the movement of these agents, causing exclusion in the process of urban spaces democratization. The lack of information related to the location of these infrastructures is mainly due to the reduced availability of databases that cover them. Despite initiatives to map them, many methodologies become technically or economically unfeasible, such as the use of satellite images and field surveys. At the same time, the use of street-level images combined with artificial intelligence techniques, such as computer vision and neural networks, have been widely used to collect data on urban infrastructure. Seeking an alternative to this issue, the objective of this work was to use street-level images to build a labeled image dataset and enable the identification of accessibility curb ramps on sidewalks through object detection. To this end, the work began by obtaining Google Street View panoramas in a strategic way, based on census information on the occurrence of ramps and population strata. The accessibility curb ramps identified in these images were manually labeled for the construction of that image database and used in the training and validation of a convolutional neural network of the YOLOv4 object detector. From then on, training was performed varying pre-processing techniques and training parameters, thus, it was found that the use of Tiling and the use of pre-trained weights resulted in an average validation accuracy of the order of 65%. The main limitation of the research occurred in this stage, due to computational memory limitations for processing. Tests indicated that the network detects objects with an average accuracy of 85%, identifying about 77% of the objects in the test set. The resulting convolutional weights allowed ramps detection with varied designs, however, the neural network had a lower performance in the evaluation of partially occluded slopes or in a poor state of conservation. During the experimental procedures, it was observed that the adoption of the 65% threshold for detection accuracy led to a better balance between the number of correct and incorrect detections, as well as to enable the identification of proportionally small objects. The labeled image dataset elaborated in this work is a relevant contribution, considering that there is currently no similar database for accessibility curb ramps that includes Brazilian cities. In addition, the detector trained to identify curb ramps on public sidewalks proved effective, enhancing future applications that may involve the mapping of these infrastructures, the inclusion of other classes and the development of more elaborate applications.

***Key-words**: Accessibility. Accessibility Curb Ramp. Computer Vision. Object Detection. YOLOv4. Google Street View. Panorama Image. Tiling.*

Lista de ilustrações

Lista de quadros

Lista de tabelas

Lista de abreviaturas e siglas

ACR *Acessible Curb Ramps* - Rampas de Acessibilidade. 64, 65, 68, 69

AP *Average Precision* - Precisão Média. 71, 72, 79–82, 98

API *Application Programming Interface* - Interface de Programação de Aplicativos. 61, 68

CNN *Convolutional Neural Network*. 9, 38, 43, 45, 47, 49, 50, 54–56, 60, 61, 63, 66, 68, 69, 73, 79, 80, 82, 83, 95, 97, 98

CUDA *Compute Unified Device Architecture* - Arquitetura de Dispositivo de Computação Unificada. 50, 61, 70

cuDNN CUDA *Deep Neural Network*. 50, 61, 70

CV *Computer Vision*. 8, 40–42

FN Falso Negativo. 71, 73, 83, 85, 87, 95

FP Falso Positivo. 71, 73, 83, 85–87, 95

FPN *Feature Pyramid Networks*. 44

G-CNN *Grid Convolutional Neural Network*. 46

GNSS *Global Navigation Satellite System* - Sistema Global de Navegação por Satélite. 19

GPUs *Graphics Processing Unit* - Unidade de Processamento Gráfico. 38, 50

GSV Google *Street View*. 8, 15, 21, 22, 53–58, 60–64, 67, 74, 75, 87, 95, 97, 98, 113, 114

IoU *Intersection-over-Union* - Intersecção sobre a União. 71, 79, 80, 98

OSM *OpenStreetMap*. 19, 20, 63

PANet *Path Aggregation Network*. 48, 49

PMU Plano de Mobilidade Urbana. 18, 30

PNMU Política Nacional de Mobilidade Urbana. 18, 30, 32, 64

R-CNN *Region-based Convolutional Neural Networks*. 43, 44, 55

R-FCN *Region-based Fully Convolutional Networks*. 44

ReLU *Rectified Linear Unit* - Unidade Linear Retificada. 38, 39

RNAs Redes Neurais Artificiais. 8, 34, 35, 37, 38

RPN *Region Proposal Network*. 44

SPP *Spatial Pyramid Pooling*. 43, 48, 49

SPP-net *Spatial Pyramid Pooling Networks*. 43, 44

SSD *Single-Shot Multibox Detector*. 46, 47

TICs Tecnologias da Informação e Comunicação. 18, 20, 21

VGI *Volunteered Geographic Information* - Informação Geográfica Voluntária. 19, 54

VP Verdadeiro Positivo. 71, 73, 83, 85–87, 95

YOLO *You Only Look Once*. 8, 46–50, 55, 56, 60, 61, 68, 73, 97

Sumário

1 Introdução

O aumento da população urbana e, consequentemente, a expansão das áreas urbanas vem acompanhado de desafios relacionados à qualidade de vida, infraestrutura e mobilidade nestes ambientes. Um aspecto que ainda precisa ser superado pela maioria das cidades brasileiras é acessibilidade. Em 2010, a população brasileira era aproximadamente 190,7 milhões, sendo que 32,8 milhões (17,2%) de pessoas se declararam com limitação funcional, ou seja, que possuem alguma dificuldade para enxergar, ouvir, andar ou subir escadas e 12,7 milhões (6,7%) se declararam com deficiência, ou seja, total ou grande incapacidade para enxergar, ouvir, andar, subir escadas ou alguma deficiência intelectual ou mental (IBGE, 2010a).

O documento “Boas Práticas de Desenvolvimento Urbano Acessível” publicado pela ONU (2016) recomenda ações voltadas à melhoria da mobilidade nos ambientes urbanos, incluindo aspectos que os tornem mais acessíveis. Esta ideia remete à definição de mobilidade urbana proposta por Lévy (2001), na qual a mesma é entendida como “relação social ligada à mudança de lugar, isto é, como um conjunto de modalidades pelas quais os membros de uma sociedade tratam a possibilidade de eles próprios ou outros ocuparem sucessivamente vários lugares”. Consequentemente, ao compreender que uma parcela dos membros da sociedade necessita de acessibilidade para ocupar ambientes, a mobilidade urbana para estes cidadãos também está sujeita à acessibilidade (MACHADO; LIMA, 2015).

De acordo com a NBR 9050 de 2004, acessibilidade é a “possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos” (ABNT, 2001). Logo, a premissa de autonomia é uma característica que se destaca nesta definição, a qual também consta no documento da ONU ao ressaltar a importância do entendimento de boas práticas de acessibilidade não apenas como ferramenta inclusiva, mas também como ferramenta para proporcionar um viver de forma independente (ONU, 2016). Nesta mesma vertente de pensamento, da perspectiva da cidade, a Constituição Federal de 1988 reforça a função social da mesma, visando oferecer os benefícios da urbanização a todos os habitantes, sem preconceitos e discriminação (BRASIL, 1988). Logo, promover acessibilidade nos espaços urbanos não é apenas uma questão de escolha, é uma ferramenta de promoção de cidadania assegurada legalmente.

Contudo, para pessoas com restrições de mobilidade, a acessibilidade e independência no ambiente urbano está muitas vezes atrelada à existência de infraestrutura. Um exemplo é o que trata o Decreto nº 5.296/2004, proferindo que toda a frota de transporte

coletivo deveria ser acessível até 2014 (BRASIL, 2004). Entretanto, dados divulgados pelo IBGE em 2017 mostraram que das 1.679 cidades que possuem transporte coletivo, somente em 197 cidades (11,7%) os ônibus eram adaptados. Em 820 cidades (48,8%), a frota era apenas parcialmente adaptada e em 662 (39,4%) não há nenhum ônibus acessível (IBGE, 2017).

Os dados apresentados refletem um cenário desanimador, mas podem indicar que a situação em relação a outros modais de transporte é ainda pior se considerarmos que, durante muitos anos, os estudos acadêmicos relacionados à acessibilidade em áreas urbanas focavam predominantemente na engenharia de tráfego ou planejamento de transportes. No que tange pessoas com restrições de mobilidade, muitos municípios passaram a considerá-las em suas políticas públicas de mobilidade urbana somente após a elaboração da Política Nacional de Mobilidade Urbana em 2012 (GUIMARÃES RAFAELLA OLIVEIRA; SANTOS, 2018). Uma infraestrutura urbana essencial para estes usuários, especificamente para usuários de cadeiras de rodas, é uma rede de pedestres equipada com rampas de acessibilidade e passeios públicos com largura mínima ideal e ausência de obstáculos e degraus (ABNT, 2001). Contrastando com esta premissa, apenas 4,7% da rede viária brasileira possui rampas para cadeirantes (IBGE, 2010a).

No Artigo 3° do Estatuto da Cidade (BRASIL, 2001) verifica-se que compete à União, dentre as atribuições da política urbana, “instituir diretrizes para desenvolvimento urbano, [...] que incluam regras de acessibilidade aos locais de uso público”. Parte destas regras de acessibilidade em espaços públicos estão descritas na NBR 9050, que trata da adequação das edificações e do mobiliário urbano (ABNT, 2001).

O Estatuto da Cidade (BRASIL, 2001) prevê ainda a elaboração de planos diretores, essencialmente, para cidades com mais de vinte mil habitantes; integrantes de regiões metropolitanas e aglomerações urbanas; inseridas em áreas de interesse turístico ou de influência de empreendimentos ou atividades com significativo impacto ambiental de âmbito regional e nacional; e incluídas no cadastro de áreas suscetíveis à deslizamentos, inundações ou outros processos correlatos. De acordo com o parágrafo 3 do Artigo 41° desta mesma legislação, estas cidades devem:

> (...) elaborar plano de rotas acessíveis, compatível com o plano diretor no qual está inserido, que disponha sobre os passeios públicos a serem implantados ou reformados pelo poder público, com vistas a garantir acessibilidade da pessoa com deficiência ou com mobilidade reduzida a todas as rotas e vias existentes, inclusive as que concentrem os focos geradores de maior circulação de pedestres, como os órgãos públicos e os locais de prestação de serviços públicos e privados de saúde, educação, assistência social, esporte, cultura, correios e telégrafos, bancos, entre outros, sempre que possível de maneira integrada com os sistemas de transporte coletivo de passageiros (BRASIL, 2001).

Paralelamente, a Política Nacional de Mobilidade Urbana (PNMU) exige o desenvolvimento do Plano de Mobilidade Urbana (PMU) para municípios acima de 20 mil habitantes, integrantes de regiões metropolitanas, integradas de desenvolvimento econômico e aglomerações urbana (com mais de 1 milhão de habitantes); e áreas de interesse turístico, inclusive cidades litorâneas (BRASIL, 2012). As cidades que desenvolverem seus PMUs precisam observar seus princípios, objetivos e diretrizes, descritos nos Artigos 5°, 7°e 24° da PNMU, dentre os quais destacam-se três tópicos relativos à acessibilidade. O primeiro deles se refere à "acessibilidade universal" sob a qual está fundamentada a PNMU, o segundo trata do objetivo de "proporcionar melhoria nas condições urbanas da população no que se refere à acessibilidade e à mobilidade" e o terceiro descreve a diretriz mínima do PMU de comtemplar "a acessibilidade para pessoas com deficiência e restrição de mobilidade" (BRASIL, 2012).

No Brasil, de acordo com o Censo 2010, a região sudeste é a que possui maior número de domicílios localizados em logradouros com acesso a rampas de acessibilidade em calçadas, sendo o estado de São Paulo com apenas 5,22% da totalidade de domicílios, Rio de Janeiro com 5,77%, Minas Gerais com 3,60% e Espírito Santo com 5,09% (IBGE, 2010a). Este cenário está longe de ser o ideal, principalmente, se forem considerados aspectos como a adequada localização das rampas, seu estado de conservação e sua conformidade com as normas regulamentadoras (SOUZA, 2019).

Além da questão de infraestrutura física, a escassez de informações sobre acessibilidade nos espaços públicos e privados faz com que pessoas com mobilidade reduzida deixem de frequentar estes locais (NETO; ROLT; ALPERSTEDT, 2018; HARA; FROEHLICH, 2015). Mesmo no contexto atual, em que tecnologias vem sendo incorporadas ao planejamento e gestão urbanas e o acesso às mesmas tem sido ampliado (ISMAGILOVA et al., 2019; SILVA FILHO, 2012), muitas delas ainda não contemplam o aspecto de acessibilidade.

1.1 Problema

Nos últimos anos, muitas cidades estão direcionando seus esforços na utilização de tecnologia e conectividade para gerenciar os desafios da mobilidade urbana. Uma tendência é a integração de Tecnologias da Informação e Comunicação (TICs) visando auxiliar a operação e planejamento das atividades de diversos setores. Estes setores podem incluir gerenciamento de tráfego e modais de transporte, monitoramento em tempo real e organização do transporte público (ISMAGILOVA et al., 2019).

A incorporação de TICs no planejamento e gestão urbana reflete a modificação do meio ambiente urbano segundo o conceito de cidades inteligentes. Segundo este conceito, as TICs se destacam pela coleta e disponibilização de dados em tempo real e escala global. Neste contexto, a disseminação dos *smartphones* foi determinante devido à facilidade

de conexão à internet e presença de diversos sensores, como câmeras e microfones. A popularização e permanência destes dispositivos se confirmou em 2013, quando a venda de *smartphones* já havia superado a venda dos celulares tidos até então como tradicionais (CARDONE et al., 2014).

Um sensor comumente acoplado à dispositivos móveis, inclusive *smartphones*, são os sensores *Global Navigation Satellite System* - Sistema Global de Navegação por Satélite (GNSS). Estes sensores viabilizam diversos instrumentos que auxiliam a melhoria da qualidade de vida nos centros urbanos. Um exemplo são serviços de roteirização e navegação que utilizam-se destes sensores para prever rotas de acordo com a necessidade do usuário e, em contrapartida, fazem uso de dados providos pelo dispositivo do próprio usuário para navegação e atualização da rota fornecida (CARDONE et al., 2014).

Estes serviços estão se tornando cada vez mais personalizados, seja para o planejamento de empresas de transporte público e de entregas, para a disponibilização de transporte privado urbano (por exemplo, Uber e 99) ou para conveniência de pedestres e motoristas em geral. Destacam-se entre os mais populares o Google Maps e Waze (SILVA et al., 2015).

Neste mesmo cenário, o surgimento de aplicativos e plataformas de informação geográfica de código aberto incentivaram uma nova geração de serviços de roteirização. Esta geração baseia-se em plataformas de *Volunteered Geographic Information* - Informação Geográfica Voluntária (VGI) – em tradução livre Informação Geográfica Voluntária – que, como o próprio nome sugere, reúnem contribuições de voluntários que coletam dados geográficos de forma colaborativa (MOBASHERI et al., 2018; ELWOOD; GOODCHILD; SUI, 2012).

Atualmente o projeto mais difundido de VGI é o *OpenStreetMap* (OSM), agregando mais de 6 milhões de usuários registrados que contribuem com o projeto em algum nível. A maior densidade de dados mapeados encontra-se nas áreas urbanas. Nestes locais, as informações mais coletadas são referentes às vias de trânsito de veículos motorizados e seus atributos (superfície, inclinação, sentido, endereço/geocódigo e existência ou não de passeios). Estes dados são utilizados para roteirização de veículos e pedestres (NEIS; ZIELSTRA, 2014; OSM, 2020).

O *Open Source Routing Machine* é um projeto que utiliza dados do OSM para calcular rotas. Com base nos dados disponíveis atualmente na plataforma VGI, a roteirização contempla apenas veículos motorizados, não havendo possibilidade de personalização para nenhum outro modal de transporte (OSRM, 2020).

No caso da roteirização para pessoas com mobilidade reduzida, como usuários de cadeira de rodas, há uma dependência ainda maior quanto ao detalhamento dos dados mapeados. Informações adicionais relacionadas especificamente às características das

calçadas são triviais, incluindo largura, superfície, inclinação, presença de obstáculos e localização de rampas. Contudo, tanto fontes de dados comerciais como voluntárias não oferecem este tipo de detalhamento para todas as localidades (ZIPF et al., 2016).

Apesar de poucos estudos se dedicarem à obtenção e tratamento deste tipo de informação (CAPINERI et al., 2016), de acordo Neis e Zielstra (2014) pesquisas com foco em especificações e aplicações de rotas para pessoas com mobilidade reduzida – incluindo usuários de cadeiras de rodas – têm crescido nos últimos anos.

No Brasil, a principal limitação para viabilizar o funcionamento de serviços de roteirização personalizado à pessoas com mobilidade reduzida e pedestres de modo geral é a ausência do mapeamento das calçadas e rampas de acessibilidade (MOBILIZE BRASIL, 2013). De acordo com MOBILIZE BRASIL (2013), as calçadas são responsabilidade dos proprietários, e não da administração municipal, não sendo portanto foco de monitoramento para manutenção e, consequentemente, não sendo alvo de mapeamento.

Na Figura 1 é possível verificar o exemplo de uma localidade que, mesmo apresentando maior ocorrência de rampas de acessibilidade, as mesmas não foram mapeadas no OSM.

Figura 1 – Percentual de ocorrência de rampas para cadeirante nas proximidades das esquinas, por grade de 1Km (à esquerda) e identificação das vias mapeadas sem informações relacionadas à calçada e rampas (à direita).
Fonte: Adaptado de IBGE (2010b) e OSMSURROUND (2020).

1.2 Motivação

Cidades mais desenvolvidas tem se aproveitado de TICs e inteligência artificial para conhecer melhor suas dinâmicas por diferentes perspectivas, como monitoramento de tráfego, planejamento urbano e logístico (WANG; GUO; YANG, 2018; ISMAGILOVA et

al., 2019). Dentre as principais aplicações, mobilidade tem sido considerada prioridade por diversos governos (WANLI et al., 2018).

Por outro lado, algumas cidades ainda não se apropriaram destas tecnologias devido, principalmente, à limitação de acesso à dados (WANG; GUO; YANG, 2018). Enquanto que TICs tem sido foco de atenção devido à possibilidade de obtenção de dados em tempo real (WANLI et al., 2018), muitas cidades tem dificuldade em obter dados básicos, como por exemplo, mapeamento de infraestruturas (REZENDE, 2019; WEISS; BERNARDES; CONSONI, 2015).

Avanços em tecnologias de visão computacional e aprendizado de máquina têm propiciado poderosas ferramentas para realizar automaticamente tarefas de mapeamento antes realizadas manualmente, acelerando e sistematizando a obtenção de resultados (WANLI et al., 2018). Pode-se mencionar como exemplo tarefas de classificação do uso do solo, vetorização de feições urbanas e mapeamento de locais de acidentes de trânsito (ALHAFNI et al., 2019; WANLI et al., 2018; DIAS et al., 2016).

No caso do mapeamento de feições de mobilidade para pedestres, esta atividade é tradicionalmente conduzida em levantamentos de campo ou, mais recentemente, via aplicativos colaborativos – o que ainda requer presença física (HARA; FROEHLICH, 2015). Contudo, observa-se surgimento de alguns estudos que se apropriam de visão computacional e imagens ao nível do solo para contemplar esta tarefa, sendo que a maioria dos estudos utiliza-se de imagens disponibilizadas pelo Google *Street View* (GSV) (CHEN et al., 2020; LIU et al., 2020; HARA; FROEHLICH, 2015).

As imagens disponibilizadas pelo GSV viabilizariam a identificação de calçadas, inclusive de rampas de acessibilidade, sem a necessidade de coleta de dados em campo ou da obtenção de imagens de alta resolução. Portanto, a utilização destas imagens ao nível do solo juntamente à sistemas de inteligência artificial representam uma possibilidade para viabilizar o mapeamento de rampas de acessibilidade, proporcionando economia de tempo e recursos.

1.3 Justificativa

Este trabalho se justifica pela viabilização do mapeamento de feições de acessibilidade – especificamente rampas de acessibilidade – sem a necessidade de levantamento de campo ou aquisição de imagem de alta resolução, por tratar do uso de imagens disponíveis na em plataforma aberta, e sem a necessidade de identificação manual de cada feição, uma vez que esta tarefa é realizada por Aprendizado de Máquina.

Este trabalho investiga também a possibilidade de disponibilizar uma base de dados treinada para identificar rampas de acessibilidade, para que a mesma possa ser utilizada

em futuros mapeamentos, além da implementação do sistema sem aquisição de poder de processamento computacional.

1.4 Objetivos

O objetivo principal deste trabalho é viabilizar a identificação de rampas de acessibilidade em passeios públicos a partir do emprego de técnicas de detecção de objetos em panorâmicas do GSV. Visando atingir este objetivo primário, os objetivos específicos deste trabalho são:

I. Criar uma base de imagens do GSV com rótulos de rampas de acessibilidade de cidades brasileiras;

II. Treinar e obter uma uma rede neural artificial, especificamente uma rede neural convolucional, capaz de identificar rampas de acessibilidade em imagens do GSV;

III. Validar e verificar o desempenho da base treinada para identificar rampas.

1.5 Estrutura geral

Este trabalho encontra-se organizado em sete capítulos, sendo esta introdução o primeiro deles. O Capítulo 2 traz uma visão geral sobre as rampas de acessibilidade nas calçadas brasileiras, mais especificamente no estado de São Paulo. O Capítulo 3 é composto pela revisão de literatura que relaciona os principais fundamentos teóricos referentes à detecção das rampas de acessibilidade. O Capítulo 4, também de revisão de literatura, focou nas características e publicações relacionadas à imagens panorâmicas do Google Street View. O Capítulo 5 descreve os materiais e métodos utilizados neste trabalho e o Capítulo 6 expõe os resultados e discussões. Por fim apresenta-se as conclusões no Capítulo 7.

2 Contextualização

Considerando que um dos objetivos deste trabalho é a disponibilização de uma base treinada para identificar rampas de acessibilidade, faz-se necessário um estudo do panorama geral quantitativo destas no Brasil afim de embasar um recorte de amostragem de imagens a ser empregado nos procedimentos metodológicos deste trabalho.

De acordo com o Censo 2010, a região sudeste brasileira é a que possui o maior número de domicílios localizados em logradouros com acesso a rampas de acessibilidade em calçadas, mesmo assim, apenas 5,22% do estado de São Paulo, 5,77% do Rio de Janeiro, 3,60% de Minas Gerais e 5,09% do Espírito Santo estão nesta situação privilegiada (IBGE, 2010a).Este cenário está longe de ser o ideal, principalmente, se forem considerados outros aspectos como localização adequada das rampas, seu estado de conservação e sua conformidade com norma regulamentadora (SOUZA, 2019).

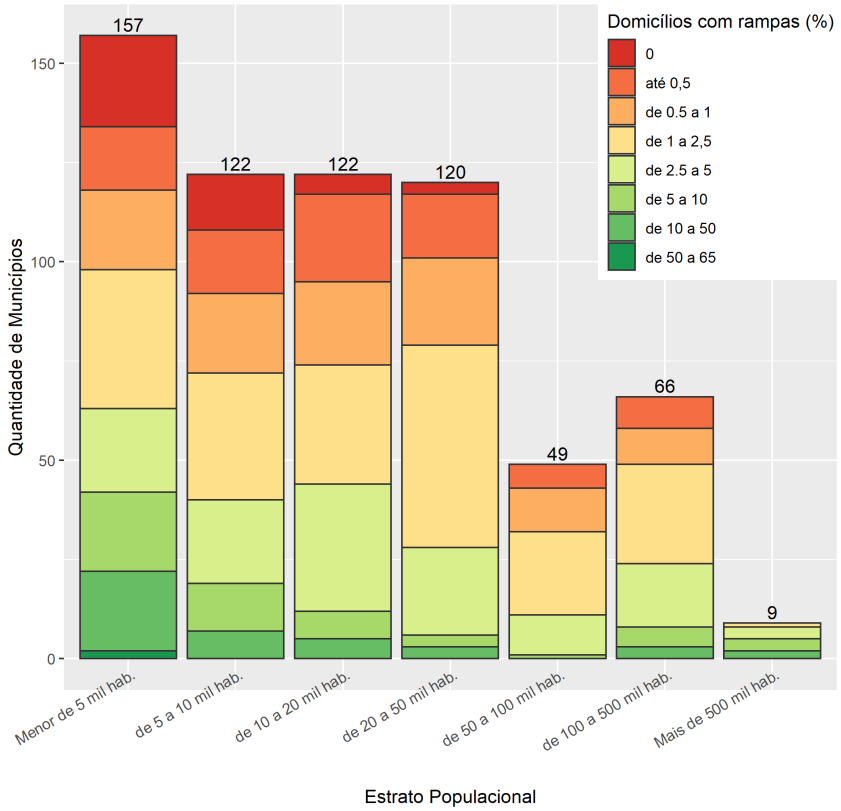
Considerando as porcentagens de domicílios com rampas em calçadas e a possibilidade de cobrir uma extensão territorial maior, portanto com maior possibilidade de disponibilização de imagens do *Google Street View*, este trabalho utilizará como recorte de estudo o Estado de São Paulo.

Assim, neste capítulo será apresentado um levantamento quantitativo da situação dos municípios do estado de São Paulo em relação às rampas de acessibilidade nos passeios públicos. Para tal, foram investigadas a porcentagem de domicílios com rampas para cadeirantes na face de acesso do logradouro nas áreas urbanas de acordo com estratos populacionais[1], levantando também questões relacionados aos fatores que possam influenciar a ocorrência destas porcentagens.

2.1 Rampas de Acessibilidade no Estado de São Paulo

No estado de São Paulo está a maior capital brasileira em termos populacionais, a cidade de São Paulo, com mais de 10 milhões de habitantes. Um fato interessante é que a soma da população de todos os municípios com até 50 mil habitantes do estado, que representam 81,77% do contingente populacional total dos 645 municípios do estado, não alcança a população do município de São Paulo. Esta situação é ilustrada na Figura 2, que apresenta o quantitativo total de municípios do estado de acordo com os estratos populacionais utilizados pelo IBGE (2017).

[1] Foram utilizados os estratos populacionais adotados pelo IBGE (2017): (i) até 5 mil habitantes, (ii) de 5 a 10 mil habitantes, (iii) de 20 a 50 mil habitantes, (vi) de 50 a 100 mil habitantes, (v) de 100 a 500 mil habitantes e (vi) mais de 500 mil habitantes.



Nota: "Domicílios com rampas (%)" refere-se às faixas de porcentagens de domicílios com rampas para cadeirantes na face de acesso do logradouro, considerando apenas as áreas urbanas dos municípios.

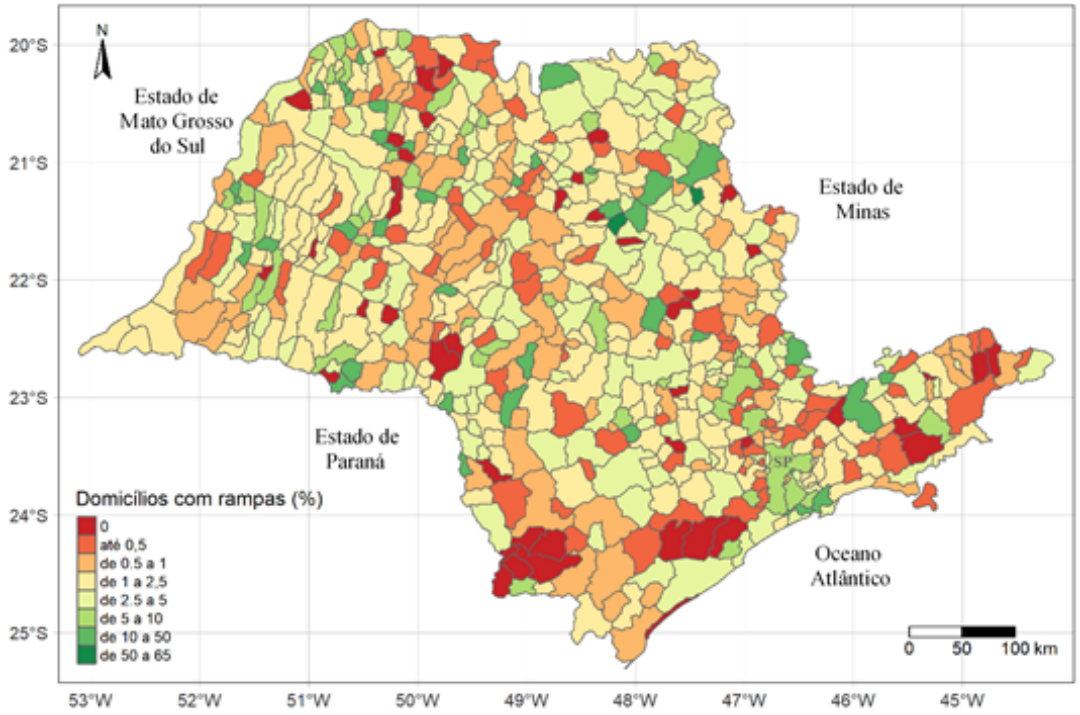
Figura 2 – Distribuição dos percentuais de domicílios com rampas para cadeirantes em cada estrato populacional, no estado de São Paulo.
Fonte: Adaptado de IBGE (2010a).

Nesta mesma figura verifica-se a distribuição quantitativa percentual de domicílios com rampas para cadeirante na face de acesso do logradouro por estrato populacional. Da figura pode-se observar que a distribuição dos dados não se apresenta numa progressão contínua. A representação escolhida visou facilitar a visualização dos dados, por exemplo, as porcentagens acima de 50%, limitando-se à 65%, foram identificadas apenas no estrato de municípios menores de 5 mil habitantes (primeira coluna).

De forma geral, é possível notar que, progressivamente, as porções referentes às porcentagens mais extremas, tanto para mais quanto para menos, diminuem conforme a população total dos municípios aumenta. É nos estratos de menor porte, até 50 mil habitantes, que soma o maior quantitativo de municípios onde não há rampas para cadeirantes (0%), ao mesmo tempo, onde há maior ocorrência das faixas percentuais entre 5 e 10% e 10 e 50%. No estrato populacional de 50 a 100 mil habitantes, as faixas referentes às porcentagens acima de 10% não constam.

Além desta abordagem quantitativa, considerando que investigar a distribuição espacial dos fenômenos constitui uma importante ferramenta para subsidiar conhecimentos

(DRUCK et al., 2004), a abordagem resultante na Figura 3 buscou compreender outros aspectos dos dados.



Nota: "Domicílios com rampas (%)" refere-se às faixas de porcentagens de domicílios com rampas para cadeirantes na face de acesso do logradouro, considerando apenas as áreas urbanas dos municípios.

Figura 3 – Mapa do percentual de domicílios com rampas para cadeirantes no estado de São Paulo.
Fonte: Adaptado de IBGE (2010a).

Considerando que o estado de São Paulo, assim como o Brasil como um todo, possui municípios com características variadas e que a estratificação populacional é apenas uma forma de construção de tipologia dos municípios brasileiros (CALVO et al., 2016), uma investigação mais aprofundada de cada estrato mostra-se relevante para que eles não sejam tratados como unidades homogêneas.

Com base nesta premissa, após a análise do comportamento geral dos dados para o estado como um todo, procedeu-se a análise para cada estrato populacional. Na sequência, os resultados referentes aos municípios com populações até 50 mil habitantes estão ilustrados na Figura 4 e aqueles referentes aos municípios com população maior que 50 mil habitantes são ilustrados na Figura 5. Para facilitar a visualização dos resultados, o esquema de cores adotados nas figuras segue o mesmo padrão daqueles das Figuras 2 e 3.

No gráfico da Figura 4a destacam-se as cidades de Santa Cruz da Esperança, com 61,35% de domicílios com rampas e Motuca com 54,00%. A partir daí, foram identificados no gráfico 7 municípios com porcentual de domicílios com rampas entre 20 e 40%. Neste estrato

populacional estão 157 municípios (vide Figura 2), sendo que os demais 148 municípios apresentam porcentagens menores do que 20%.

O gráfico da Figura 4b permite uma comparação de extremos entre as cidades de Coronel Macedo e Braúna, com aproximadamente o mesmo contingente populacional – 5.021 e 5.001 habitantes, respectivamente. Porém, Braúna possui 27,72% dos domicílios com rampa na face do logradouro, enquanto Coronel Macedo não possui nenhum (0%).

Na comparação com os gráficos de municípios com população menor que 10 mil habitantes e porcentagens superiores à 20%, o gráfico da Figura 4c, de municípios entre 10 e 20 mil habitantes, limita-se à 16,56%, no município de Itirapina, seguido por Cerqueira César, com 16,40%. Um aspecto interessante pode ser constatado ao comparar os municípios de Águas de Lindóia e Cerqueira César, que possuem população similar, porém com porcentagem de domicílios com rampas muito discrepantes – equivalente à apenas 6,43% em Águas de Lindóia.

Outra análise possível considerando as informações da Figura 4c se refere às cidades tradicionalmente turísticas. Tomemos Águas de Lindóia e Holambra, por exemplo, que possuem entre 5 e 10% de domicílios com rampas nas faces dos logradouros, e Itirapina, entre 15 e 20%. Nestes casos essa diferença não parece ser influenciada pelo contingente populacional – Itirapina, com melhor desempenho percentual, tem 15.524 habitantes e Holambra e Águas de Lindóia 8.184 e 17.111 habitantes, respectivamente.

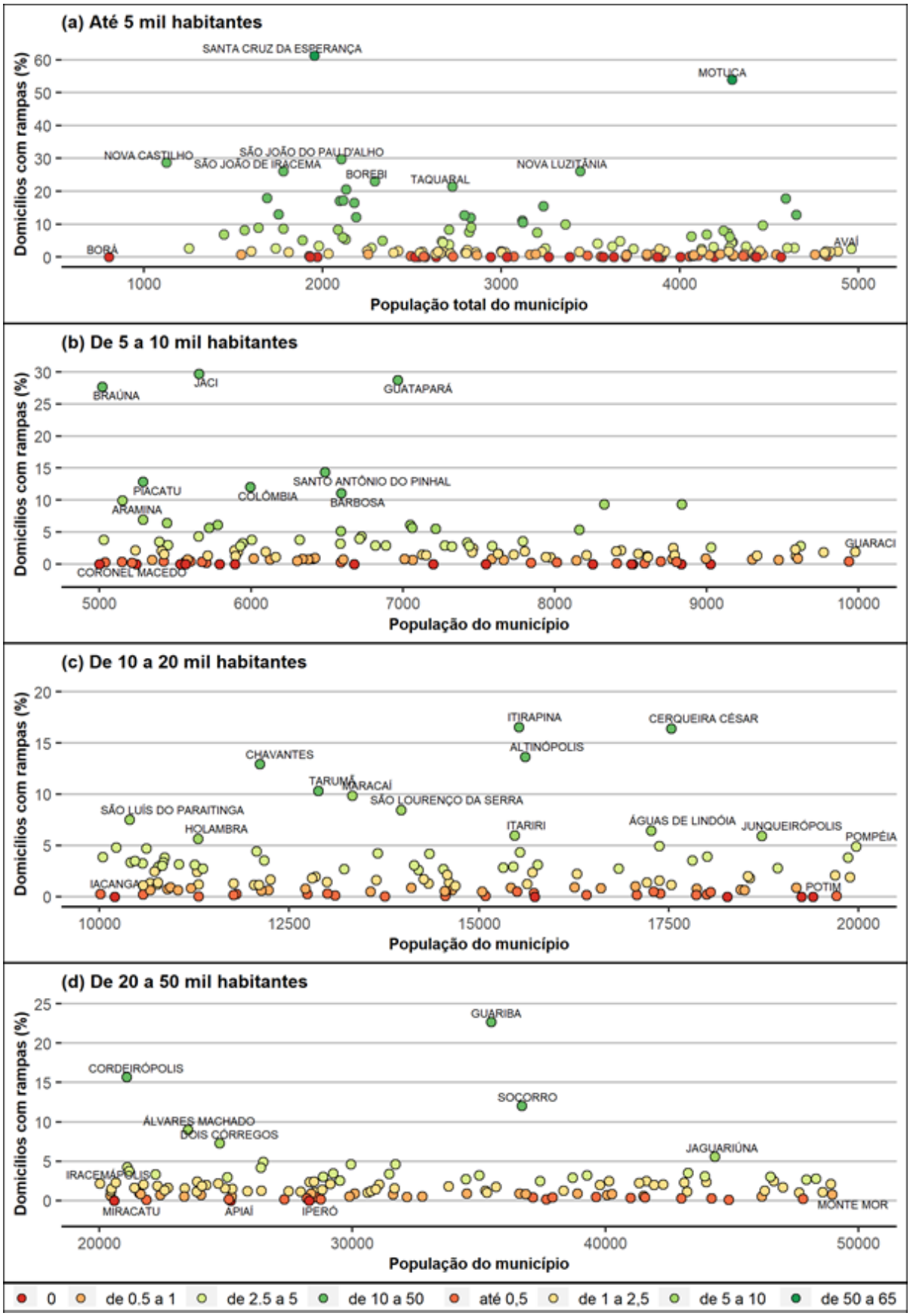
Por fim, o gráfico da Figura 4d não difere muito dos demais no que diz respeito à concentração de pontos na base, referindo-se à reduzida porcentagem de domicílios com rampas. O que percebe-se é a redução expressiva de pontos tocando o eixo x do gráfico, que se refere aos municípios com ausência de domicílios com acesso a rampas, sendo apenas 3, as cidades de Miracatu, Iperó e Apiá. Em contrapartida, também há 3 municípios com porcentagens acima de 10%: as cidades de Guariba (22,72%), Cordeirópolis (15,69%) e Socorro (12,05%).

Agora tratando das análises da Figura 5, especificamente na Figura 5a, percebemos uma redução significativa na variabilidade das porcentagens, como já previsto na Figura 2 e relativamente esperado devido à redução de municípios neste estrato. O município de Batatais, com a maior porcentagem de domicílios com rampas, 7,9%, apresenta-se como um ponto discrepante no gráfico. Os outros municípios, exceto por Mirassol com porcentagem de 4,07%, não ultrapassam 4%. Um fato a ser considerado é que neste gráfico não há município com ausência de domicílios com rampas, apesar de Arujá estar muito próximo de zero, com 0,05%.

Na Figura 5b, o segundo e o terceiro município com maior porcentagem de domicílios com rampas são litorâneos e confrontantes, são eles Santos (22,38%) e São Vicente (10,73%). Também estão entre os mais populosos do estrato populacional em questão,

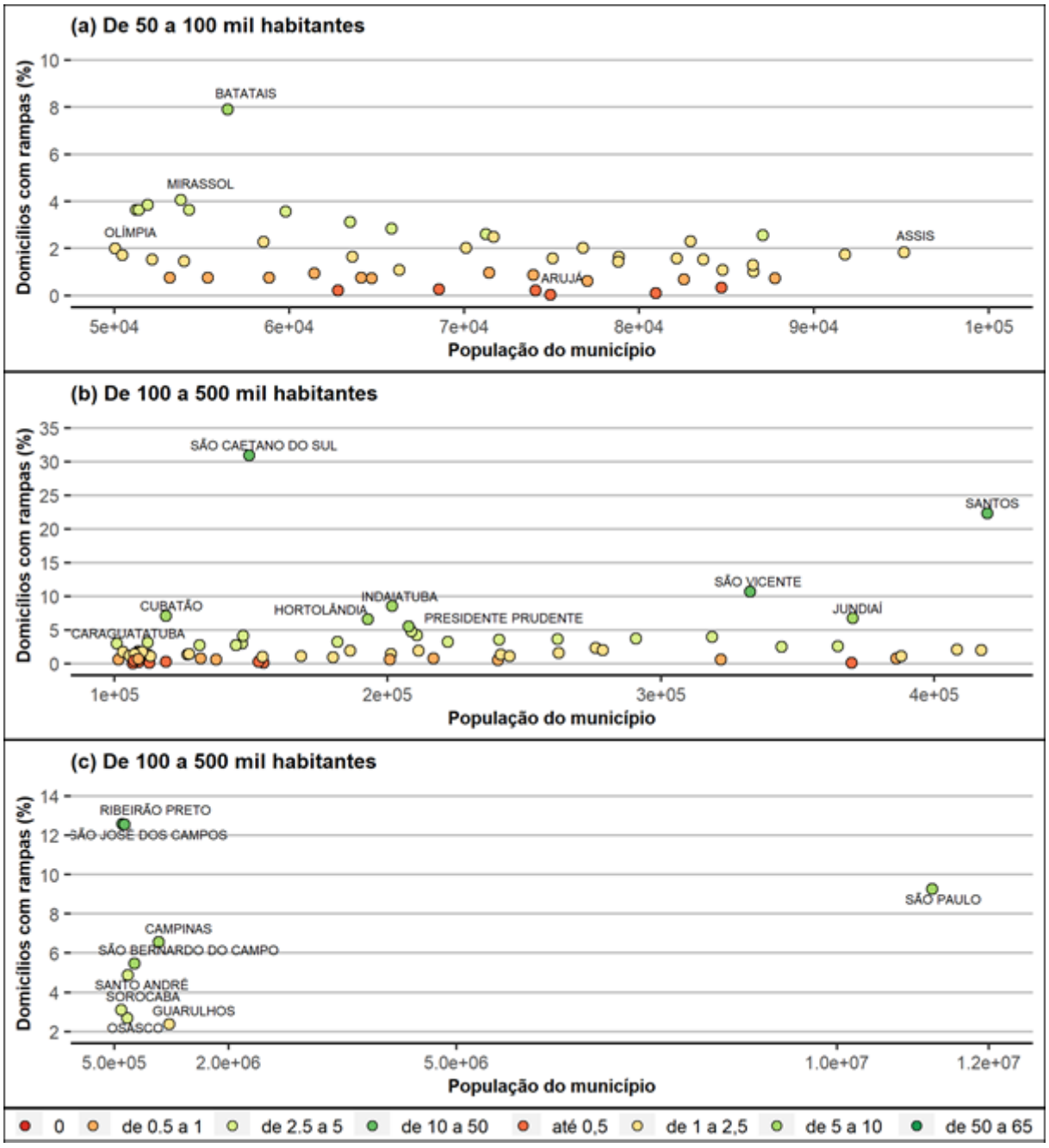
sendo Santos o mais populoso, com 419.400 habitantes, e São Vicente o décimo mais populoso, com 332.445. Em contrapartida, Caraguatatuba, que também é litorâneo, mas o menos populoso do estrato populacional, possui apenas 3,02% dos domicílios com face para logradouros com rampas para cadeirantes.

Enfim, no gráfico da Figura 5c, o município de São Paulo apresenta-se como ponto discrepante devido ao elevado número de habitantes, 11.253.503, contudo limitando-se a apenas 9,28% de domicílios com face para logradouros com rampas para cadeirantes. Dentre os outros municípios do estrato populacional destacam-se Ribeirão Preto e São José dos Campos, com porcentagens aproximadas de 12,5% e população total na faixa dos 600 mil habitantes.



Nota: A legenda se refere às faixas percentuais de domicílios com rampas para cadeirantes na face de acesso do logradouro.

Figura 4 – % de domicílios com rampas para cadeirantes em função da população do município (Parte 1)
Fonte: Adaptado de IBGE (2010a).



Nota: A legenda se refere às faixas percentuais de domicílios com rampas para cadeirantes na face de acesso do logradouro.

Figura 5 – % de domicílios com rampas para cadeirantes em função da população do município (Parte 2)
Fonte: Adaptado de IBGE (2010a).

2.2 Debate

De acordo com Maia e Quadros (2009), a organização dos municípios em grupos relativamente homogêneos é um grande desafio devido à variabilidade de estágios de desenvolvimento regional. Apesar disso, muitas legislações brasileiras adotam critérios populacionais, como é o caso do Estatuto da Cidade e a PNMU, ao definir os municípios que precisam desenvolver Plano Diretor e PMU, respectivamente. Contudo, cabe enfatizar estas legislações ainda consideram outras características de desenvolvimento regional como áreas de interesse turístico, áreas sob influência de impacto ambiental, cidades litorâneas, regiões metropolitanas e populosas (BRASIL, 2012; BRASIL, 2001).

Considerando, por exemplo, o recorte dos municípios que precisam desenvolver PMU, as Figuras 2, 3, 4 e 5 podem indicar o esforço que alguns municípios precisarão empenhar a fim de atingir os objetivos da PNMU de “melhoria das condições urbanas da população no que se refere à acessibilidade e à mobilidade” e “acessibilidade para pessoas com deficiência e restrição de mobilidade” (BRASIL, 2012).

Por exemplo, considerando apenas critérios populacionais, os municípios que necessitam desenvolver Plano Diretor e PMU encontram-se representados nas Figuras 4d e 5. A maior porcentagem de domicílios com acesso a rampas na face do logradouro é 30,96%, em São Caetano do Sul, o que está bem longe de atingir a acessibilidade para o universo dos usuários de cadeira de rodas. Outro indicativo alarmante é o desempenho de importantes municípios-sede de grandes regiões metropolitanas, como São Paulo (9,28%) e Campinas (6,59%), ambas com mais de 1 milhão de habitantes cada e, portanto, também precisam contemplar acessibilidade em seus respectivos PMU (BRASIL, 2012).

Maia e Quadros (2009) atentam ainda para a hipótese de condicionantes históricos, culturais e ambientais como influenciadores do grau de desenvolvimento socioeconômico de uma região. Estes condicionantes podem explicar o fato, por exemplo, de 9 municípios de até 5 mil habitantes e 3 municípios até 10 mil habitantes apresentarem porcentagens de domicílios com acesso à rampas acima de 20%. Destes, 7 foram fundados a menos de 30 anos e 5 a menos de 70 anos.

Leite (2011) aponta que “a questão da acessibilidade não é um tema recente”, porém, a preocupação com a mesma ainda está começando a fazer parte da cultura. Uma possível interpretação é que estes municípios mais recentemente fundados, com áreas urbanas mais recentemente desenvolvidas, possam ter incorporado a premissa de acessibilidade no planejamento urbano. Esta característica também pode ser resultado de aprendizado pela experiência vivenciada por cidades maiores (SILVEIRA, 2020). Outro ponto a ser considerado é que, anteriormente ao Estatuto da Cidade (BRASIL, 2001) e a PNMU (BRASIL, 2012), que tratam de premissas de acessibilidade, o planejamento urbano dos municípios era guiados primordialmente pela Lei de Parcelamento do Solo Urbano

(BRASIL, 1979), que não tange este tópico. Portanto, uma hipótese é que os municípios que incorporaram práticas de acessibilidade à seus projetos o fizeram por meio de legislação municipal.

Esta mesma análise pode se dar no sentido contrário, para explicar o porquê muitos municípios tem tanta dificuldade em tornar suas calçadas acessíveis. Como já mencionado, a Lei de Parcelamento do Solo Urbano (BRASIL, 1979) não trás nenhuma exigência sobre acessibilidade, nem o Manual de Aprovação de Projetos Habitacionais da Secretaria de Habitação do Estado de São Paulo (GRAPROHAB, 2019) faz menção à nenhuma exigência de projetos que prevejam rampas para cadeirantes. Na prática, tanto os municípios em estágio de inicial expansão urbana, tanto os que já se encontram em estágios mais avançados de desenvolvimento e adensamento acabam por não incorporar requisitos de acessibilidade à seus projetos – a não ser nos casos específicos mencionados que os municípios tenham desenvolvido legislação municipal específica. Logo, caberá às prefeituras, ao longo dos anos, providenciarem as adaptações necessárias.

2.3 Considerações Finais do Capítulo

De forma geral, pode-se concluir que a porcentagem de domicílios que tem seu acesso no passeio público servido de rampa de acessibilidade é relativamente baixa em todos os estratos populacionais estudados neste trabalho. Dentre os que apresentaram porcentagens altas, aparecendo como pontos discrepantes dentre o estrato populacional de até 5 mil habitantes, inclusive dentre todos os estratos populacionais, pode-se destacar Santa Cruz da Esperança e Motuca, com porcentagens de domicílios com rampas acima de 50%.

Os outros estratos populacionais também apresentaram algumas características específicas, cabendo salientar comparações relevantes como municípios de populações similares, porém com porcentagens muito discrepantes, por exemplo, Coronel Macedo e Braúna, no estrato de 5 a 10 mil habitantes, e Águas de Lindóia e Cerqueira César, no estrato de 10 a 20 mil habitantes. Além disso, nos estratos a partir de 50 mil habitantes não há mais municípios com domicílios sem rampas. Do estrato populacional mais populoso evidenciam-se positivamente em termos percentuais Ribeirão Preto e São José dos Campos e negativamente os municípios mais representativos em termos populacionais, São Paulo, Guarulhos e Campinas.

Apesar do estrato populacional ter sido utilizado como fator condicionante para agrupar municípios, observou-se que fatores regionais e históricos também podem ser explicativos nas análises. Foi o caso de municípios de estratos menos populosos que foram mais recentemente emancipados e que demonstraram melhor desempenho em porcentagem de rampas. Uma possível explicação pode ser encontrada no desenvolvimento

pautado em legislações urbanísticas mais recentes, federais e locais, que passaram a incorporar princípios de acessibilidade, somada à mudanças culturais, resultando em cidades que se expandem de forma mais acessível (ao invés de precisar se adaptar depois de consolidada).

Observando-se as características dos municípios litorâneos no estrato populacional entre 100 e 500 mil habitantes também é possível refletir mais a fundo sobre fatores regionais e culturais. Santos e São Vicente são municípios confrontantes e litorâneos, com porcentagens de domicílios com rampas na face do logradouro que se destacam no grupo. Em contrapartida, Caraguatatuba, que também é litorâneo, mas com população relativamente menor, apresenta desempenho percentual bem menor, assim como os outros municípios litorâneos do estado.

De acordo com Guimarães Rafaella Oliveira e Santos (2018), quando o foco passa a ser o pedestre, a calçada é um requisito básico de circulação na cidade. Similarmente, quanto o foco são usuários de cadeiras de rodas, rampas acessíveis nas calçadas são requisitos básicos de circulação na cidade, previstos legalmente na Constituição de Constituição Federal de 1988 (BRASIL, 1988), no Estatuto da Cidade (BRASIL, 2001) e na PNMU (BRASIL, 2012). Consequentemente, os dados analisados apontam que os municípios de todos os estratos populacionais estão muito distantes de proporcionarem uma mobilidade independente aos usuários de cadeiras de rodas em ambientes urbanos, isso porque foram analisadas apenas as informações quantitativas de rampas de acessibilidade em passeios públicos, sem considerar outros aspectos como condições das rampas e das calçadas e travessias de pedestres em si.

Essa realidade nos leva a refletir sobre a importância de novos estudos que analisem estes outros aspectos, bem como a acessibilidade em seu nível universal. Cabe reconhecer que cada município e região é resultado de um processo histórico, cultural e ambiental específico e se encontra num estágio de desenvolvimento diferente. Paralelamente, observamos na prática que dinâmicas de desenvolvimento urbano tendem a replicar processos já consolidados em regiões metropolitanas. Logo, considerando esta tendência, algumas cidades ainda estão em tempo de prevenir alguns erros já vivenciados por regiões mais maduras e metrópoles.

3 Identificação Automática de Objetos

Este capítulo abordará brevemente os conceitos utilizados na elaboração da técnica de detecção de rampas de acessibilidade, cujo entendimento foi essencial para o desenvolvimento prático da aplicação proposta.

Na sequência são introduzidos os conceitos de Aprendizado de Máquina, Redes Neurais Artificiais, Aprendizagem Profunda e Redes Neurais Convolucionais. São abordados tópicos de visão computacional e detecção de objetos, além de ser apresentado um levantamento de estudos considerados relevantes no desenvolvimento deste trabalho.

3.1 Aprendizado de Máquina

O termo Aprendizado de Máquina (do inglês, *Machine Learning*) foi utilizado pela primeira vez por Arthur Samuel, referindo-se à área da Ciência da Computação dedicada ao estudo da habilidade das máquinas aprenderem ou executarem uma tarefa sem terem sido necessariamente programadas para a mesma (SAMUEL, 1959). Alguns autores apresentam o Aprendizado de Máquina com um ramo da Inteligência Artificial (FRANCO, 2014).

Diferentemente da abordagem tradicional para solução de problemas com suporte computacional, quando o usuário determina o algoritmo a ser utilizado, na abordagem de Aprendizado de Máquina o algoritmo é construído com base na experiência obtida a partir de dados (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018). Estes dados podem se dar na forma de conjunto de dados rotulados, sendo que a qualidade e quantidade de dados e rótulos têm impacto direto na performance (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018).

Um dos exemplos mais tradicionais é o filtro anti-*spam*. Neste exemplo, o filtro identifica o que é ou não *spam*, conforme a classificação que o usuário fez anteriormente. Este filtro também pode se desenvolver além da classificação do usuário, como por exemplo, em casos em que *e-mails* com idiomas que diferem do utilizado naquela conta são considerados *spam* – mesmo que o usuário não o tenha indicado (FRANCO, 2014). Este exemplo nos leva aos conceitos de aprendizado supervisionado e não supervisionado.

Para compreender estes conceitos é necessário ter em mente que um algoritmo pode ser entendido como uma sucessão de instruções capazes de transformar uma entrada em uma saída (FRANCO, 2014). Em aprendizado de máquina, estes algorítimos foram descritos por Mitchell (1997): *"[...] um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e performance P, se sua performance P nas tarefas em T, na forma medida por P, melhoram com a experiência E"*.

Neste sentido, o aprendizado supervisionado ocorre quando, a partir do fornecimento de um conjunto de dados de entrada e suas respectivas saídas, desenvolve-se a experiência para prever ou produzir a saída para entradas até então inexistentes no conjunto inicial de treinamento. O objetivo central é encontrar os parâmetros de relação entre os dados de entrada e saída (BRAGA; CARVALHO; LUDERMIR, 2000; FRANCO, 2014).

Já no caso do aprendizado não supervisionado, são conhecidas apenas as entradas. A máquina identificará os possíveis padrões (semelhanças e diferenças) para a partir daí, elaborar saídas corretas (FRANCO, 2014). A obtenção da experiência ocorre então devido à características de correlação e redundância dos dados de entrada (BRAGA; CARVALHO; LUDERMIR, 2000). Existem ainda outros tipos de aprendizado como o semi-supervisionado, o aprendizado ativo e o aprendizado por reforço (NILSSON, 1998).

De acordo com Mohri, Rostamizadeh e Talwalkar (2018), o aprendizado supervisionado é geralmente aplicado em tarefas de classificação e regressão, enquanto o aprendizado não supervisionado abrange tarefas de agrupamento e redução dimensional. Estas são consideradas tarefas recorrentes em aprendizado de máquina e os referidos autores consideram cada uma como de:

- Classificação: atribui uma categoria conhecida para determinado item;
- Regressão: prevê um valor numérico para determinado item;
- Agrupamento: fraciona um conjunto de itens em subconjuntos homogêneos;
- Redução dimensional: transforma a representação inicial dos itens em uma representação com menos dimensões, preservando as principais propriedades da representação inicial. O exemplo mais comum é o processamento de imagens

Redes Neurais Artificiais

Redes Neurais Artificiais (RNAs) surgiram a partir dos mecanismos propostos por Pitts e McCulloch (1947) na década de 40, mas foi no final da década de 80 que passaram a ser mais estudadas. A ideia principal deste modelo computacional é resolver problemas simulando processos tais como ocorrem no cérebro humano (ZHAO et al., 2019). Nesse sentido, as RNAs desempenham um papel proeminente no aprendizado de máquina (NILSSON, 1998).

As RNAs são modeladas de forma similar ao cérebro humano, sendo compostas por neurônios. Enquanto o cérebro humano possui milhões de neurônios que se comunicam continuadamente em processos não lineares (FRANCO, 2014), as RNAs podem ser formadas por milhares de unidades de processamento simples (neurônios), que atuam

como sistemas de processamento paralelo e distribuído (BRAGA; CARVALHO; LUDERMIR, 2000).

A Figura 6 representa o modelo proposto por McCulloch e Pitts, no qual o neurônio é representado por entradas (x), seus respectivos pesos (w), corpo e suas respectivas saídas (y). No corpo ocorre a soma do produto das entradas pelos pesos, que posteriormente passam por uma função de ativação (*threshold*), produzindo saídas.

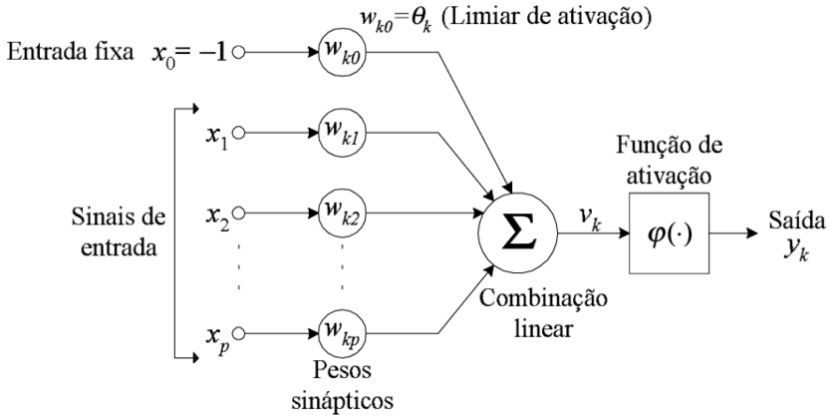


Figura 6 – Modelo do neurônio artificial de Mcculloch e Pitts
Fonte: Fernandes (1999)

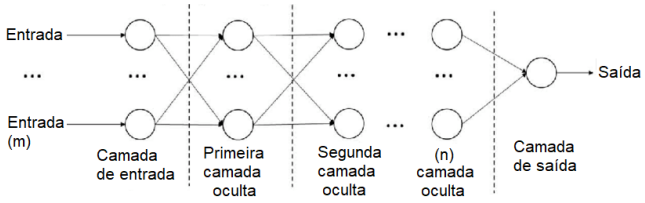
Diferentemente do neurônio biológico, existe a possibilidade de uma entrada fixa (x_0), também conhecida como *bias*, permitindo maior *"liberdade e adaptação da rede ao conhecimento a ela fornecido"* (FRANCO, 2014). Esta entrada extra pode ser, por exemplo, uma entrada de treinamento. Mas este modelo ainda limita-se ao fato dos pesos serem fixos, não permitindo ajustes.

No fim da década de 50 surge o modelo de RNA *Perceptron*, um classificador desenvolvido inicialmente para reconhecer padrões de fala e escrita (BRAGA; CARVALHO; LUDERMIR, 2000). É neste modelo que insere-se a possibilidade de aprendizado às RNAs. Entretanto, obstáculos relacionados à problemas de sobreajuste, falta de dados de treinamento, baixo desempenho geral, principalmente, escassa capacidade computacional, levaram a certo desinteresse pela área (ZHAO et al., 2019).

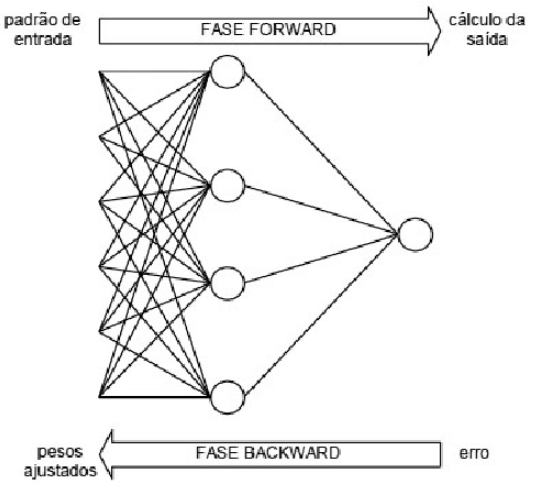
As RNAs passaram a emergir novamente a medida que estas limitações foram sendo superadas e a partir do surgimento das RNAs multicamadas (vide Figura 7a), abrindo caminho para os primeiros conceitos de Aprendizagem Profunda (em inglês, *Deep Learning*) (ZHAO et al., 2019). A principal inovação do modelo multicamadas refere-se ao uso de camadas intermediárias ou escondidas, viabilizando ajustes de peso, consequentemente, atribuindo maior agilidade em ambientes adaptativos (FRANCO, 2014).

Visando otimizar ainda mais estas redes, foram desenvolvidos algoritmos de *back-propagation* (vide Figura 7b), criados a partir da generalização da Regra Delta. A regra de aprendizado Delta foi introduzida às redes *Perceptron* visando o tratamento de dados não

separáveis linearmente (BRAGA; CARVALHO; LUDERMIR, 2000). A grande vantagem do modelo de *backpropagation* é a sua adequação à sistemas linearmente ou não separáveis, além de propiciar conexões das entradas com as saídas, utilizando erros para refinar as decisões (NILSSON, 1998).



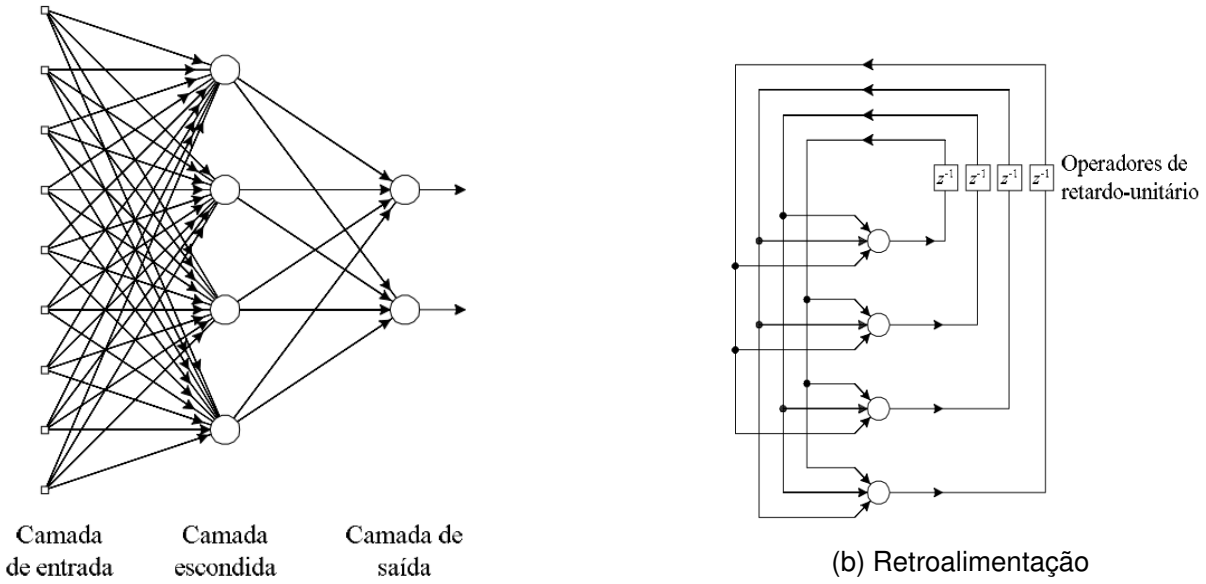
(a) Rede multicamadas



(b) Rede com backpropagation

Figura 7 – Tipos de redes neurais
Fonte: (a) De Jesus et al. (2019) e (b) Corrêa et al. (2016)

Quanto à arquitetura dos neurônios, existem basicamente duas formas de propagação: redes de propagação direta (do inglês *feedfoward*, vide Figura 8a e redes com retroalimentação ou realimentação (do inglês *recurrent*, vide Figura 8b. Nas arquiteturas de propagação direta o fluxo de informação é unilateral, enquanto na arquitetura de realimentação, não há restrição quanto à interconexões de camadas e neurônios (FERNANDES, 1999).



(a) Propagação direta

(b) Retroalimentação

Figura 8 – Arquiteturas de RNAs
Fonte: Fernandes (1999)

3.2 Aprendizagem Profunda

A aprendizagem profunda, juntamente com o aprendizado de máquina, são áreas da inteligência artificial. De acordo com LeCun, Bengio e Hinton (2015), as técnicas de Aprendizado de Máquina possuem limitações no processamento de dados brutos. Até então, estes dados brutos precisavam ser transformados em dados compatíveis com os algoritmos, de forma a atender aos requerimentos dos cálculos de aprendizado.

A aprendizagem profunda, por sua vez, permite que os dados sejam inseridos na sua forma natural, sendo possível a detecção automática dos dados relevantes para realizar as tarefas desejadas. Além disso, pode-se manipular conjuntos de dados gigantes, incluindo *Big Data* (LECUN; BENGIO; HINTON, 2015).

De forma geral, os modelos de aprendizagem profunda tratam de redes neurais com muitas camadas, por isso mais profundas e mais complexas (BENGIO, 2009). Como já mencionado anteriormente, enquanto o aprendizado de máquina fica delimitado por algoritmos de aprendizado supervisionado, não supervisionado, semi-supervisionado e por reforço, a aprendizagem profunda atua nos moldes do aprendizado não supervisionado. Logo, mesmo que apenas os dados de entrada estejam disponíveis, é possível decodificar padrões e desenvolver a habilidade de criar grupos e classes automaticamente (RUSSELL; NORVIG, 2009).

É justamente o grande volume de dados de entrada que agrega redundância ao modelo, viabilizando a aprendizagem de funções complexas (RUSSELL; NORVIG, 2009). Adicionalmente, a incorporação do algoritmo de *backpropagation* auxilia a máquina no

ajuste dos parâmetros internos utilizados para calcular a representação em cada camada, incluindo camadas anteriores (LECUN; BENGIO; HINTON, 2015).

Os principais progressos da aprendizagem profunda são resultantes da habilidade dos modelos computacionais aprenderem representações de dados mais sofisticados, com múltiplos níveis de abstração. A ampliação da capacidade computacional, como o desenvolvimento de *Graphics Processing Unit* - Unidade de Processamento Gráfico (GPUs) potentes, e a disponibilidade de bases de dados em grande escala, também tiveram papel decisivo no avanço dos modelos de aprendizagem profunda (LIU et al., 2020).

As técnicas de aprendizagem profunda vem se mostrando promissoras, principalmente no que diz respeito ao processamento de imagens, vídeos e áudios. Neste campo de atuação, as principais expectativas se relacionam a tarefas de entendimento da linguagem natural, como identificação de tópicos, análise de sentimentos, resposta à perguntas e tradução de idiomas (LIU et al., 2020; LECUN; BENGIO; HINTON, 2015; BENGIO, 2009).

Redes Neurais Convolucionais

A busca por um maior nível de otimização das RNAs aliada à evolução das técnicas de aprendizagem profunda levaram ao desenvolvimento das Redes Neurais Convolucionais – *Convolutional Neural Network* (CNN)s, consideradas o modelo mais representativo de aprendizagem profunda (ZHAO et al., 2019). De forma geral, as CNNs apropriam-se de camadas qualificadas para identificar os padrões mais relevantes para a rede, sendo que os neurônios se tornam especialistas na base de dados utilizada (DATA SCIENCE ACADEMY, 2019).

De acordo com LeCun, Bengio e Hinton (2015), CNNs são compatíveis no processamento de dados do tipo matrizes múltiplas, como por exemplo, imagens coloridas composta por três matrizes 2D (*arrays*), as quais contém intensidades de pixel nestes três canais de cores (camadas RGB). Dentre as modalidades de dados destacam-se matrizes 1D de sinais e sequências, como séries de dados temporais e de linguagem; matrizes 2D, como imagens e áudios (espectogramas); e matrizes 3D, como imagens volumétricas e vídeos (LECUN; BENGIO; HINTON, 2015; GOODFELLOW; BENGIO; COURVILLE, 2016).

Diferentemente das RNAs que utilizam multiplicação geral da matriz, as CNNs são redes neurais que utilizam, em pelo menos uma camada, um tipo de operação linear especializada, chamada convolução. Como ilustrado na Figura 9, as RNCs se caracterizam por redes dotadas de camadas de convolução intercaladas com funções de ativação não lineares – *Rectified Linear Unit* - Unidade Linear Retificada (ReLU) – aplicadas aos resultados e com camadas de subamostragem (*pooling*), além de uma última camada totalmente conectada (*fully connected*), a qual conecta todos os neurônios provenientes da

camada de *pooling* aos neurônios de saída (GOODFELLOW; BENGIO; COURVILLE, 2016; FARIA, 2018).

Para facilitar a visualização, são apresentadas na Figura 9 apenas uma camada de cada etapa, contudo, os modelos atuais são compostos por inúmeras camadas alternadas.

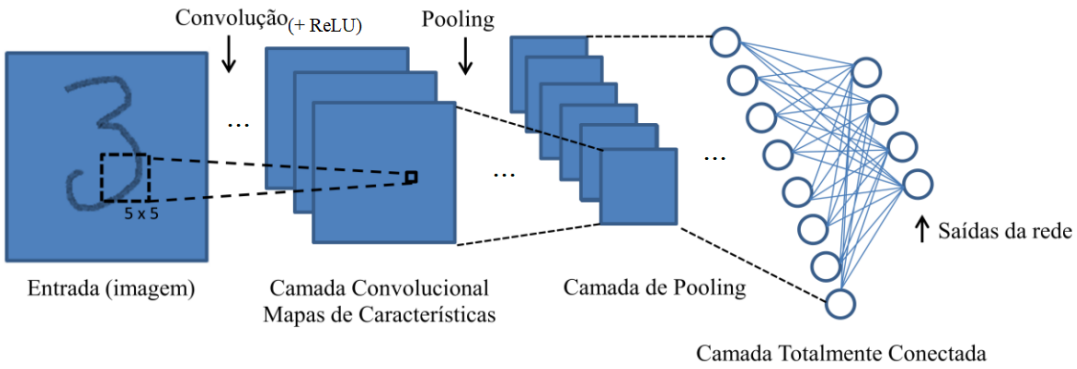
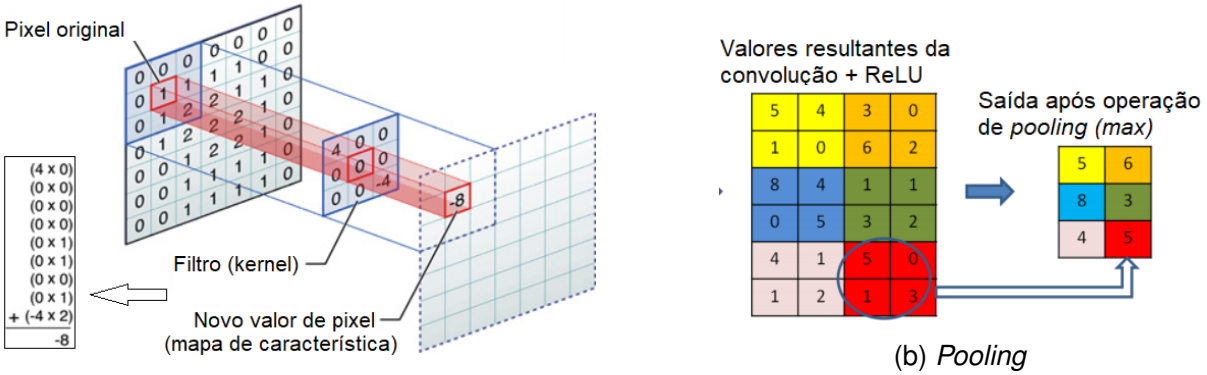


Figura 9 – Arquitetura simplificada de uma Rede Neural Convolucional
Fonte: Adaptado de Faria (2018)

Neste modelo, as camadas convolucionais submetem as unidades de uma imagem à filtros convolucionais (chamados *kernels*), resultando em novas imagens (mapa de característica) que, por padrão, tem suas dimensões relativamente reduzidas, onde as feições detectadas são acentuadas (vide detalhamento na Figura 10a). Posteriormente, a ativação linear resultante da etapa de convolução é submetida à funções não lineares, dentre as mais utilizadas a ReLU. Ao se introduzir não linearidade ao modelo, habilita-se um melhor gradiente de propagação. (GOODFELLOW; BENGIO; COURVILLE, 2016).



(a) Convolução

(b) *Pooling*

Figura 10 – Exemplos de tipos de camadas
Fonte: Adaptado de (a) Apple (2016) e (b) Faria (2018)

Numa próxima etapa, os resultados das camadas convolucionais são submetidos à camadas de *pooling*, as quais permitem identificar a informação mais importante e significativa (vide detalhamento na Figura 10b) (FARIA, 2018). Por fim, a última etapa sumariza estaticamente a característica mais relevante de acordo com as saídas vizinhas, a

partir da camada anterior. Nesta camada são determinadas as classes, de acordo com um conjunto de treinamento. (GOODFELLOW; BENGIO; COURVILLE, 2016; FARIA, 2018).

Na Figura 11 é possível observar um exemplo dos produtos das etapas intermediárias da RNC.

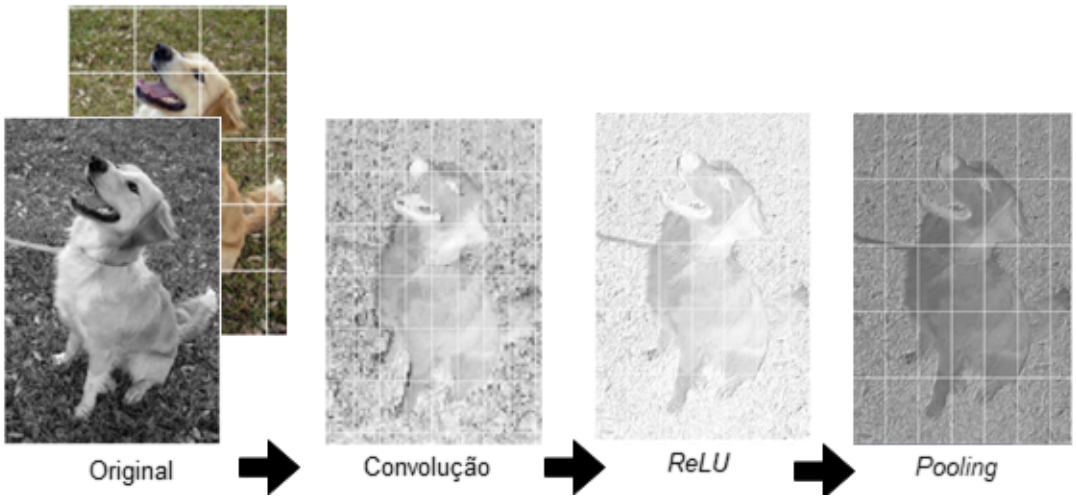


Figura 11 – Exemplo das saídas de cada etapa numa Rede Convolucional.
Fonte: Adaptado de Shen (2018)

3.3 Visão Computacional

Visão computacional – *Computer Vision* (CV) – é uma das tarefas mais estudadas no campo da aprendizagem profunda (BALLARD; HINTON; SEJNOWSKI, 1983). Como o próprio nome sugere, o objetivo da CV é imitar a funcionalidade do olho humano e dos componentes do cérebro responsáveis pelo seu senso de visão (GOODFELLOW; BENGIO; COURVILLE, 2016).

As primeiras iniciativas de CV surgiram a partir de trabalhos como o de Hubel e Wiesel (1959), que descreveram as propriedades de resposta dos neurônios do córtex visual de gatos, o qual é organizado de forma hierárquica. Apesar de não estarem diretamente relacionadas à CV, as descobertas destes estudos serviram de base para o desenvolvimento de RNCs (FARIA, 2018).

Até 2012, os trabalhos de visão computacional se baseavam, predominantemente, em Máquina de Vetores de Suporte, Florestas Aleatórias e *Bootsting*. A partir do trabalho de Krizhevsky, Sutskever and Hinton, que venceram a competição *ImageNet Challenge* ao classificar mais de 1000 diferentes classes de imagens com alta performance e precisão, as RNCs ganharam destaque neste campo (JUNIOR; COLOVAN, 2018).

Atualmente, a CV é um campo muito amplo, abrangendo variadas formas de processamento de imagens e vídeos, visando a identificação de feições, alocação em classes

e até mesmo o desenvolvimento de novas habilidades visuais (GOODFELLOW; BENGIO; COURVILLE, 2016). Dentre as tarefas mais comuns envolvidas em CV estão a detecção, reconhecimento, segmentação, classificação e localização. De forma geral, não há consenso na literatura quanto à definição desses termos ao se relacionarem à tarefas de CV (ANDREOPOULOS; TSOTSOS, 2013), assim, neste trabalho será utilizada a diferenciação mais aceita, definida por Li, Johnson e Yeung (2017), relacionada a seguir:

- **Classificação:** atribuir um rótulo a determinado item a partir de um conjunto fixo de categorias;
- **Classificação + localização:** além da tarefa de classificação, requer também localizar um objeto desejado através de uma caixa delimitadora, obtendo suas respectivas coordenadas;
- **Detecção:** semelhante à tarefa atribuída à classificação + localização, contudo, refere-se à rotulação e delimitação de diversos itens;
- **Segmentação semântica:** realiza a classificação dos pixels para identificar as classes de objetos;
- **Segmentação de instância:** semelhante à segmentação semântica, mas marca os pixels de várias instâncias do mesmo objeto separadamente.

A Figura 12 apresenta um exemplo das tarefas de VC para imagens.

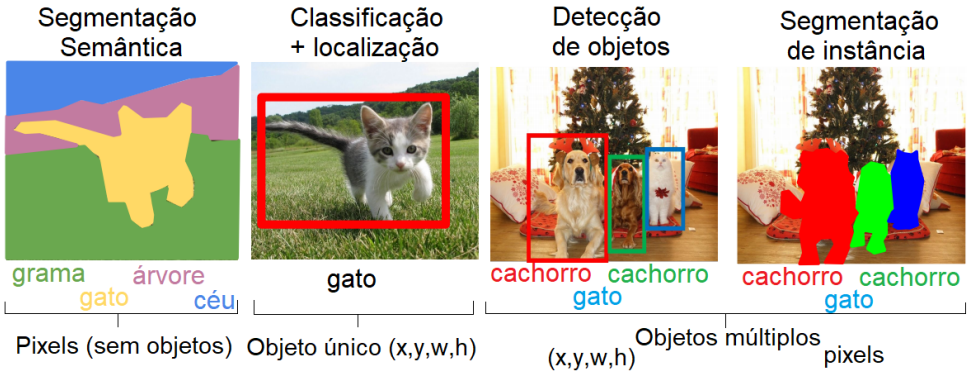


Figura 12 – Exemplo das tarefas de CV.
Fonte: Adaptado de Li, Johnson e Yeung (2017)

Como ilustrado na Figura 12 a localização espacial dos objetos e/ou classes podem ser identificadas por caixas delimitadoras (denomominadas mais frequentemente de *bounding box*), por máscaras de segmentação dos pixels ou ainda por uma máscara com limite próximo aos dos objetos investigados (LIU et al., 2020).

3.3.1 Detecção de Objetos

O reconhecimento de objetos é considerado o eixo central da CV, sendo todas as demais tarefas, incluindo a de detecção de objetos, ramificações deste eixo central (ANDREOPOULOS; TSOTSOS, 2013). De acordo com Szeliski (2011), quando se trata da tarefa de reconhecimento de objetos na qual o que se busca é conhecido, tem-se então o que se denomina detecção de objetos.

No âmbito do processamento de imagens, a CV pode ser entendida como a subárea que estuda os métodos e técnicas que possibilitam um sistema computacional interpretar imagens (GOODFELLOW; BENGIO; COURVILLE, 2016). Então, a detecção de objetos em imagens envolve escanear uma imagem para verificar onde determinada correspondência ocorre e identificá-la (SZELISKI, 2011).

Em resumo, a detecção de objetos pode ser entendida como o procedimento para identificar um objeto, determinar a que classe o objeto pertence e estimar a sua localização – exibindo a caixa delimitadora ao redor do mesmo (LIU et al., 2020; PATHAK; PANDEY; RAUTARAY, 2018; RUSSAKOVSKY et al., 2015). Neste sentido, pode-se entender a tarefa de detecção de objetos como um problema de classificação – categorizar um objeto – e de regressão – definir a localização da caixa delimitadora.

Da mesma forma que olhamos para um objeto e, inconscientemente, identificamos o que é e onde está, a CV desenvolve a detecção de objetos. Entretanto, no ambiente computacional, esta atividade de reconhecimento visual pode envolver diversos desafios (vide Figura 13). Os principais se relacionam à variações de angulações, escala e condições de iluminação, partes ocultas, pouco contraste em relação ao fundo da imagem, entre outros.

Figura 13 – Desafios da detecção de objetos.
Fonte: Elaborado pela autora

Outro desafio quanto à detecção de objetos se refere à localização de uma quantidade desconhecida de objetos na imagem, ou seja, não se sabe quantas instâncias de uma determinada categoria de objeto serão identificadas em uma cena. Por esta razão, a abordagem tradicional de CNNs torna-se inviável pois, enquanto o tamanho da saída destas são fixas, o tamanho da saída na detecção de objetos é variável. Para superar este obstáculo, foram desenvolvidos duas estruturas básicas de CNNs para detecção de objetos: *Region Proposal Based Framework* e *Regression/Classification Based Framework Region* (WANGENHEIM, 2018).

Region Proposal Based Framework

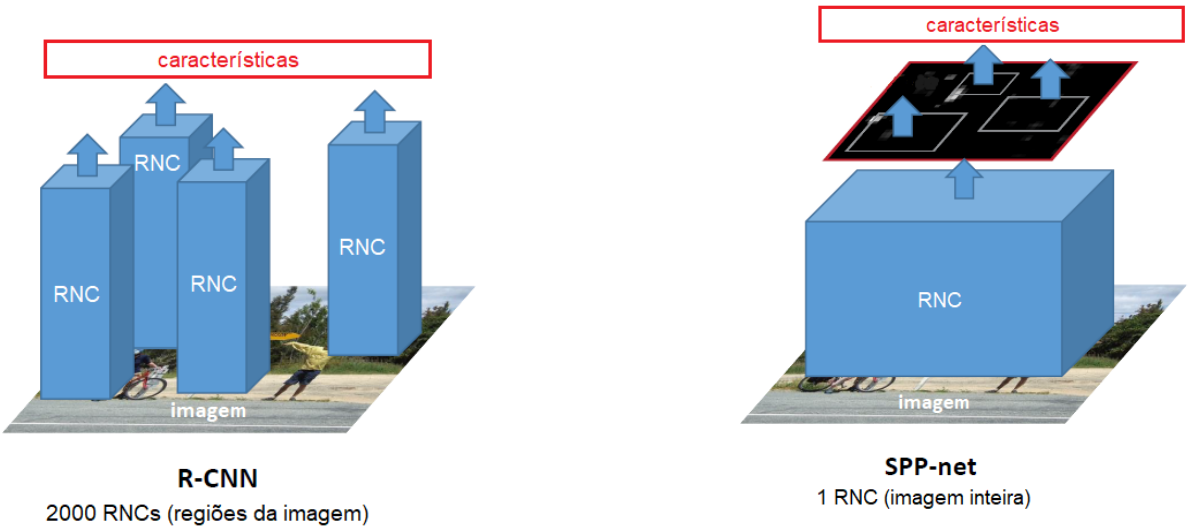
Region Proposal Based Framework são estruturas para detecção de objetos que podem ser divididas em duas etapas, similares ao mecanismo de detecção do cérebro humano – primeiro é desenvolvida uma varredura geral no cenário e, a partir daí, focaliza-se em regiões de interesse (ZHAO et al., 2019).

O *Region-based Convolutional Neural Networks* (R-CNN), proposto por Girshick et al. (2013), é um dos modelos mais representativos deste tipo, que atua de forma associada à extratores de características baseados em CNNs (ZHAO et al., 2019). Inicialmente, propõe-se uma busca seletiva, gerando inúmeras sub-regiões candidatas à classificação. Estas regiões são combinadas em regiões maiores de acordo com suas semelhanças através de uma CNN, e posteriormente, cada uma dessas regiões é submetida à um classificador linear (GIRSHICK et al., 2013).

Uma característica do R-CNN é que a camada totalmente conectada requer um tamanho fixo para a imagem de entrada, ocasionando lentidão e perda de acurácia no processo de teste pois cada região avaliada precisa ser re-computada – distorcida ou cortada – para um mesmo tamanho. Em média são geradas cerca de 2.000 sub-imagens neste procedimento (ZHAO et al., 2019).

Visando aprimorar este o processo, He et al. (2014) propôs o modelo *Spatial Pyramid Pooling Networks* (SPP-net), o qual insere a imagem como um todo na CNN antes da criação de sub-regiões (vide Figura 14). Esta melhoria é possível graças à adição da camada *Spatial Pyramid Pooling* (SPP), que remove a restrição de tamanho fixo – daí o nome SPP-net.

Estes dois modelos são considerados multietapas, sendo a primeira etapa de extração das características e a segunda de classificação das regiões e definição das localizações. A partir destes, Girshick (2015) apresentou o modelo *Fast* R-CNN, propondo o desenvolvimento destas etapas em um único algoritmo. Neste modelo, ao invés de utilizar uma camada SPP que possui vários níveis de *bins* espaciais, utiliza-se a camada de *Pooling* de Região de Interesse, a qual contém apenas um nível. Para cada saída desta camada,



(a) *R-CNN* (b) *SPP-net*

Figura 14 – Diferenças estruturais das CNNs.
Fonte: Adaptado de He et al. (2014)

é realizada uma classificação com outra camada que prediz a categoria do objeto e sua respectiva caixa delimitadora. Esta alteração facilita o treinamento e permite um ajuste mais refinado das camadas convolucionais (GIRSHICK, 2015).

De acordo com Wangenheim (2018), o modelo R-CNN exigia um tempo de treinamento da ordem de 84 horas e tempo de teste de 49 segundos. Este tempo foi reduzido para 25,5 horas e 4,3 segundos no modelo SPP-net e 8,75 horas e 2,3 segundos no modelo *Fast* R-CNN.

Considerando os modelos desenvolvidos a partir da estrutura de Classificadores de Região, os próximos aprimoramentos giravam em torno da busca pela detecção de objetos em tempo real. Então, Ren et al. (2015) adicionaram ao modelo *Fast* R-CNN uma rede totalmente convolucional *Region Proposal Network* (RPN), que diferentemente dos modelos até então conhecidos, permite métodos de aprendizagem orientada por dados. Neste contexto, surge o *Faster* R-CNN, que aprimorou o tempo de teste para 0,2 segundos (REN et al., 2015).

Posteriormente, os modelos mais recentes dentre as estruturas *Region Proposal Based Framework* são: *Region-based Fully Convolutional Networks* (R-FCN), *Feature Pyramid Networks* (FPN) e *Mask* R-CNN. As principais melhorias em cada um diz respeito à, respectivamente, redução do trabalho de processamento em cada camada *Pooling* de Região de Interesse, produção de mapas de características em várias escalas e possibilidade de aplicação para Segmentação de Instância (ZHAO et al., 2019).

Regression/Classification Based Framework

Diferentemente da proposta de multi-etapas do *Region Proposal Based Framework* que aborda a detecção de objetos de forma segmentada, a estrutura do *Regression/Classification Based Framework* trata a detecção de objetos como um problema de regressão único: desde os pixels da imagem até as coordenadas das caixas delimitadoras e suas respectivas classes prováveis (ZHAO et al., 2019).

Os modelos baseados em *Region Proposal Based Framework* utilizam sub-regiões da imagem para identificar objetos. Mesmo nas propostas em que a imagem é inserida na CNN como um todo, em alguma etapa, são geradas sub-regiões. Já nos modelos baseados em *Regression/Classification Based Framework*, a rede sempre considera a imagem como um todo. A Figura 15 apresenta um esquema básico destas duas estruturas, onde é possível identificar as diferenças mencionadas.

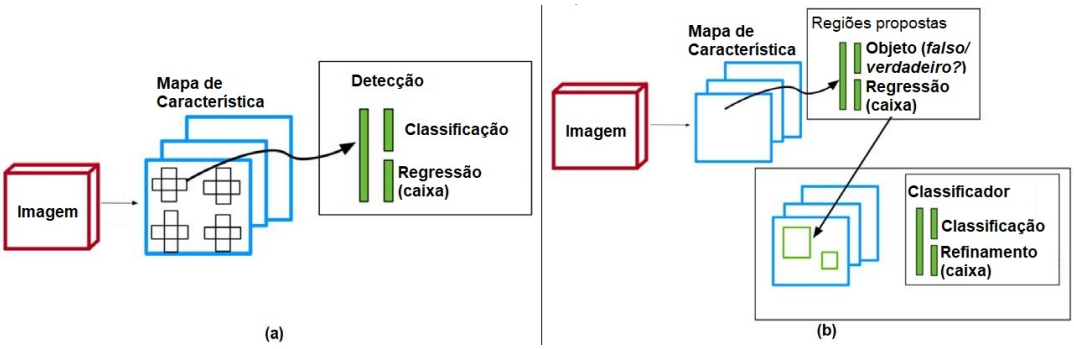


Figura 15 – Esquema de detecção em (a) *Regression/Classification Based Framework* e (b) *Region Proposal Based Framework*. Fonte: Adaptado de Ndonhong, Bao e Germain (2019).

Uma das primeiras iniciativas baseada em *Regression/Classification Based Framework* foi o *DetectorNet*, desenvolvido por Szegedy, Toshev e Erhan (2013), que formularam um detector de objetos a partir de aprendizagem profunda. Contudo, o modelo mostrou várias limitações relacionados à objetos sobrepostos. Em seguida, Sermanet et al. (2013), elaboraram o *OverFeat*, que também utiliza de aprendizagem profunda para melhorar a localização e delimitação dos objetos, além de uma janela multi-escala e deslizante. Dentre os principais modelos de *Regression/Classification Based Framework*, o *OverFeat* é o único em que o tamanho das imagens de entrada não é fixo (LIU et al., 2020).

Uma das etapas críticas de um classificador é eleger uma lista de objetos candidatos à classificação. Pinheiro, Collobert e Dollár (2015) refinaram esta etapa quando propuseram um sistema que, num primeiro momento, gerava uma máscara de segmentação de classes agnósticas e, num segundo momento, estimava a probabilidade de um determinado segmento estar centrado em um objeto completo. De forma similar, Erhan et al. (2014) criaram uma regressão baseada em *MultiBox*, que atribui uma caixa delimitadora contendo uma classe agnóstica e associa uma pontuação única de acordo com a probabilidade da mesma conter o objeto de interesse.

Yoo et al. (2015) propuseram um modelo de classificação capaz de convergir para uma caixa delimitadora resultante de um conjunto de previsões iterativas. A principal limitação deste modelo foi a atuação para múltiplas categorias. Numa abordagem diferenciada, Najibi, Rastegari e Davis (2015) formularam um modelo que dispensa o uso de algoritmos para criação de regiões candidatas: o *Grid Convolutional Neural Network* (G-CNN). O processo de detecção tem início a partir de um *grid* com escalas variadas de caixas delimitadoras. Em seguida, pelo processo de regressão, os elementos deste *grid* são movimentados e dimensionados iterativamente, até que se obtenha caixas que se aproximem dos objetos. Contudo, o modelo apresentava dificuldade em identificar objetos sobrepostos.

A partir deste princípio de divisão das imagens por um *grid*, foram desenvolvidos os modelos mais significativos dentre as estruturas de *Regression/ Classification Based Framework*, conhecidas também como detectores de etapa única: *Single-Shot Multibox Detector* (SSD), *RetinaNet*, *CornerNet*, *You Only Look Once* (YOLO) e suas versões atualizadas. Estes modelos também possuem a característica marcante de detecção em tempo real.

Em 2015, REDMON et al. propuseram o detector de objetos YOLO, que prevê simultaneamente várias caixas delimitadoras e probabilidades de classe para essas caixas, numa única rede convolucional. Para isso, o modelo divide a imagem num *grid* fixo, gerando *n* caixas delimitadoras utilizando este *grid* como referência. As caixas delimitadoras com probabilidades acima de um limiar são selecionadas para localizar e classificar um objeto dentro da imagem (vide Figura 16). Na Figura 16 verifica-se que na metodologia YOLO as imagens de entrada são redimensionadas e uma única rede convolucional passa a ser executada, limitando as detecções resultantes pela supressão não-máxima (que consiste na eliminação das detecções cujos valores não ultrapassam o limiar de confiança do modelo) (REDMON et al., 2015).Uma vez que YOLO analisa a imagem como um todo ao fazer detecções, implicitamente, este modelo computa informações contextuais sobre classes dos objetos (LIU et al., 2020).

Por outro lado, YOLO apresenta algumas limitações quanto à detectar objetos pequenos e agrupados, pois o modelo permite a detecção de apenas dois objetos numa determinada localização. Além disso, como apenas o último mapa de característica é utilizado para detecção, o modelo é limitado quanto à escalas e proporções variadas (WU; SAHOO; HOI, 2020). Liu et al. (2016) propuseram então o SSD, buscando superar estas limitações e encontrar um equilíbrio entre velocidade e precisão.

Este modelo foi inspirado essencialmente no já mencionado *MultiBox* e, diferentemente do YOLO que adota um *grid* fixo, o SSD discretiza o espaço de saída das caixas delimitadoras num conjunto de caixas de ancoragem com múltiplas escalas e proporções (WANGENHEIM, 2018). Cada caixa de ancoragem é refinada com valores aprendidos pelos regressores, na qual são associadas à probabilidades categóricas pelos classificadores.

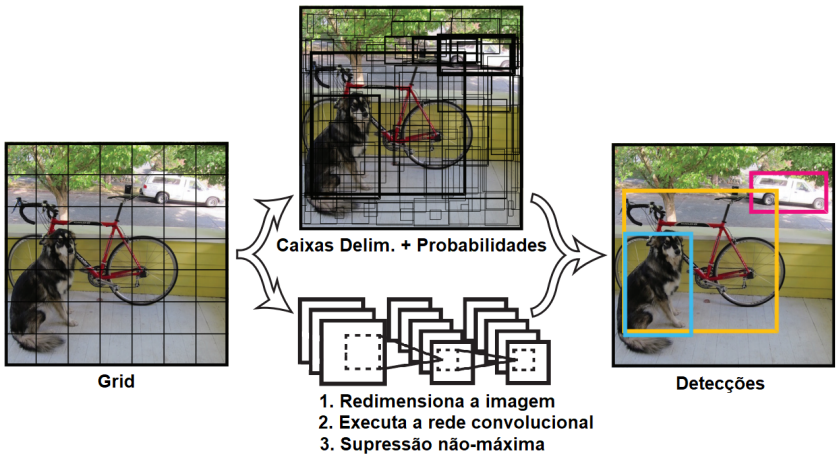


Figura 16 – Modelo e sistema de detecção YOLO.
Fonte: Adaptado de Redmon et al. (2015).

Em termos de acurácia, o SSD pode ser comparado ao *Faster* R-CNN, todavia, com a possibilidade de detecção em tempo real (ZHAO et al., 2019).

Apesar das melhorias implementadas, os modelos baseados em *Regression/Classification Based Framework* ainda deixavam a desejar no que diz respeito à diferenciação entre primeiro plano e plano de fundo. Isso ocorre devido ao desequilíbrio entre amostras positivas e negativas durante o treinamento (LIN et al., 2017). *RetinaNet*, elaborado por Lin et al. (2017), utiliza uma camada específica, conhecida por camada de perda focal, para balancear o peso das amostras negativas, evitando que seu grande número sobrecarregue o detector durante o treinamento (WU; SAHOO; HOI, 2020).

Nos modelos até então descritos, para definição de uma caixa delimitadora são estabelecidas 4 dimensões básicas: coordenadas x e y do centro, largura e altura. Em 2019, Law e Deng (2019) propôs um modelo que detecta objetos por apenas um par de coordenadas (canto superior esquerdo e canto inferior direito). Esta proposta embasou alguns outros trabalhos, como o de Duan et al. (2019) and Zhou, Wang e Krähenbühl (2019).

Redmon e Farhadi (2017) aprimoraram a primeira versão do detector YOLO, para o YOLOv2, que incorporou entre suas principais melhorias as estratégias de caixas de ancoragem (*cluster* de dimensão), normalização em lote e treinamento em várias escalas (ZHAO et al., 2019). Outra variação relevante no YOLOv2 foi a troca das camadas totalmente conectadas, presentes no YOLO, por uma CNN totalmente conectada, em que todos os neurônios da rede estão conectados a todos os neurônios em camadas adjacentes (DATA SCIENCE ACADEMY, 2019).

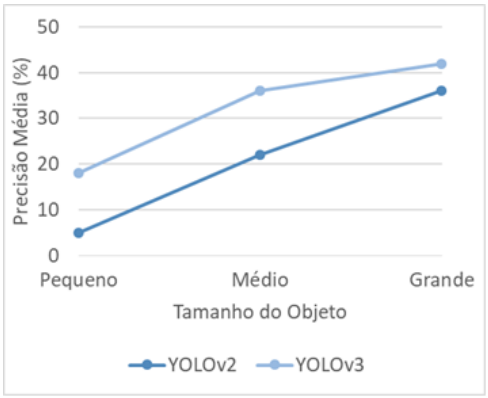
Posteriormente, Redmon e Farhadi (2018) desenvolveram ainda o YOLOv3, que é capaz de realizar rotulações múltiplas na classificação, como por exemplo, identificar um cachorro e a raça deste cachorro. Outra melhoria significante refere-se à detecção de objetos em três escalas diferentes: objetos em escala pequena, média e grande (vide

Figura 17b). Em relação às caixas de ancoragem, o modelo foi de 5 para 9 por célula do *grid*, sendo 3 para cada escala. Consequentemente, enquanto YOLOv2 pode prever 845 caixas delimitadoras, com essa nova configuração, o YOLOv3 pode prever 10.647.

Na Figura 17a é possível verificar as diferenças de precisão e velocidade entre as três propostas da família YOLO. Na Figura 17b fica evidente o aprimoramento em precisão, considerando a escala dos objetos.

Detector	Entrada	Precisão média (%)	Quadros por segundo
YOLO	448x448	62,5	42,34
YOLOv2	416x416	73,82	64,65
YOLOv2	544x544	75,95	39,14
YOLOv3	416x416	88,09	51,26

(a)



(b)

Figura 17 – Principais dados de performance dos detectores da família YOLO (a) e comparação considerando o tamanho dos objetos (b).
Fonte: Adaptado de Zhang, Li e Yang (2019) (a) e Derakhshani et al. (2019) (b).

A última atualização oficial do algoritmo da família YOLO é YOLOv4, que apresenta melhoria de 12% na precisão média (quando comparado à YOLOv3), não impactando negativamente no quesito tempo para tal melhoria (BOCHKOVSKIY; WANG; LIAO, 2020a). Há uma atualização ainda não oficial, YOLOv5, contudo, resultados preliminares indicam que a precisão média da versão YOLOv4 é ainda superior (NELSON; SOLAWETZ, 2020).

3.3.2 YOLOv4

Como já mencionado, o modelo YOLOv4 é um detector promissor por equilibrar precisão e velocidade. De acordo com Bochkovskiy, Wang e Liao (2020a), sua arquitetura consiste em:

- ***Backbone*:** CSPDarknet53 (ou apenas Darknet);
- ***Neck*:** SPP e *Path Aggregation Network* (PANet); e
- ***Head*:** YOLOv3.

Desta forma, como ilustrado na Figura 18, YOLOv4 utiliza o *backbone* Darknet para extrair o mapa de característica das imagens de entrada. Um aspecto relevante deste *backbone* é a separação do mapa de características em duas partes, para que apenas

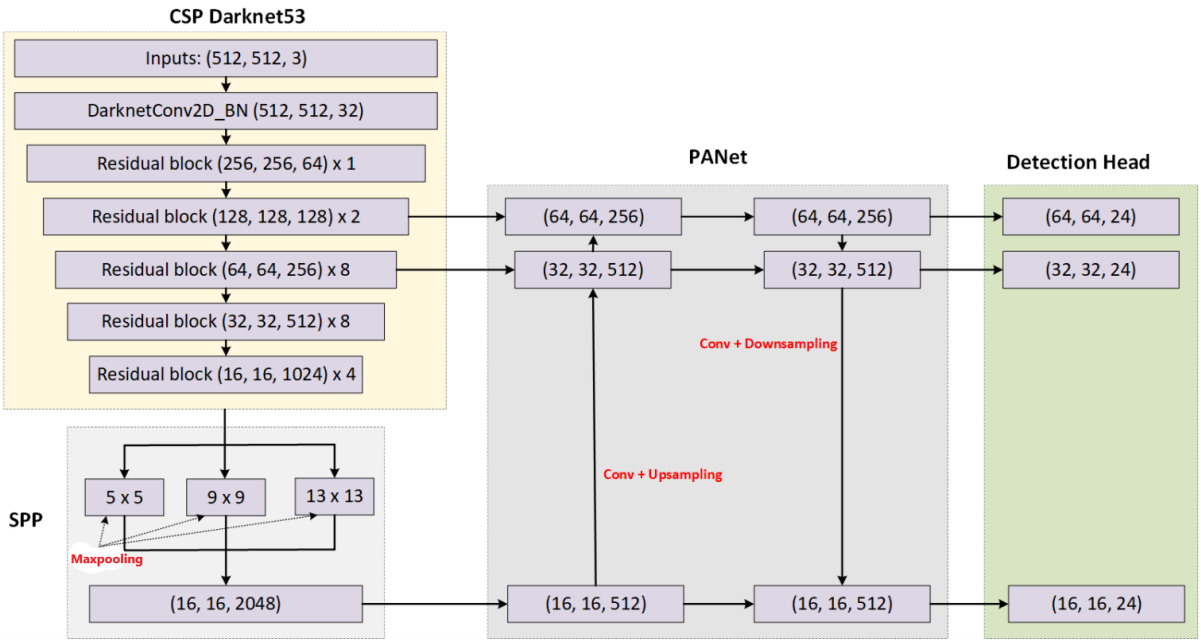


Figura 18 – Arquitetura geral do modelo YOLOv4.
Fonte: Adaptado de Chen, Zhong e Zhang (2021).

uma delas passe por uma camada densa de convolução, reduzindo assim o número de parâmetros (CHEN; ZHONG; ZHANG, 2021).

Adicionalmente, visando melhorar a detecção de objetos em diferentes escalas, YOLOv4 emprega o bloco SPP e o PANet, sendo o primeiro bloco responsável pela extração de características em vários níveis e o segundo pela concatenação dos mapas de características vizinhos vindos do fluxo descendente do *backbone* (CHEN; ZHONG; ZHANG, 2021).

Por fim, incorporando o modelo YOLOv3 como bloco de predição (*Detection Head*), YOLOv4 realiza a detecção em três escalas diferentes, resultando nas coordenadas das caixas delimitadores, juntamente com a pontuação de confiança por classe (BOCHKOVSKIY; WANG; LIAO, 2020a).

YOLOv4 inclui ainda dois pacotes de técnicas para facilitar o treinamento e melhorar o desempenho das detecções, *Bag of Freebies* e *Bag of Specials*. Em resumo, o conjunto de técnicas *Bag of Freebies* altera a estratégia ou o custo de treinamento para aprimorar a precisão do modelo. O conjunto *Bag of Specials* contém *plugins* e módulos de pré-processamento que aumentam o custo de inferência em uma pequena quantia, mas podem melhorar drasticamente a precisão do detector de objetos (BOCHKOVSKIY; WANG; LIAO, 2020a). Ambos são utilizados no *backbone* e na detecção (*head*).

Essa combinação de arquitetura, métodos e técnicas conferem à esta CNN resultados que permitem consolidá-la no estado da arte dos detectores de objetos.

3.4 Ferramentas Computacionais

De forma geral, a YOLOv4 e outros detectores de objetos foram aprimorados no aspecto velocidade e desempenho graças à disponibilização de GPUs, ampliando o poder computacional de processamento e de treinamento das CNNs (LIU et al., 2020). De acordo com TESLA (2017), as principais estruturas de Aprendizagem Profunda ainda são desenvolvidas em GPUs do tipo *hardware* NVIDIA. Contudo, soluções do tipo *cloud* vem se tornando cada vez mais populares (NVIDIA, 2018).

As principais vantagens no desenvolvimento de CNN em ambiente *cloud* estão relacionadas à eliminação de custos de manutenção e de reposição de *hardware* devido à depreciação e flexibilidade para obtenção de maior ou menor poder de processamento (LAWRENCE et al., 2017). De acordo com Carneiro et al. (2018), as principais plataformas *cloud* pagas com disponibilidade de GPUs configuradas para Aprendizagem Profunda são Amazon, Intel, Azure e Google Cloud.

Existe ainda o Google Colaboratory, ou apenas Colab, um serviço do tipo *cloud* gratuito, dotado de GPU, para finalidades de educação e pesquisa em Aprendizado de Máquina. Apesar de recente, uma busca realizada no Scopus[2] em 14/05/2021 revelou que as publicações acadêmicas acerca da utilização desta ferramenta cresceram de 8, em 2018, para 76 em 2020. Em 2021, até o momento de realização da busca, a base já registrava 29 publicações.

Uma das principais vantagens deste serviço é que o mesmo é conectado ao serviço de armazenamento de dados do Google Drive, além de conter as principais bibliotecas de Inteligência Artificial e Aprendizado de Máquina como o TensorFlow, o Matplotlib, o Keras e o OpenCV. Adicionalmente, o Colab é baseado em Jupyter Notebooks, incluindo Python 2 e 3 (BISONG, 2019; CARNEIRO et al., 2018).

Outro aspecto interessante é a possibilidade de utilizar o GPU acelerado por *Compute Unified Device Architecture* - Arquitetura de Dispositivo de Computação Unificada (CUDA)[3] buscando otimizar o desempenho da rede. O Colab também inclui a biblioteca CUDA *Deep Neural Network* (cuDNN), que permite implementar camadas de convolução de propagação direta (*feedfoward*) e de retroalimentação (*recurrent*), *backpropagation*, *pooling*, ativação, entre outras. Pesquisadores utilizam frequentemente o cuDNN para aceleração do treinamento das estruturas de aprendizagem profunda (NVIDIA, 2020b).

[2] Consiste numa base de indexação de títulos científicos da Elsevier que cobre diversas áreas de pesquisa, além de ser reconhecida como uma das mais abrangentes (MONGEON; PAUL-HUS, 2016). Esta base de indexação pode ser consultada em: <https://www.scopus.com/>.

[3] Consiste numa plataforma de computação paralela e modelo de programação para computação geral em GPUs (NVIDIA, 2020a).

3.5 Trabalhos Correlatos

Vários estudos que utilizam técnicas de visão computacional vêm sendo desenvolvidos (ARAI; KAPOOR, 2019). Dentre as aplicações mais populares, como análise de imagens médicas e reconhecimento facial e de texto, umas das mais recorrentes trata da detecção de objetos para viabilizar a dirigibilidade de veículos autônomos (UCAR; DEMIR; GUZELIS, 2017). O Quadro 1 inclui os principais artigos de revisão do estado da arte sobre o tema, publicados desde 2000, dentre os quais é possível verificar a predominância desta aplicação.

Apesar de não objetivar especificamente a detecção de feições de acessibilidade e mobilidade de pedestres, os trabalhos analisados no Quadro 1 utilizam-se de métodos que, posteriormente, foram incorporados a estes estudos (AHMETOVIC et al., 2017; AHMETOVIC et al., 2014; HARA et al., 2014; HARA; FROEHLICH, 2015; WELD et al., 2019). Isso se deve ao fato de que ambas as aplicações, dirigibilidade de veículos e mobilidade de pedestres, estão associadas à percepção do ambiente, sendo que no primeiro caso os principais alvos de detecção são ruas, outros veículos e pedestres, e no segundo caso, calçadas, rampas de acessibilidade e faixas de travessia. Em ambos os casos, o processo ocorre por o meio da captura de vídeos ou imagens ao nível do solo (JANAI et al., 2017).

Quadro 1 – Artigos de revisão abordando visão computacional para veículos autônomos

Título	**Ano**	**Nºde Citações**	**Conteúdo**	**Referência**
On road vehicle detection: a review	2006	830	Revisão acerca da visão computacional para sistemas de detecção de veículos na estrada	Sun, Bebis e Miller (2006)
Monocular pedestrian detection: survey and experiments	2009	861	Comparação entre três detectores de pedestres	Enzweiler e Gavrila (2009)
Survey of pedestrian detection for advanced driver assistance systems	2010	704	Artigo de revisão sobre detecção de pedestres para sistemas de assistência ao motorista	Geronimo et al. (2010)
Pedestrian detection: an evaluation of the state of the art	2012	1684	Avaliação detalhada dos detectores de imagens aplicáveis à deteção de pedestres	Dollar et al. (2012)
Deep learning	2015	17327	Uma introdução ao aprendizado profundo e aplicações	LeCun, Bengio e Hinton (2015)
Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving	2017	112	Aborda a detecção de objetos em relação à construção e reutilização de mapas de longo prazo em várias condições para veículos autônomos	Bresson et al. (2017)
A survey on deep learning: Algorithms, techniques, and applications	2018	84	Revisão abrangente do estado da arte da aprendizagem profunda para processamento de imagens, áudio e texto, incluindo aplicações inovadoras	Pouyanfar et al. (2018)
Object Detection with Deep Learning: A Review	2019	148	Levantamento histórico sobre a detecção de objetos com aprendizagem profunda, com foco em detecção facial e de pedestres	Zhao et al. (2019)
Deep Learning for Generic Object Detection: A Survey	2020	15	Uma Revisão bibliográfica e teórica sobre aprendizagem profunda para detecção de objetos genéricos	Liu et al. (2020)
Recent advances in deep learning for object detection	2020	3	Revisão sistemática da detecção de objetos abordando componentes de detecção, estratégias de aprendizado e aplicações	Wu, Sahoo e Hoi (2020)

Fonte: Elaborado pela autora

4 Imagens *Street-Level*

Este capítulo tem foco na identificação das características relevantes nas pesquisas e publicações relacionadas ao uso de imagens ao nível do solo para detecção de objetos, buscando-se trabalhos que se relacionam ao uso do GSV e mobilidade acessível.

4.1 *Street-Level*

O uso de imagens ao nível do solo, denominadas nas publicações como imagens *Street-Level* ou *Street View* (ou apenas panorâmicas), vem chamando a atenção nas aplicações de detecção de objetos. A Figura 19 apresenta o quantitativo de publicações envolvendo estas duas temáticas resultantes de uma busca realizada em 15/06/2020, na base de indexação Scopus[2]. Verifica-se um crescimento acentuado nas publicações a partir de 2008, ano em que foi identificada a primeira publicação com ambos os termos "detecção de objetos"e "*street view*"como palavras-chave e/ou termos constantes nos resumos destes documentos.

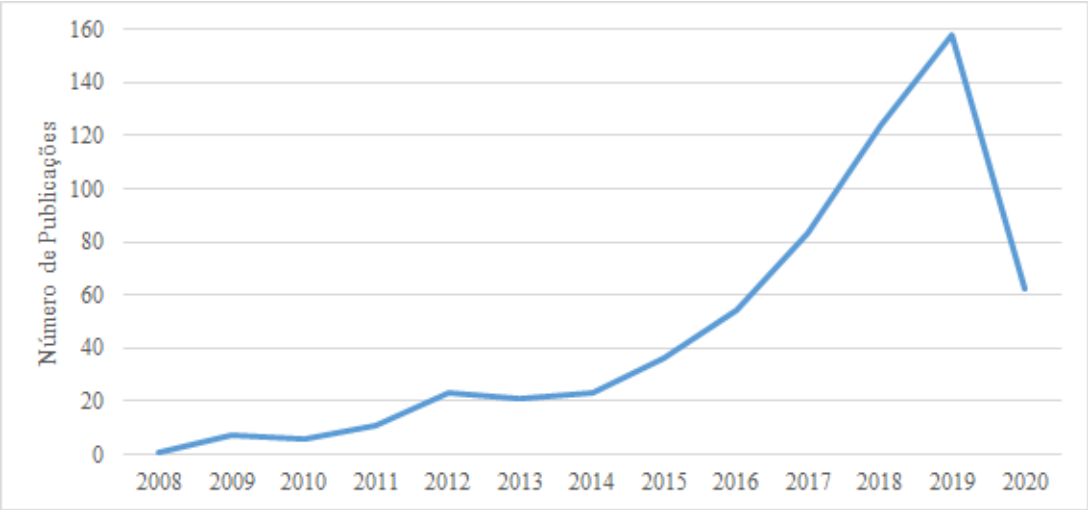


Figura 19 – Publicações que contém como palavras-chave "object detection"e "street view".
Fonte: Adaptado dos resultados da busca avançada realizada no Scopus em 15/06/2020

Além destes, outros termos recorrentes nas publicações referentes aos estudos que estão sendo desenvolvidos, reforçam a importância dos aspectos levantados no Capítulo 3 deeste trabalho, como pode ser observado na Figura 20.

Com a popularização da detecção de objetos a partir da percepção ao nível do solo, vários estudos passaram a ser desenvolvidos com as mais variadas aplicações: detecção de acidentes de trânsito a partir de câmeras de monitoramento (WANLI et al., 2018), monitoramento de vagas de estacionamento em centros comerciais (VAHTRA;

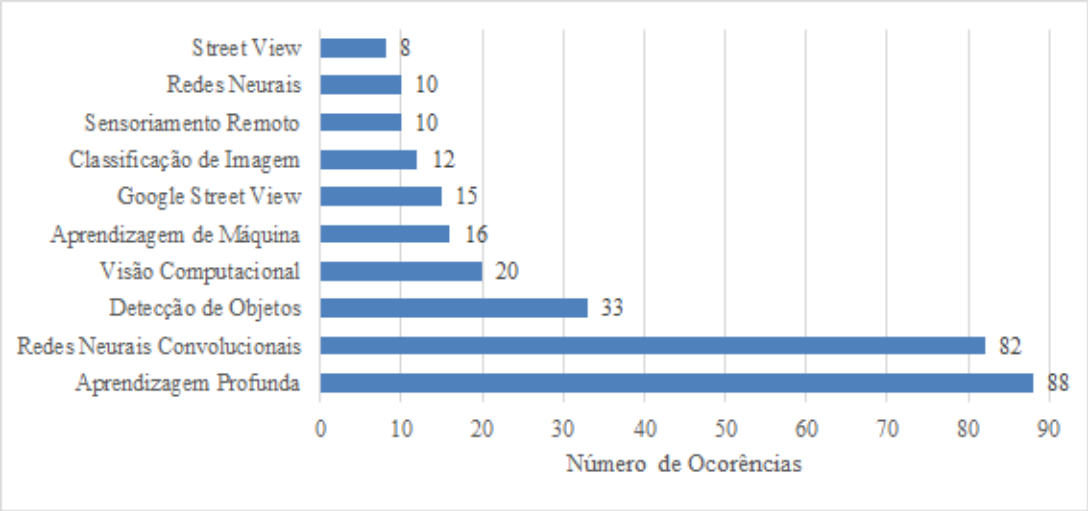


Figura 20 – Termos recorrentes nas publicações que contém como palavras-chave "object detection"e "street view".Fonte: Elaborado pela autora.

ANBARJAFARI, 2019), estimativa de volume de pedestres (CHEN et al., 2020), previsão de preços de imóveis (LAW; PAIGE; RUSSELL, 2019) e monitoramento de vegetação urbana (LI; RATTI; SEIFERLING, 2017; LI et al., 2015).

O que se observa em comum nestes estudos é o uso do GSV como fonte primária de dados. Este fato somado ao fato de que, como pode ser observado na Figura 20, o termo "*Google Street View*" tem maior destaque do que o termo "*Street View*", o qual é mais genérico ao referir-se à imagens panorâmicas ao nível do solo, reforça a importância deste recurso como fonte de dados de imagens.

Especificamente no que diz respeito à trabalhos com foco em mobilidade de pedestres, destacam-se os trabalhos de Ahmetovic et al. (2017) e Ahmetovic et al. (2014), que propuseram a identificação da travessia de pedestres a partir de detecção de objetos em imagens de satélite e panoramas do GSV. Nesta mesma linha de pesquisa, Hara et al. (2014) e Hara e Froehlich (2015) desenvolveram um método semi-automático de detecção de rampas de acessibilidade através de Aprendizado de Máquina e Visão Computacional, utilizando para tal a rotulação manual das rampas através de VGI em imagens do GSV.

4.2 Detecção de Objetos no Google Street View

Dentre os trabalhos descritos, a detecção se dá por diversas CNNs, conforme exposto no Capítulo 3, na descrição dos modelos de *Region Proposal Based Framework* e *Regression/ Classification Based Framework*. Visando identificar os principais métodos utilizados na detecção de objetos em imagens do tipo *Street View* e em mobilidade pedestres, inclusive acessibilidade em calçadas, foi feito um levantamento dos métodos utilizados nas publicações resultantes das seguintes buscas:

I. **Busca 1**: *Object Detection* (detecção de objetos) and *Street View*; e

II. **Busca 2**: *Object Detection* (detecção de objetos) and *Sidewalk* (calçada) ou *Ramp* (rampa) ou *Wheelchair* (cadeira de roda).

As principais CNNs resultantes da busca, bem como a frequência com que são utilizadas nas publicações analisadas – até a data da investigação, 15/06/2020 – podem ser observadas na Figura 21.

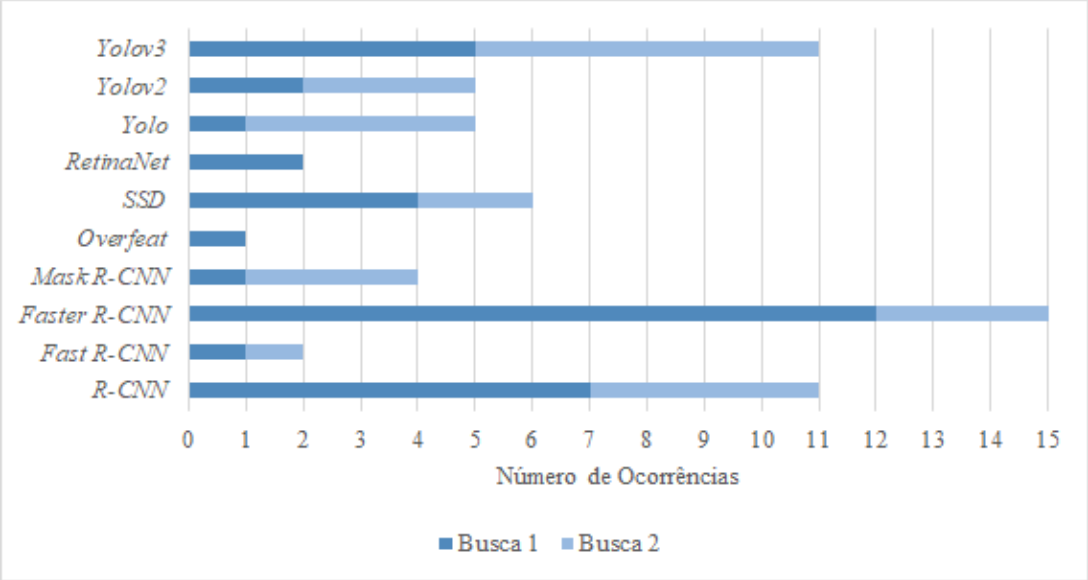


Figura 21 – Detectores de objetos encontrados nas publicações resultantes da Busca 1 e Busca 2
Fonte: Elaborado pela autora com base nos resultados da busca avançada realizada no Scopus em 15/06/2020

Das CNNs mais utilizadas, destacam-se a *Faster* R-CNN, a R-CNN e a YOLOv3. Um aspecto interessante é que a *Faster* R-CNN e a R-CNN, ambos do tipo *Region Proposal Based Framework*, são predominantes nas publicações resultantes da Busca 1, relacionadas à imagens do tipo *Street View*. Por outro lado, os modelos da família YOLO, do tipo *Regression/ Classification Based Framework*, possuem grande destaque nas publicações resultantes da Busca 2, relacionadas à mobilidade e acessibilidade, principalmente a YOLOv3.

Conforme demonstrado na Figura 21, a YOLOv3 chama a atenção pelo fato de ser uma das principais CNNs utilizadas nos trabalhos investigados, além de ser bastante empregada nas publicações relacionadas à ambas as buscas: imagens do tipo *Street View* e mobilidade/acessibilidade. Buscando melhorar este cenário, Kumar et al. (2019) propuseram uma adaptação da YOLOv3 para detecção de objetos em vídeos 360°, os quais se assemelham às imagens panorâmicas do GSV.

Apesar do aprimoramento em termos de precisão desenvolvido na atualização da YOLOv4, a mesma ainda não aparece dentre os principais métodos utilizados nas

publicações relacionadas à *Street View* e mobilidade acessível. Provavelmente isso se deve ao fato da atualização ser muito recente (abril de 2020) (BOCHKOVSKIY; WANG; LIAO, 2020a). Mesmo não sendo a principal CNN dentre imagens *Street View*, estudos indicam que a YOLOv4 é a mais consolidada em termos gerais e isso vai muito além das melhorias em velocidade e precisão, se deve também ao fato de que esta versão foi modificada para detectar objetos pequenos (BOCHKOVSKIY; WANG; LIAO, 2020a).

4.3 Características gerais das imagens GSV

As panorâmicas do GSV são imagens circundantes de projeção esférica, 360° na horizontal e 180° na vertical, resultante de um mosaico de oito imagens originais capturadas sequencialmente, conforme demonstrado na Figura 22 (a e b).

Para remover a distorção radial acentuada causada pela lente da câmera, as capturas são sobrepostas horizontalmente para a composição do mosaico (a porcentagem de sobreposição é de cerca de 28,8%) (KRYLOV; KENNY; DAHYOT, 2018). A Figura 22 (d e e) exemplificam esta sobreposição. Apesar desta sobreposição, ao aplicar técnicas de geometria de perspectiva à panorâmicas, acumulam-se erros em virtude da variação de escala e resolução espacial desuniforme internas (RAY, 2002).

Outra característica relevante das panorâmicas do GSV é que as mesmas possuem *geotagging* (em tradução livre, marcação geográfica), que, de acordo com Humphreys e Liao (2011) é um vínculo de "*uma palavra, frase ou imagem a um local físico específico, usando um sistema de referência geográfica padrão*". Na prática, isso significa que o centro perspectivo da panorâmica tem coordenadas geográficas. No caso GSV o sistema de referência geográfica é o WGS-84 e a acurácia posicional varia ente 1 e 5 metros (GOOGLE MAPS PLATFORM, 2020).

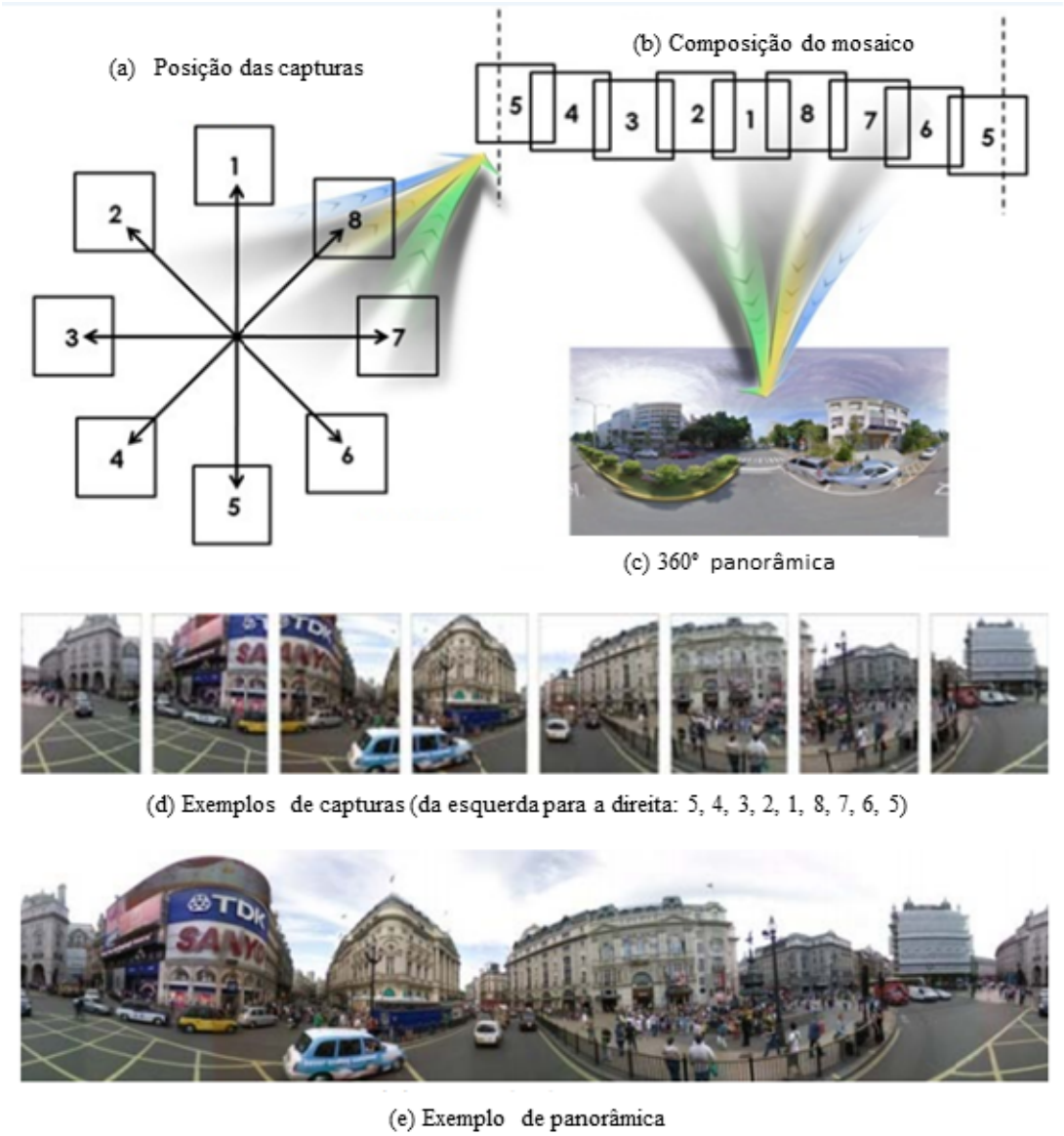


Figura 22 – Composição de uma panorâmica do GSV
Fonte: Adaptado de Tsai e Chang (2013)

4.4 Trabalhos Correlatos

Considerando as características das panorâmicas do GSV e a possibilidade de detecção de objetos nas mesmas, diversos estudos vem sendo desenvolvidos, incluindo a dirigibilidade de veículos autônomos, os quais lidam com o posicionamento relativo de objetos identificados em imagens e/ou vídeos panorâmicos (KRYLOV; KENNY; DAHYOT, 2018; SUN et al., 2013; KUMAR et al., 2019).

Destacam-se também aplicações em planejamento urbano, monitoramento e mapeamento de infraestruturas e serviços baseados em localização (KRYLOV; KENNY; DAHYOT, 2018; KUMAR et al., 2019; HARA; FROEHLICH, 2015; WELD et al., 2019; HUMPHREYS; LIAO, 2011). Muitos destes estudos visam estimar a localização de objetos em coordenadas

geográficas, apropriando-se da característica de *geotagging* encontrada nas panorâmicas do GSV.

O Quadro 2 relaciona as principais publicações neste contexto, reafirmando as perspectivas e possibilidades de aplicações a partir da detecção de objetos em imagens a nível do solo.

Quadro 2 – Relação de trabalhos relacionados à localização em imagens panorâmicas

Título	**Ano**	**Nºde Citações**	**Conteúdo**	**Referência**
Framework for natural landmark-based robot localization	2012	10	*Landmark-based*	Montero et al. (2012)
Improving urban vehicle localization with traffic sign recognition	2015	11	*Landmark-based*	Welzel, Reisdorf e Wanielik (2015)
Ocrapose: An indoor positioning system using smartphone/tablet cameras and OCR-aided stereo feature matching	2015	6	*Landmark-based*	Sadeghi, Valaee e Shirani (2015)
Pose estimation based on four coplanar point correspondences	2009	13	*Landmark-based*	Yang et al. (2009)
Semi-supervised logo-based indoor localization using smartphone cameras	2014	5	*Landmark-based*	Sadeghi, Valaee e Shirani (2014a)
An analytic solution for the perspective 4-point problem	1989	249	*Landmark-based*	Horaud et al. (1989)
Image based localization in indoor environments	2013	68	*Image retrieval-based*	Horaud et al. (1989)
A weighted KNN epipolar geometry-based approach for vision-based indoor localization using smartphone cameras	2014	26	*Image retrieval-based*	Sadeghi, Valaee e Shirani (2014b)
Three-dimensional positioning from Google street view panoramas	2013	13	*Image retrieval-based*	Tsai e Chang (2013)
Coarse-to-fine vision-based localization by indexing scale-invariant features	2006	111	*Image retrieval-based*	Wang, Zha e Cipolla (2006)
Visual localization by linear combination of image descriptors	2011	36	*Image retrieval-based*	Torii, Sivic e Pajdla (2011)
2DTriPnP:*A Robust Two-Dimensional Method for Fine Visual Localization Using Google Streetview Database*	2017	8	*Image retrieval-based*	Sadeghi, Valaee e Shirani (2017)
Automatic Discovery and Geotagging of Objects from Street View Imagery	2018	15	*Image retrieval-based*	Krylov, Kenny e Dahyot (2018)
Geometric context from a single image	2005	479	*Image retrieval-based*	Hoiem, Efros e Hebert (2005)
Recognizing scene viewpoint using panoramic place representation	2012	104	*Image retrieval-based*	Xiao et al. (2012)
Putting Objects in Perspective	2006	321	*Image retrieval-based*	Hoiem, Efros e Hebert (2006)
Feature Positioning on Google Street View Panoramas	2012	8	*Image retrieval-based*	Tsai e Chang (2012)
Monocular urban localization using street view	*2016*	5	*Image retrieval-based*	Yu et al. (2016)
Deeper Depth Prediction with Fully Convolutional Residual Networks	2016	468	*Image retrieval-based*	Laina et al. (2016)

Fonte: Elaborado pela autora

5 Materiais e Métodos

Segundo Gerhardt e Silveira (2009), esta pesquisa é de natureza aplicada, podendo, do ponto de vista dos objetivos, ser considerada descritiva, ao buscar informações para compreender o fenômeno estudado, e exploratória, ao proporcionar familiaridade com o problema, visando desenvolver uma hipótese. Quanto à abordagem do problema é quantitativo, pois os resultados da pesquisa podem ser quantificados e, quanto aos procedimentos, trata-se de uma pesquisa experimental.

A fase preliminar do método consiste na fundamentação teórica e pesquisa bibliométrica-bibliográfica desenvolvidas nos Capítulos 2, 3 e 4 deste trabalho. Foram realizadas buscas no portal de periódicos da CAPES e na base de indexação *Scopus*[2]. Também foram realizadas buscas em outras fontes variadas como instituições de pesquisa estatística e censitária, relatórios técnicos e plataformas digitais diversificadas. Os dados brutos do Censo Demográfico constantes no Capítulo 2 foram processados em linguagem de programação R, no *software* RStudio.

A partir daí, a abordagem proposta para detecção de rampas de acessibilidade é composta pelas etapas mencionadas na Figura 23, que tem início pela obtenção de imagens do GSV para criar um banco de imagens. A partir daí, foi realizada a rotulação manual das rampas nestas imagens e foram aplicados alguns processos de pré-processamento, criando assim uma base de dados para treinamento das CNNs. Posteriormente, foram executados e validados diversos treinamentos, utilizando diversos parâmetros, visando averiguar a melhor CNN. Com a CNN treinada e validada, foi possível testar o detector de rampas de acessibilidade.

Para o desenvolvimento destas etapas foram utilizados os seguintes materiais:

- **R Studio 1.4 e R 4.0.2:** para manipulação e análise dos dados censitários;
- **QGIS 3.10:** para *download* e manipulação de dados espaciais;
- **Google Street View API:** para obtenção das panorâmicas do GSV;
- **Ferramenta Python Google Street View Panorama Image Downloader:** para *download* das panorâmicas do GSV;
- **LabelImg:** para rotulação das rampas de acessibilidade nas imagens;
- **Conversor Python de "*.xml"para "*.txt":** para a conversão dos arquivos de rótulos para o formato compatível com YOLO;

- **Roboflow:** para implementação de técnicas de pré-processamento (redimensionamento e *Data Augmentation*);
- **Ferramenta Python de *Tiling*:** para implementação da técnica de *Tiling*;
- **Google Drive:** para armazenamento de arquivos dos treinamentos de CNNs (entrada, *backup* e saída), inclusive arquivos utilizados nos testes;
- **Google Colab (Python *Notebook* com GPU Tesla T4, CUDA 11020, cuDNN 7.6.5 e OpenCV 3.2.0):** para execução dos treinamentos e testes das CNNs;
- **YOLOv4 (com Darknet AlexeyAB):** para estruturação e execução dos treinamentos e testes das CNNs.

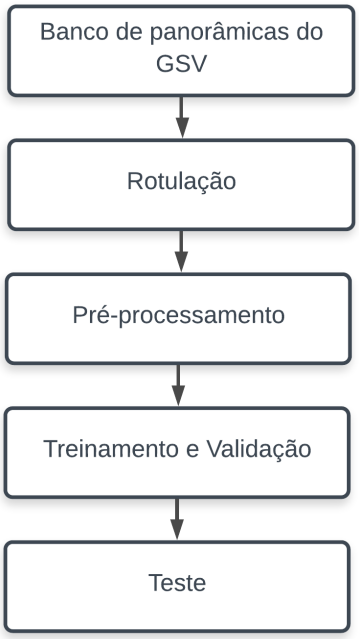


Figura 23 – Fluxograma descrevendo as etapas do método
Fonte: Elaborado pela autora

5.1 Banco de imagens para treinamento

As imagens do GSV foram obtidas com o uso do API do Google Street View, o qual exige um cadastro para posterior fornecimento de uma chave de acesso para download das imagens (GOOGLE MAPS PLATFORM, 2021). A aquisição das imagens se deu a partir da adaptação da ferramenta Python "Google Street View Panorama Image Downloader", desenvolvida por (LETCHFORD; ZARZELLI; BERRIEL, 2018) (as adaptações realizadas na ferramenta constam no Apêndice A). Além do download das imagens, as adaptações realizadas permitiram a obtenção de parâmetros adicionais como data da imagem, coordenada geográfica de captura (latitude, longitude) e ID da panorâmica.

As panorâmicas do GSV possuem algumas propriedades específicas, como:

- ***Size* (Tamanho):** tamanho da imagem em pixel, sendo o máximo de 1.664x832;
- ***Location* (Localização):** coordenada geográfica – latitude e longitude – em WGS84, em formato decimal e indicação de Sul/Oeste com sinal negativo;
- **FOV (*field of vision*, campo de visão):** varia de 0 à 120 graus, sendo que quanto maior o valor, menor o *zoom*;
- ***Head* (Rotação):** trata-se do ângulo horizontal de rotação em graus da câmera em relação ao Norte, portanto, varia de 0 a 360 graus;
- ***Pitch* (Inclinação):** indica o ângulo vertical de inclinação em graus da câmera em relação veículo que a suporta. Varia de -90 a 90 graus, sendo que os valores positivos indicam inclinação para cima e valores negativos para baixo, podendo-se assumir que o valor 0 representa o plano horizontal.

Na Figura 24 são apresentados exemplos dos parâmetros descritos acima.

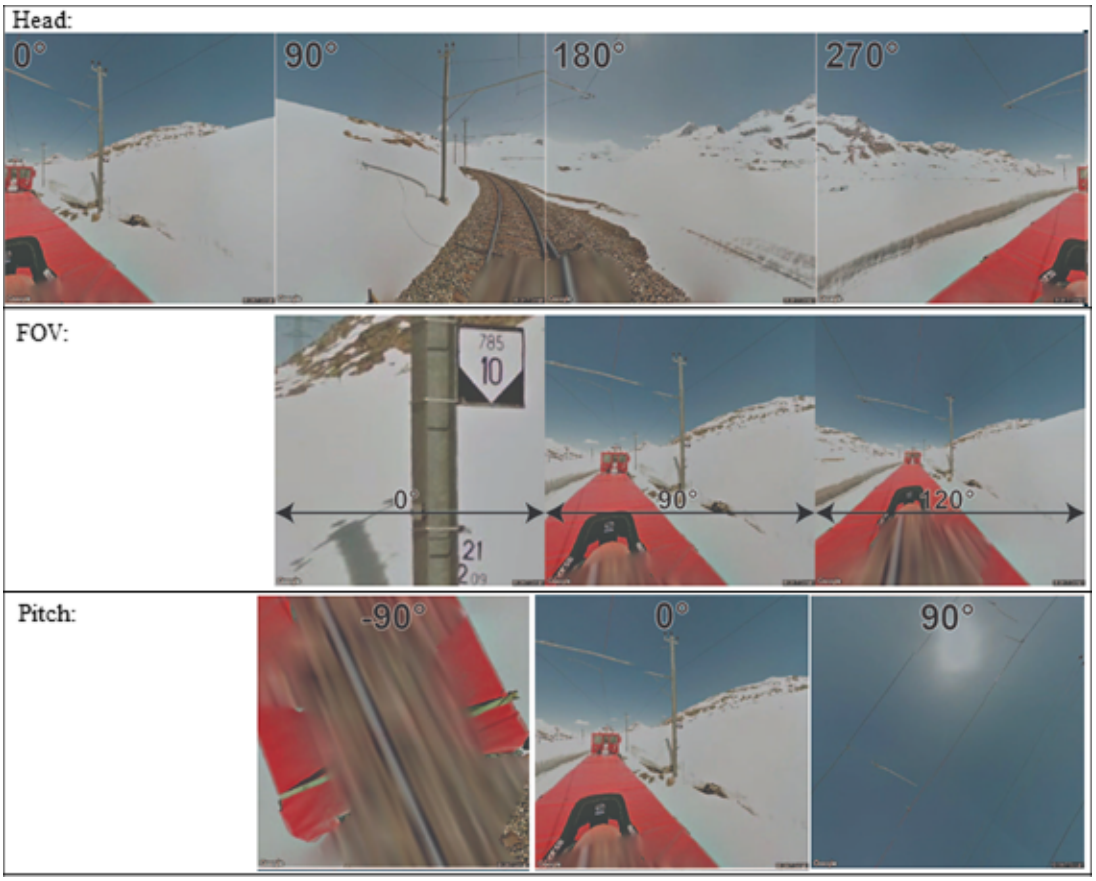


Figura 24 – Exemplo de imagens conforme parâmetros de aquisição do GSV
Fonte: Adaptado de GOOGLE MAPS PLATFORM (2021)

No caso das adaptações realizadas na ferramenta *"Google Street View Panorama Image Downloader"*, os parâmetros selecionados foram *size* de 1.664x832 pixels, *pitch* de 0 graus, FOV de 90 graus e *head* referente à panorama completa (360 graus).

A principal entrada desta ferramenta é a coordenada da panorâmica (latitude, longitude), sendo que foram consideradas as coordenadas das intersecções da malha viária, uma vez que as rampas de acessibilidade são mais propensas a ocorrer nestes locais. As intersecções de malha viária georreferenciadas foram obtidas no OSM e processadas no *software* QGIS.

Outra precaução para obter panorâmicas com o maior número de amostras de rampas possível foi utilizar coordenadas de intersecções aleatórias, inseridas em regiões com maior concentração de calçadas com rampas. Está informação foi obtida no Censo Nacional 2010 (IBGE, 2010a), nas características do entorno, e processada no *software* QGIS.

O objetivo final desta etapa é disponibilizar um conjunto de dados rotulados para futuras aplicações em visão computacional e, portanto, considerando as recomendações técnicas de 1.000 imagens por classe para treinamento da CNN (BOCHKOVSKIY; WANG; LIAO, 2020a), optou-se pelo número mínimo de 1.250 rótulos de rampas. Esse número foi selecionado prevendo 80% de rótulos para treinamento (mínimo de 1.000) e 20% para validação (cerca de 250) – atentando-se para o fato de que as panorâmicas do GSV podem conter várias rampas, ou nenhuma, por esta razão foi estipulado o número mínimo de rótulos (e não de imagens).

De acordo com Yao et al. (2020) e Russakovsky et al. (2015) o processo de construção de uma banco de imagens envolve três preocupações centrais: escalabilidade, diversidade e acurácia. Escalabilidade está relacionada à distribuição balanceada do número de imagens candidatas por classe de objeto, o que não se aplica no caso de se ter apenas uma classe (rampa de acessibilidade).

Diversidade se refere à variabilidade de aparências, pontos de vista e planos de fundo destes objetos. Tratando-se de panorâmicas do GSV, diversidade ocorre naturalmente devido à variação de distâncias e ângulos no momento da captura dos objetos e, também, à grande diversidade de planos de fundo e oclusões aos quais os objetos estão sujeitos (SHAPIRO, 2018).

Yao et al. (2020) e Russakovsky et al. (2015) descrevem acurácia de formas diferentes: acurácia dos rótulos e acurácia da busca. Russakovsky et al. (2015) se referem à acurácia do rótulo, ou acurácia humana, como a habilidade humana de identificar e delimitar determinado objeto. Por se tratar de um atributo relacionado à rotulação, este aspecto será verificado no próximo item.

Yao et al. (2020), por outro lado, explicam que a acurácia da busca resulta dos erros de indexação nos sistemas de busca de imagens, por exemplo, termos de busca em sistemas de indexação como Google Images ou Flick. Estes casos requerem métodos auxiliares para redução de ruído e inclusão de sinônimos. Considerando que a aquisição de imagens no GSV não é baseada em sistemas de busca tradicionais, mas sim na inquisição de coordenadas geográficas (latitude, longitude) (GOOGLE MAPS PLATFORM, 2021), a acurácia da busca se resume à encontrar imagens candidatas que contenham a classe de objeto desejado.

Outro aspecto relevante é o recorte geográfico escolhido, no caso, rampas de acessibilidade das cidades brasileiras. Neste sentido, Souza (2019) enfatiza a não padronização das rampas em todo o país, resultando na necessidade da composição de um banco de imagens heterogêneo. Assim, para construir um conjunto que inclua variabilidade de padrões de rampas e considerando a maior ocorrência das mesmas em calçadas dos municípios da região Sudeste (conforme demonstrado no Capítulos 2), optou-se pela abrangência do estado de São Paulo para seleção de imagens candidatas à rotulação, equilibrando-se entre os estratos populacionais e priorizando os setores censitários com maior ocorrência de rampas nos passeios públicos.

Como já mencionado, o GSV não possui cobertura em todos os municípios brasileiros, principalmente nos de pequeno porte, o que reduz muito a acurácia da busca nestes municípios. Pensando nisso, os estratos populacionais até 20 mil habitantes foram agrupados. Esse estrato também foi escolhido como limite inferir devido ao fato de muitas legislações relacionadas ao tema utilizá-lo como referência, como o Estatuto da Cidade (BRASIL, 2001) e a PNMU (BRASIL, 2012).

Os métodos adotados – refinamento dos setores censitários com maior ocorrência de rampas e variabilidade entre porte de municípios – aliados às características das panorâmicas do GSV garantem uma ampla variedade de padrões de rampas, perspectiva, posição, ângulo, tamanho e desordem de fundo e oclusão.

5.2 Processo de Rotulação

Na literatura é possível identificar diversos *dataset* genéricos para detecção de objetos, como por exemplo COCO, que consiste num conjunto de dados de detecção rotulados, com 91 categorias de itens (LIN et al., 2014). Uma outra alternativa é o banco de imagens rotuladas do *Google Images*, no qual é possível procurar imagens com diversas classes de objetos já rotuladas.

Contudo, rampas de acessibilidade não fazem parte destes nem de outros *dataset* genéricos criados até o momento. Portanto, visando a disponibilização pública de um banco de imagens de rampas, optou-se pela classe denominada *Acessible Curb Ramps* - Rampas

de Acessibilidade (ACR) e foi realizada a rotulação manual das imagens adquiridas na etapa anterior. A rotulação foi realizada no software *LabelImg* (TZUTA, 2018).

Nesta ferramenta é possível rotular manualmente objetos em imagens e exportar um arquivo contendo a indicação da classe associada e as respectivas coordenadas do rótulo criado (TZUTA, 2018). Originalmente, o *LabelImg* gera um arquivo de rótulo para Imagenet, do tipo "*.xml", o qual precisa ser convertido para arquivo de rótulo da Darknet, do tipo "*.txt". Então, foi utilizado o *script* desenvolvido por Tashiev (2020)[4]. Na Figura 25 é ilustrado um exemplo de marcação manual e seu o respectivo arquivo de rótulo já convertido.

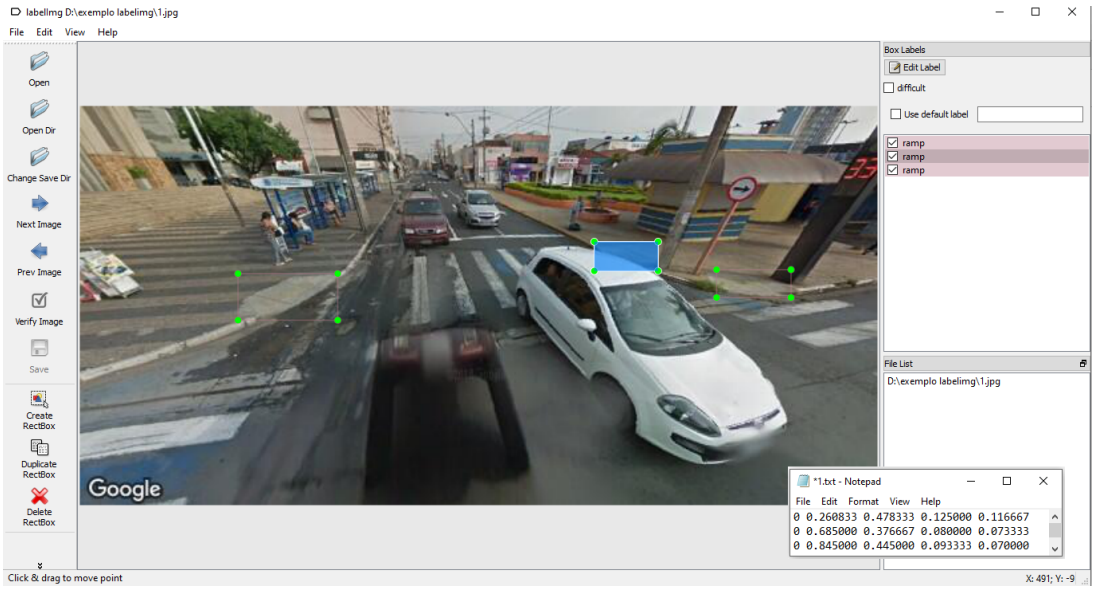


Figura 25 – Exemplo de marcação manual e arquivo de rótulo gerado
Fonte: Elaborado pela autora

A etapa de rotulação depende diretamente da capacidade humana de rotular objetos corretamente e terá impacto significativo na etapa de treinamento. Dois aspectos centrais precisam ser considerados: erro humano e discordância entre usuários quando o contexto de plano de fundo é confuso ou há oclusão parcial (RUSSAKOVSKY et al., 2015). Estes aspectos se referem à identificação do objeto em si ou ainda ao posicionamento da caixa delimitadora do mesmo. Visando mensurar o que Russakovsky et al. (2015) chamaram de acurácia da rotulação, um subgrupo de imagens rotulado por 10 pessoas diferentes foi analisado em termos de quantidade e qualidade.

A acurácia quantitativa de rampas rotuladas foi medida pelo nível de concordância (ou nível de confiança) do usuário – que é a razão entre o número de rampas anotadas e rampas totais. O nível de confiança varia de 0 a 1, sendo mais próximo de 1 quando os usuários concordam que o objeto é uma rampa.

A acurácia qualitativa da atribuição de caixas delimitadoras foi verificada acessando o desvio padrão das coordenadas do centroide das rampas rotuladas. Para entender esse

[4] Disponível em <https://github.com/Isabek/XmlToTxt>.

processo, precisamos entender como funciona o *LabelImg*: para cada objeto anotado ele retorna quatro coordenadas de pixels (*Xmin*, *Ymin*, *Xmax*, *Ymax*). Usando esses dados é possível calcular as coordenadas do centroide das caixas delimitadoras (*xC*; *yC*) utilizando-se as Equações 5.1 e 5.2.

$$xC = Xmin + (\frac{Xmax - Xmin}{2}) \tag{5.1}$$

$$yC = Ymin + (\frac{Ymax - Ymin}{2}) \tag{5.2}$$

E então, pode-se estimar o quão concordantes foram as delimitações realizadas pelos diferentes usuários calculando-se o desvio padrão (σ, Equação 5.3) para resultados obtidos nas Equações 5.1 e 5.2:

$$\sigma = \sqrt{\frac{\Sigma(xC - \overline{xC})^2}{(n-1)}} \tag{5.3}$$

sendo $\overline{xC}$ a média das coordenadas de centroide e *n* o número de rótulos para uma mesma rampa. A Equação 5.3 também é valida para os valores de *yC* e $\overline{yC}$.

Cabe enfatizar que, na avaliação qualitativa, foram consideradas apenas as coordenadas do centro do objeto pois o objetivo final é verificar se os usuários concordam com o posicionamento da caixa delimitadora. Logo, não foram considerados aspectos relativos ao tamanho da caixa delimitadora, uma vez que este não é o objetivo da detecção proposta nesta pesquisa (que é constatar se há rampa de acessibilidade, sendo irrelevante portanto o tamanho da mesma).

5.3 Técnicas de Pré-processamento

Uma prática comum na preparação de imagens para treinamento de CNNs é o uso de técnicas de pré-processamento, destacando-se o redimensionamento. Esta ferramenta é importante para lidar com problemas de limitação de memória computacional necessária para o treinamento, que ocorre quando as imagens são muito grandes. Contudo, esta técnica pode ser um problema quando os objetos alvo de detecção são muito pequenos, pois o redimensionamento se aplica à imagem e rótulos como um todo, fazendo com que os objetos sejam representados por menos pixels ainda.

No banco de imagens em questão podem ocorrer objetos muito pequenos em relação ao tamanho da imagem, sendo indicada a técnica de *Tiling* (Ünel; Özkalayci; Çiğla, 2019). Nesta técnica as imagens são subdivididas em imagens menores e, consequentemente, a quantidade de pixels que representam os objetos rotulados em relação ao tamanho da imagem é maior. Desta forma, ao mesmo tempo que os objetos menores são representados,

proporcionalmente, por mais pixels, a resolução da imagem não é alterada, mas o tamanho da mesma é reduzido (contribuindo também para a questão da limitação de memória computacional) (Ünel; Özkalayci; Çiğla, 2019).

Além destas técnicas de pré-processamento, *Data Augmentation* é uma categoria de técnicas que objetivam gerar novos exemplares de dados de treinamento a fim de aumentar a generalidade do modelo. Dentre as técnicas amplamente utilizadas, descritas por Shorten e Khoshgoftaar (2019), optou-se por dobrar o número de imagens originalmente rotuladas a partir da incorporação de:

- ***Flipping***: que se resume em espelhar as imagens vertical ou horizontalmente a fim de tornar o modelo menos sensível à orientação dos objetos. Optou-se apenas pelo *flip* horizontal, uma vez as rampas são capturadas nas panorâmicas *street-level* somente em posição ortostática;
- ***Shear***: trata da adição de uma variabilidade de perspectivas para tornar o modelo mais resiliente à diferentes posicionamentos da câmera em relação aos objetos. Nessa transformação, fixa-se um eixo e estica-se a imagem em um determinado ângulo, conhecido como ângulo de cisalhamento. O'Gara e McGuinness (2019) não recomendam um ângulo maior do que 20°, portanto, foi adotado um ângulo de 7° - assim como o utilizado por Zaworski (2018) em suas experimentações;
- ***Brightness***: ao adicionar variação no brilho nas imagens, o modelo tende a ser mais eficiente para detectar objetos com luminosidades variadas no momento da captura. Com base nos experimentos conduzidos por Zaworski (2018), adotou-se variação de 20% no brilho, tanto para mais, quanto para menos;
- ***Blur***: esta técnica se relaciona à possibilidade de identificar objetos em imagens com certo nível de desfoque, sendo indicada nos casos em que a câmera, os objetos ou ambos estão sujeitos à movimentos (NELSON, 2020). No caso da captura de rampas nas panorâmicas do GSV, a câmera é a que está em movimento, contudo, como a mesma já é preparada para lidar com esta questão, optou-se por desfocar apenas 1 pixel em algumas imagens a serem utilizadas no treinamento.

Para fins de comparação, optou-se por elaborar bancos de dados com diferentes técnicas de pré-processamento. Para o redimensionamento das imagens e implementação das técnicas de *Data Augmentation* mencionadas, utilizou-se a ferramenta Roboflow em sua versão gratuita (ALEXANDROVA; TATLOCK; CAKMAK, 2015). Para a implementação de *Tiling*, foi adotado o *script* Python desenvolvido por Neskorozhenyi (2021)[5].

[5] Disponível em <https://github.com/slanj/yolo-tiling>.

5.4 Treinamento

A partir da composição dos bancos de imagens rotuladas, pode-se implementar a etapa de treinamento da CNN. Inicialmente, é necessária a criação de dois arquivos: *"*.names"* e *"*.data"*. O primeiro deles, neste trabalho denominado *"obj.names"*, contém o nome das classes de objetos a serem detectados, neste caso ACR – mesma classe utilizada na etapa de rotulação. O outro, identificado como *"obj.data"*, contém respectivamente, em cada linha, o número de classes, o diretório das imagens e seus respectivos rótulos para treino, o diretório das imagens e seus respectivos rótulos para validação, diretório com o arquivo que contém as classes (*"*.names"*) e diretório de destino do arquivo *backup* de treinamento (vide Figura 26).

```
classes = 1
train = data/train.txt
valid = data/test.txt
names = data/obj.names
backup = /mydrive/yolov4/backup
```

Figura 26 – Conteúdo do arquivo obj.data
Fonte: Elaborado pela autora

A YOLOv4 foi a CNN escolhida neste projeto. Para tanto, a *Application Programming Interface* - Interface de Programação de Aplicativos (API) utilizada foi a Darknet AlexeyAB, de Bochkovskiy, Wang e Liao (2020b), tratando-se de uma versão melhorada da Darknet original desenvolvida pelos criadores da YOLOv3 (VAHTRA; ANBARJAFARI, 2019). O algoritmo YOLOv4 pode ser implementado de diferentes formas, sendo a Darknet umas das estruturas de rede neural de código aberto mais utilizada como API desta CNN. Ao todo, foram utilizadas 53 camadas de rede neural Darknet, além de 53 camadas adicionais para detecção (REDMON; FARHADI, 2018).

Apesar de não ser essencial para o caso de treinamento de um detector de objetos customizado, a incorporação de pesos convolucionais pré-treinados pode otimizar o tempo de treinamento e pode impactar positivamente na acurácia final (HE; GIRSHICK; DOLLAR, 2019). Desta forma, foram realizados treinamentos utilizando o arquivo de pesos "yolov4.conv.137" [6]. Para fins de comparação, também foi realizado um treinamento inicial sem o emprego dos pesos convolucionais pré-treinados.

Além disso, ainda é necessário criar dois arquivos "*.txt":

- **train.txt**: este arquivo deve listar o diretório das imagens e rótulos a serem utilizadas para treino; e

[6] Adquirido no site de seu criador <https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal /yolov4.conv.137>. Este arquivo de pesos foi pré-treinado para as classes do *dataset* genérico COCO.

- **test.txt**: este arquivo deve listar o diretório das imagens e rótulos a serem utilizadas para validação.

Estes arquivos foram criados a partir do *script* exibido no Anexo A e foram utilizados na etapa de treinamento e validação.

Outro ponto fundamental para a etapa de treinamento trata-se de carregar a Darknet, procedimento condicionado pelos parâmetros inseridos no arquivo padrão *"yolov4-obj.cfg"*[6]. Diferentemente do arquivo de pesos, este arquivo é indispensável pois "*contém a estrutura interna da rede a ser treinada, incluindo todas as camadas, e também os parâmetros necessários para configurar a rede*"(CARATA et al., 2019).

De acordo com Carata et al. (2019) e Mahalleh, ALQutami e Mahmood (2019), é preciso configurar os seguintes parâmetros neste documento:

- *Width* e *Height*;
- Número de classes (N);
- Quantidade de Filtros;
- *Max_batches*;
- *Steps*;
- *Batch size* (tamanho do lote);
- Subdivisões.

Width e *height* se referem ao tamanho das imagens do banco de dados, em pixels. Optou-se por realizarem-se testes com o tamanho original das imagens (1.664x832 pixels), com as imagens redimensionadas em 50% (832x416 pixels) e com as imagens submetidas à *Tiling* (416x416 pixels). Já o número de classes é a quantidade de tipos de objetos para qual a CNN será treinada para detectar, que neste caso é apenas ACR.

De acordo com Bochkovskiy, Wang e Liao (2020b), a quantidade de filtros é obtida pela Equação 5.4:

$$Filtros = (N + 5) \times 3 \tag{5.4}$$

N = número de classes

que para a aplicação proposta é 18.

O mesmo autor descreve ainda o cálculo de *max_batches* a partir da Equação 5.5:

$$Max\_batches = N \times 2.000 \tag{5.5}$$

N = número de classes

sendo que o mínimo recomendado é 6.000, o qual foi adotado.

Os parâmetros *steps* provêm da Equação 5.6:

$$Steps = (0,8 \times Max\_batches), (0,9 \times Max\_batches) \tag{5.6}$$

logo, foram adotados os valores 4.800 e 5.400 (BOCHKOVSKIY; WANG; LIAO, 2020b).

O *batch size* (tamanho do lote) é um termo usado em aprendizado de máquina para referir-se ao número de exemplos de treinamento usados em uma iteração do algoritmo (DATA SCIENCE ACADEMY, 2021b). O mesmo autor, DATA SCIENCE ACADEMY (2021b), ao realizar experimentações com os *batch sizes* 64, 256 e 1.024 verificou empiricamente que tamanhos de lote menores têm uma dinâmica de treinamento mais rápida, mantendo-se a eficiência do modelo. Para a condução dos treinamentos propostos, definiu-se um *batch size* de 64.

O parâmetro de subdivisões é utilizado para dividir os dados de treinamento em pequenos lotes e, consequentemente, proporcionar o manejo da memória computacional disponível durante o treinamento. Paralelamente, as subdivisões (ou *mini-batches*) influenciam diretamente na velocidade do treinamento (DATA SCIENCE ACADEMY, 2021a). Segundo recomendações de Bochkovskiy, Wang e Liao (2020b) foram realizados testes com 16 e 32 subdivisões, sendo 16 subdivisões para as imagens de tamanho 1.664x832 pixels e 32 para as demais dimensões (devido à limitações de memória para processamento no *Google Colab*).

Depois da organização de todos os arquivos e parâmetros necessários, os treinamentos foram executados no Google Colab Python *Notebook*, utilizando o GPU disponibilizado gratuitamente, CUDA 10, cuDNN e OpenCV, pelo comando:

```
!./darknet detector train data/obj.data cfg/yolov4-obj.cfg yolov4.conv.137 -map
```

5.5 Validação

Bochkovskiy, Wang e Liao (2020b) recomendam um mínimo de 2.000 iterações por classe, sendo no mínimo 6.000 iterações por treinamento – independentemente da quantidade de classes. Logo, como o treinamento proposto contempla apenas 1 classe, um mínimo de 6.000 iterações são necessárias, sendo que a cada 1.000 iterações um arquivo *backup* de pesos é salvo no diretório indicado em *"obj.data"*. Métricas realizadas com base nestes treinamentos intermediários indicam o momento adequado de parar o treinamento ou se são necessárias mais imagens rotuladas para que a rede melhore as detecções (MAHALLEH; ALQUTAMI; MAHMOOD, 2019; BOCHKOVSKIY; WANG; LIAO, 2020b)

De acordo com Wu, Sahoo e Hoi (2020), as principais métricas utilizadas para avaliar a acurácia dos algoritmos de detecção são *Average Precision* - Precisão Média (AP) e F1-*Score*. Estas métricas têm como princípio básico relacionar objetos rotulados e objetos detectados pelo algoritmo. Na prática, a etapa de validação compara as detecções resultantes do algoritmo treinado com os rótulos do diretório "test.txt" demarcados pelo usuário.

Para compreender AP e F1-*Score*, é necessário o entendimento dos conceitos de *Intersection-over-Union* - Intersecção sobre a União (IoU), *Precision* (precisão) e *Recall* (revocação). De forma geral, pode-se afirmar que um bom detector de objetos é aquele capaz de localizar bem os objetos, classificá-los com precisão e que perde poucas detecções. Analogicamente, o primeiro aspecto se refere à IoU, o segundo à *Precision* e o terceiro à *Recall* (PINTO, 2019).

A IoU é a média da razão das intersecções entre as áreas rotuladas e detectadas sobre a união destas áreas (vide Figura 27) (WU; SAHOO; HOI, 2020), onde:

$$IoU = \frac{\textit{Área de Intersecção}}{\textit{Área de União}} \tag{5.7}$$

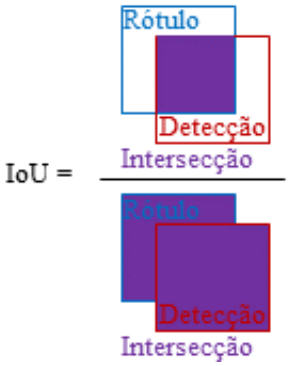


Figura 27 – Ilustração do conceito de IoU
Fonte: Adaptado de Bochkovskiy, Wang e Liao (2020b)

A partir de um determinado valor de IoU, geralmente 0,25, 0,50 ou 0,75, as detecções são classificadas em Verdadeiros Positivos (VP), Falsos Positivos (FP) e Falsos Negativos (FN), sendo:

- Verdadeiro Positivo (VP): representa uma detecção correta do objeto (quando há detecção e era para haver);
- Falso Positivo (FP): representa uma detecção errada do objeto (quando há detecção, mas não deveria haver);
- Falso Negativo (FN): representa uma detecção errada para o que não é o objeto (quando não há detecção, onde haveria).

A partir do enquadramento nestas classes é possível calcular a *Precision* e o *Recall* utilizando-se as Equações 5.8 e 5.9:

$$Precision = \frac{VP}{VP + FP} \tag{5.8}$$

$$Recall = \frac{VP}{VP + FN} \tag{5.9}$$

Por fim, o F1-*Score* é a média harmônica entre *Precision* e *Recall*, sendo calculado pela Equação 5.10:

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.10}$$

A AP, por sua vez, é resultante da área da curva suavizada do gráfico *Precision* x *Recall*, interpolado em 11 pontos, conforme exemplificado na Figura 28.

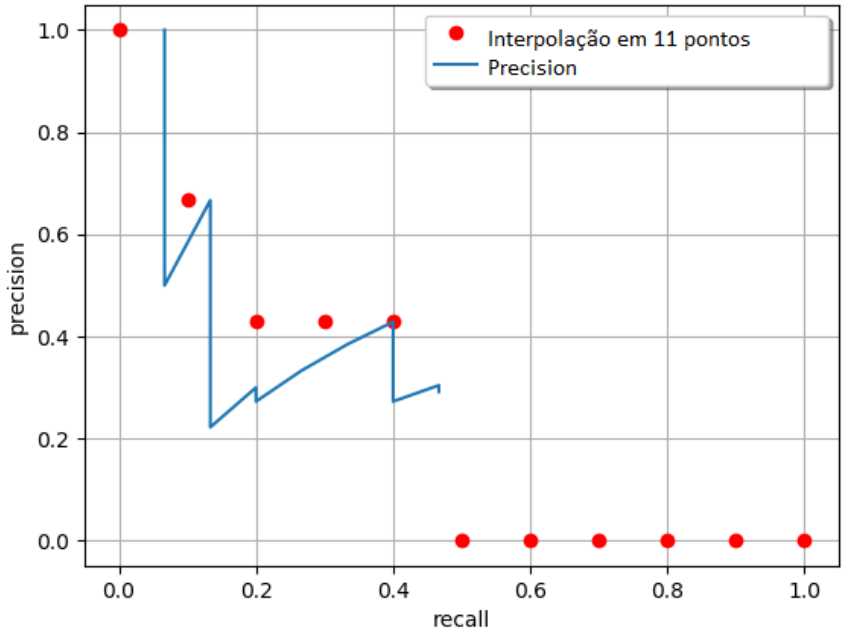


Figura 28 – Ilustração do conceito de AP.
Fonte: Amorin (s/d).

Ao fim do treinamento, o comando "map" retorna os valores das métricas mencionadas para os arquivos de pesos a cada 1.000 iterações de treinamento. Foram obtidas as métricas de validação para todos os treinamentos propostos.

5.6 Etapa de Teste

Uma vez validado o treinamento, o detector foi testado para as imagens de um grupo independente de imagens, ou seja, não utilizadas para treinamento e validação - que não

possuem rótulos. Estas imagens foram adquiridas pelo mesmo método utilizado no item 5.1 deste capítulo, contemplando equitativamente todos os estratos populacionais empregados neste trabalho (<20.000; ≥20.000 e <50.000; ≥50.000 e <100.000; ≥100.000 e <500.000; e ≥500.000 habitantes). Desta forma, o desempenho da CNN pode ser avaliado para diferentes padrões de rampas, permitindo inclusive uma investigação qualitativa acerca da influência desses padrões nas detecções.

A YOLOv4, por padrão, fornece como saída das detecções a caixa delimitadora, a porcentagem de precisão e o tamanho, em pixels, da caixa delimitadora. Estes dados foram então coletados para todas as imagens do conjunto de teste e foram aferidas visualmente a fim de averiguar se as detecções se tratavam de:

- VP: aferindo também com que precisão uma rampa identificada corretamente foi detectada. Apesar de Wu, Sahoo e Hoi (2020) afirmarem que é comum adotar-se um *threshold* [7] de 0,50 para testes, os valores encontrados em VP poderão indicar o *threshold* adequado para que a rede treinada encontre mais verdadeiros positivos;
- FP: aferindo com que precisão uma rampa identificada incorretamente foi detectada. Da mesma forma que no item VP, os valores encontrados em FP poderão indicar o *threshold* adequado para que a rede treinada ignore falsos positivos;
- FN: verificando se alguma rampa que deveria ter sido detectada não foi.

Também foram computadas a área, em pixels quadrados, das rampas detectadas, considerando que este aspecto é relevante para avaliar a performance do detector em relação ao tamanho dos objetos.

A etapa de teste também foi desenvolvida no Google Colab Python *Notebook*, utilizando-se os mesmos recursos do item 5.4 deste capítulo, contudo, para esta finalidade, o comando de detecção é dado por:

```
!./darknet detector test data/obj.data cfg/yolov4-obj.cfg /mydrive/yolov4/backup/yolov4-obj_final.weights -dont_show -ext_output < data/test.txt > result.txt
```

Outra modificação necessária na transição do treinamento para teste se refere ao arquivo "yolov4-obj.cfg", no qual, segundo recomendações de Carata et al. (2019) e Mahalleh, ALQutami e Mahmood (2019), os parâmetros "*batch*" e "subdivisões" precisam ser alterados para 1. Para a realização dos testes foram utilizadas as imagens no tamanho original (1.664x832 pixels), sem emprego de técnicas de pré-processamento, logo, o parâmetro "*size*" também foi modificado no referido arquivo.

[7] Um *threshold* de 0,50, por exemplo, indica uma a pecisão de detecção para aquele objeto é de 50%.

6 Resultados e Discussões

6.1 Banco de Imagens

O banco de imagens, denominado **ACR-Street View**, foi elaborado com base em 473 panorâmicas completas do GSV (360 graus na horizontal). Na Tabela 1 são sintetizas as informações referentes à estruturação do banco de imagens de acordo com os estratos populacionais dos municípios constantes no recorte geográfico utilizado.

Tabela 1 – Síntese do banco de imagens ACR-Street View

Estratos Populacionais (habitantes)	Busca no GSV	Panorâmicas Adquiridas	Acurácia de busca (panorâmicas)	Panorâmicas Candidatas	Acurácia de busca (rampas)
<20.000	5.483	447	8,15%	134	29,98%
≥20.000 <50.000	546	199	36,45%	81	40,07%
≥50.000 <100.000	515	232	45,05%	86	37,07%
≥100.000 <500.000	351	191	54,42%	101	52,88%
≥500.000	145	132	91,03%	71	53,79%
Total	7.040	1.201	17,06%	473	39,38%

Nota: Panorâmicas se referem à panorâmicas; Busca no GSV representa o número de buscas realizadas (entrada de par de coordenadas necessários para retornar aproximadamente o número mínimo de rampas rotuladas estabelecido); Panorâmicas Adquiridas indica o número de imagens resultantes das buscas realizadas; Acurácia de Busca (panorâmicas) corresponde a porcentagem de panorâmicas adquiridas em relação ao total de buscas realizadas; Panorâmicas Candidatas são as panorâmicas adquiridas as quais possuem rampas de acessibilidade; Acurácia de busca (rampas) corresponde a porcentagem de panorâmicas candidatas em relação às panorâmicas adquiridas.

Fonte: Elaborado pela autora

A partir desta tabela, pode-se verificar que, para obter aproximadamente o mesmo número de rampas rotuladas, a busca no GSV é maior nos municípios pequenos - reforçando ainda que esta busca foi realizada nos setores censitários com os maiores números de rampas de acessibilidade na calçada, segundo os dados do Censo (IBGE, 2010a). Isso evidencia que nos municípios de estrato populacional menor que 20 mil habitantes, no qual foram necessárias 5.483 buscas (vide coluna "Busca no GSV" na Tabela 1), ocorre menor abrangência de dados do GSV.

As porcentagens exibidas na coluna acurácia de busca (panorâmicas) reforçam este padrão, ou seja, aumentam a medida que aumentam os estratos populacionais. Percebe-se também que a acurácia de busca (rampas), que é o percentual de panorâmicas com rampas de acessibilidade em relação ao total de panorâmicas adquiridas, também aumenta de acordo com os estratos populacionais, porém, com menor variação.

Verificou-se também que a acurácia total de busca (panorâmicas) é 17,06% e a acurácia total de busca (rampas) é 39,38%. Conforme exposto na Tabela 1, apesar de se ter agrupado todos os municípios com população abaixo de 20 mil habitantes num único estrato, a acurácia de busca (panorâmicas) é consideravelmente menor quando comparado aos outros estratos, indicando que a fusão em estrato único foi uma opção viável.

6.2 Rotulação

A proposta de verificação da acurácia dos rótulos visou, de forma geral, avaliar a variação dos mesmos entre pessoas com perfis diferenciados. Para mensurar esta acurácia, foi atribuída a tarefa de rotulação a 10 delas, as quais demarcaram rampas em 10 imagens panorâmicas, com diversas rampas cada. Na Tabela 2 verifica-se os resultados deste experimento.

O nível de confiança médio dos rótulos foi de 0,9, indicando que a maioria dos usuários concordou, em quantidade, na identificação do que é uma rampa de acessibilidade em calçada. As principais dificuldades relatadas foram oclusões parciais ou totais das rampas por veículos e pedestres, falta de padrão das rampas (como por exemplo, pintura e piso tátil) e plano de fundo confuso (quando não há alteração visível de piso entre a calçada como um todo e a rampa).

Quando à delimitação das rampas, a média do desvio padrão dos centroides foi $\overline{\sigma}_{Cx} = 3{,}7$ e $\overline{\sigma}_{Cy} = 2{,}1$ pixels (em imagens de 1.664x832 pixels). Na Figura 29 são apresentados alguns exemplos de variação de caixas delimitadoras entre usuários.

A maior parte da variação entre delimitação de rampas ocorreu devido à discordância ao assumir qual é o limite de uma rampa de acessibilidade. Por exemplo, alguns consideraram apenas a inclinação interna na mesma, outros incluíram a transição entre rampa de calçada e entre rampa e leito carroçável. Visando padronizar os rótulos do banco de imagens, foi adotado como limite de rampa todo o conjunto que a compõe (compreendendo a transição entre rampa de calçada e entre rampa e leito carroçável).

Na Figura 29 verificam-se também exemplos da falta de padronização das rampas de acessibilidade em passeios públicos, enfatizando a importância de incluir uma grande variedade de municípios de diferentes estratos populacionais na coleta de dados no GSV, uma vez que isso refletirá a cobertura de diferentes padrões, cores (pintura), presença ou ausência de piso tátil, entre outros.

Uma vez conhecida a acurácia dos rótulos, todas as panorâmicas candidatas obtidas na etapa anterior foram rotuladas, resultando um total de 1.413 rótulos. Conforme especificado no primeiro item do Capítulo 5, priorizou-se a rotulação de no mínimo 1.250 rótulos

Tabela 2 – Acurácia dos rótulos demarcados por múltiplos usuários

ID Rampas	**É rampa**	**Não é rampa**	**Nível de Confiança**	σ_{Cy}	σ_{Cy}
1-A	10	0	1,0	2,5	1,8
1-B	10	0	1,0	3,0	2,2
2-A	10	0	1,0	1,2	1,0
2-B	10	0	1,0	1,0	1,4
2-C	10	0	1,0	3,0	1,4
3-A	9	1	0,9	2,9	2,0
3-B	10	0	1,0	3,6	1,5
3-C	10	0	1,0	2,1	1,1
4-A	10	0	1,0	2,6	2,8
4-B	10	0	1,0	7,3	3,3
5-A	10	0	1,0	2,3	1,0
5-B	1	9	0,1	N/A	N/A
5-C	10	0	1,0	2,7	1,3
5-D	9	1	0,9	4,0	3,0
6-A	10	0	1,0	1,8	1,2
6-B	10	0	1,0	2,9	2,1
7-A	10	0	1,0	4,5	3,2
7-B	10	0	1,0	7,6	6,6
7-C	3	7	0,3	10,2	0,8
8-A	10	0	1,0	2,4	1,2
8-B	10	0	1,0	1,2	1,7
9-A	9	1	0,9	0,6	1,2
9-B	10	0	1,0	3,4	2,5
9-C	10	0	1,0	2,7	1,1
9-D	10	0	1,0	2,3	1,5
10-A	10	0	1,0	4,0	4,4
10-B	10	0	1,0	12,8	2,5
10-C	3	7	0,3	3,8	2,6
10-D	8	2	0,8	2,7	1,4
10-E	9	1	0,9	5,5	4,6
Média	–	–	**0,9**	**3,7**	**2,1**

Em ID das rampas, o número identifica a imagem e a letra a rampa correspondente. A segunda coluna representa quantos dos usuários identificaram o objeto como rampa e a terceira quantos não identificaram aquela rampa.

Fonte: Elaborado pela autora

divididos equitativamente entre os estratos populacionais. Na Tabela 3 estão contabilizados o número de rampas delimitadas em cada estrato populacional.

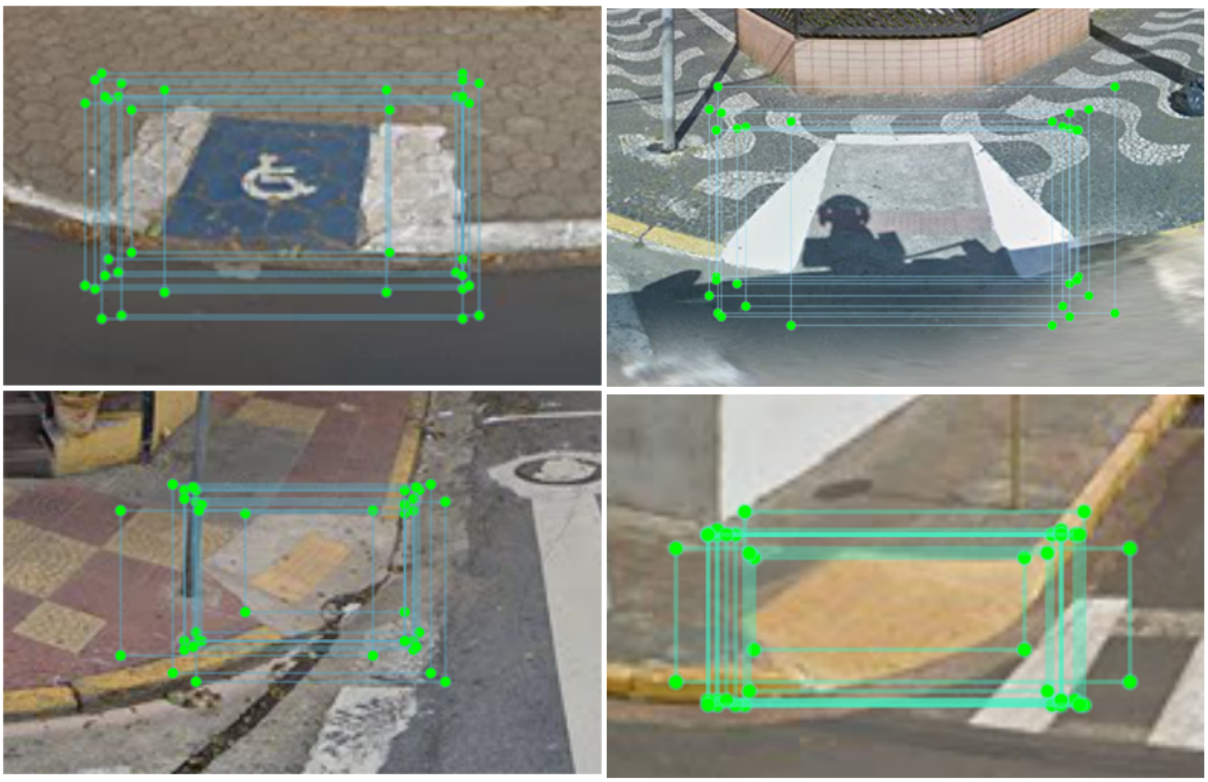


Figura 29 – Exemplos de variação entre caixas delimitadoras de diversos usuários
Fonte: Elaborado pela autora

Tabela 3 – Número de rampas rotuladas por estrato populacional

Estratos Populacionais (habitantes)	**Rampas Rotuladas**
<20.000	264
≥20.000 <50.000	300
≥50.000 <100.000	276
≥100.000 <500.000	280
≥500.000	293
Total	1.413

Fonte: Elaborado pela autora

6.3 Pré-processamento

O emprego da técnica de *Tiling* resultou na divisão das imagens que, originalmente possuíam 1.664x832 pixels, em porções de 416x416 pixels, de forma que cada imagem poderia gerar no máximo 8 porções. Considerando que um dos objetivos desta técnica é otimizar a memória computacional para processamento, mantiveram-se apenas as porções as quais possuíam anotações de rampas, eliminando-se portanto as porções sem rótulos (exemplificado na Figura 30).

Ressalta-se que os rótulos não têm seu tamanho alterado, apenas são proporcionalmente mais representativos em relação ao tamanho da imagem fracionada. Por exemplo, na

Figura 30 – Exemplo de *Tiling* em uma panorâmica.
Fonte: Elaborado pela autora.

Figura 30, a primeira rampa rotulada, a partir da esquerda, possui 27x10 pixels. Em relação a imagem original, isso equivale à 1,62% em x (*width*) e 1,20% em y (*height*). Já em relação à imagem *tiled* é mais representativo, 6,49% em x (*width*) e 2,40% em y (*height*).

Diferentemente desta técnica, o redimensionamento modifica o tamanho da imagem e dos rótulos simultaneamente, não resultando em modificação da porcentagem relativo do rótulo. A vantagem encontra-se na otimização da memória computacional, ignorando o fato do objeto ser relativamente grande ou pequeno. O redimensionamento e a *Data Augmentation* foram implementadas no Roboflow, sendo que na Figura 31 tem-se exemplos das técnicas aplicadas.

Figura 31 – Exemplo de *Data Augmentation* realizada no Roboflow: (a) original, (b) *flipping*, (c) *blur*, (d) *brightness* e (e) *shear*.
Fonte: Elaborado pela autora.

É importante enfatizar que no caso da *Data Augmentation* o número de rótulos foi dobrado, uma vez que optou-se pelo incremento de 100% no número de imagens, aplicando-se aleatoriamente os processos exemplificados na Figura 31. Isso não ocorre na técnica de *Tiling*, pois o número de imagens resultantes é maior, porém o número de rótulos permanece o mesmo.

Desta forma, nos treinamentos sem implementação de técnicas de pré-processamento, o banco de imagens foi composto por 473 imagens e 1.413 rampas rotuladas, conforme contabilizado na Tabela 3. No treinamento que utilizou *Tiling*, o banco de imagens foi composto por 1.095 imagens e 1.413 rampas rotuladas e no treinamento que utilizou *Data Augmentation* além de *Tiling*, o banco de imagens foi composto por 2.190 imagens e 2.826 rampas rotuladas.

6.4 Experimentação de treinamentos e validação

Ao todo foram treinadas 5 CNNs, sendo que o tempo necessário para a execução de 6.000 iterações cada variou, principalmente em função da utilização dos pesos convolucionais pré-treinados e tamanho das imagens. No Quadro 3 estão resumidas as configurações principais dos 5 treinamentos propostos para as CNNs e o tempo necessário para realizá-los.

Quadro 3 – Treinamentos propostos e suas configurações principais

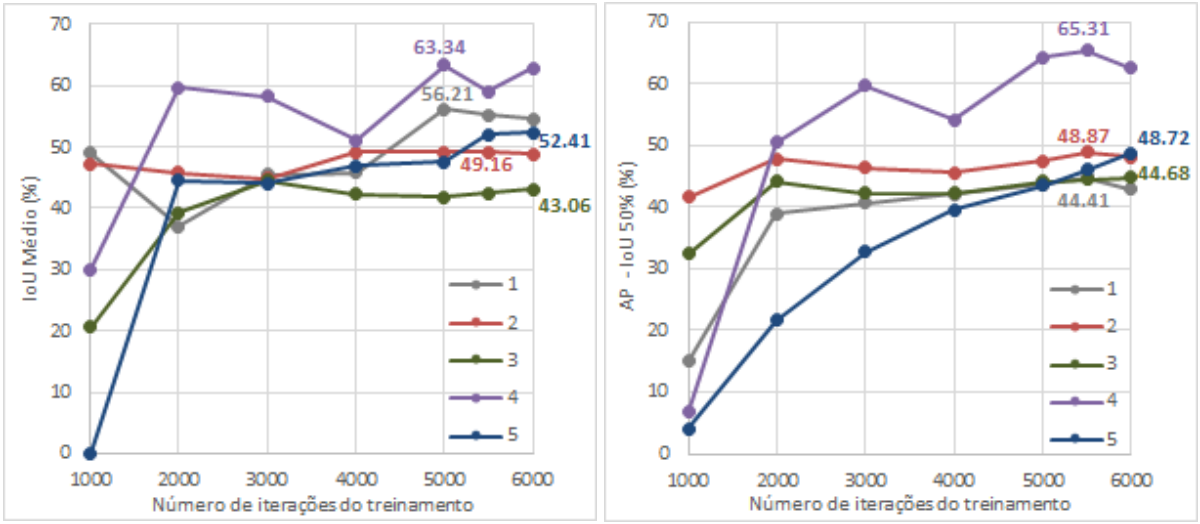
T*	Subdivisões	*Width x Height* (pixels)	Com Pesos Convolucionais	Com *Data Augmentation*	Com *Tiling*	Tempo (horas)
1	32	832x416	Não	Não	Não	21,0
2	32	832x416	Sim	Não	Não	12,0
3	16	1.664x832	Sim	Não	Não	15,0
4	32	416x416	Sim	Não	Sim	11,0
5	32	416x416	Sim	Sim	Sim	10,5

T* = Número referência do Treinamento.

Fonte: Elaborado pela autora

Na Figura 32 pode ser observada a evolução dos valores de IoU e AP registrados ao longo dos treinamentos, a cada 1.000 iterações. Na Tabela 4 são apresentados os valores de *precision*, *recall* e *F1-score* observados.

Segundo Mahalleh, ALQutami e Mahmood (2019) e Bochkovskiy, Wang e Liao (2020b), com base nestes gráficos, o momento ideal para interromper o treinamento seria na iteração onde o AP e o IoU atingiram os maiores valores. Por esta razão, em alguns casos, foram obtidas métricas entre as iterações 5.000 e 6.000, uma vez que estas se mostraram mais promissoras.



(a) IoU Médio. (b) AP (IoU 50%).

Figura 32 – Métricas de validação das CNNs treinadas
Fonte: Elaborado pela autora.

Na Figura 32a percebe-se que os treinamentos 1 (linha cinza) e 2 (linha vermelha) são os únicos que apresentam queda na porcentagem de IoU entre as iterações 1.000 e 2.000. Estes treinamentos se referem ao banco de imagens que foi redimensionado em 50% (832x416 pixels).

O treinamento 1, o qual não utilizou pesos convolucionais pré-treinados, apresentou evolução considerável até a iteração 5.000 e o treinamento 2, o qual utilizou pesos convolucionais pré-treinados comportou-se como um platô. Em contrapartida, na Figura 32b, o treinamento 2 resultou em uma porcentagem de AP superior ao treinamento 1 entre as iterações 5.000 e 6.000, indicando que a utilização de pesos convolucionais influencia positivamente na acurácia do modelo. Outro ponto relevante é que a incorporação de pesos convolucionais reduziu o tempo total do treinamento.

Ainda sobre estes dois treinamento, 1 e 2, verifica-se na Tabela 4, também entre as iterações 5.000 e 6.000, que os valores de *F1-score* são muito próximos, contudo, a *precision* é maior no treinamento 1 e o *recall* é maior no treinamento 2. Assim, pode-se inferir que o treinamento 1 é capaz de detectar com maior precisão enquanto que o treinamento 2 perde menos detecções.

Outra comparação interessante se dá entre os treinamentos 2 e 3, sendo que, respectivamente, um priorizou o *batch size* em 32 e sacrificou o tamanho das imagens ao redimensioná-las para 832x416 pixels e o outro priorizou o tamanho original das imagens (1.664x832 pixels) e reduziu o *batch size* para 16. Olhando para as Figuras 32a e 32b verifica-se que a performance do treinamento 2 é superior, mesmo que as imagens possuam tamanho reduzido.

Tabela 4 – Precision, recall e F1-score

Nº de Iterações	Treinamentos									
	1		2		3		4		5	
1.000	P	0,74	P	0,67	P	0,30	P	0,50	P	0,00
	R	0,07	R	0,34	R	0,39	R	0,00	R	0,00
	F1	0,12	F1	0,45	F1	0,34	F1	0,01	F1	0,00
2.000	P	0,53	P	0,66	P	0,56	P	0,85	P	0,69
	R	0,39	R	0,42	R	0,44	R	0,33	R	0,09
	F1	0,45	F1	0,51	F1	0,49	F1	0,48	F1	0,15
3.000	P	0,67	P	0,63	P	0,62	P	0,81	P	0,68
	R	0,38	R	0,44	R	0,42	R	0,47	R	0,09
	F1	0,49	F1	0,51	F1	0,50	F1	0,60	F1	0,15
4.000	P	0,66	P	0,67	P	0,60	P	0,75	P	0,71
	R	0,39	R	0,42	R	0,43	R	0,49	R	0,23
	F1	0,49	F1	0,52	F1	0,50	F1	0,59	F1	0,35
5.000	P	0,80	P	0,68	P	0,59	P	0,87	P	0,71
	R	0,38	R	0,45	R	0,46	R	0,51	R	0,31
	F1	0,52	F1	0,54	F1	0,52	F1	0,65	F1	0,43
>5.000 <6.000	P	0,78	P	0,69			P	0,83		
	R	0,38	R	0,45			R	0,53		
	F1	0,51	F1	0,54			F1	0,65		
6.000	P	0,77	P	0,68	P	0,60	P	0,87	P	0,78
	R	0,36	R	0,45	R	0,46	R	0,53	R	0,34
	F1	0,49	F1	0,54	F1	0,52	F1	0,66	F1	0,48

P = *precision*; R = *recall*; F1 = *F1-score*.

Destaque em cinza para os resultados da iteração de cada treinamento com o maior valor de AP.

Fonte: Elaborado pela autora

Este fato é reforçado pela diferença nos valores de *precision* entre os dois casos, 0,69 entre 5.000 e 6.000 iterações no treinamento 2 e 0,60 na iteração 6.000 no treinamento 3. Considerando que o *recall* praticamente não apresentou variação (0,45 e 0,46), isso é um indicativo de que o treinamento 2 é mais preciso ao classificar como positivo o que realmente é positivo. Infere-se, portanto, que o *batch size* em 32 mostrou-se mais promissor.

Cabe ainda comparar o treinamento 2 com o 4, no qual implementou-se a técnica de pré-processamento *Tiling*. Tanto na Figura 32a quanto na Figura 32b a performance do treinamento 4 foi superior (vide iteração 5.000 na Figura 32a e iteração entre 5.000 e 6.000 na Figura 32b). Especificamente quanto à AP (Figura 32b) o treinamento 4 se destaca entre os demais com cerca de 18,64% de melhoria na precisão das detecções. Neste caso, o aprimoramento significativo ocorreu também nas medidas de *F1-score*, *precision* e *recall*, indicando que este modelo foi o melhor modelo para acertar as predições corretamente e recuperar muitos exemplos da classe de interesse.

Por fim, numa última tentativa de aprimorar o modelo, o treinamento 5 incorporou a técnica de pré-processamento *data augmentation*. Nas Figuras 32a e 32b não verificou-se

aprimoramento do treinamento 5 em relação ao treinamento 4, apesar de que o modelo apresentou AP muito próximo do treinamento 2, o segundo mais promissor. Quando às outras métricas, quando comparado ao treinamento 4, o treinamento 5 mostrou queda significativa no *F1-score*, sendo que este foi mais afetado pela queda no valor do *recall* (de 0,53 para 0,34). Estes resultados demonstram que o processo de *data augmentation* não contribuiu para o aperfeiçoamento das detecções.

Os treinamentos 1, 2 e 3 comportaram-se predominantemente como um platô de AP a partir da iteração 2.000, indicando que com base no banco de imagens preparado, não há indícios de que um maior tempo no treinamento possa resultar em melhoria do modelo (MAHALLEH; ALQUTAMI; MAHMOOD, 2019; BOCHKOVSKIY; WANG; LIAO, 2020b). Por outro lado, os treinamentos 4 e 5, principalmente o 4, não parecem ter atingido este platô, indicando que a continuidade no treinamento possa ainda resultar melhoria na AP.

Por este motivo, optou-se por prosseguir com o treinamento dessas duas CNNs para verificar se há possibilidade de aperfeiçoamento. Os resultados constam na Figura 33.

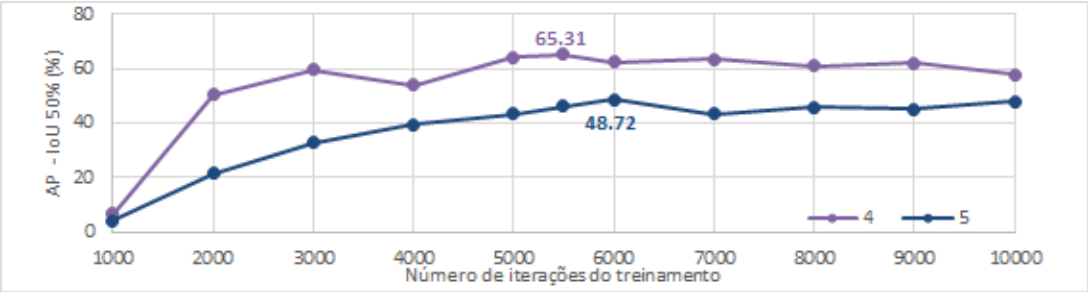


Figura 33 – AP – (IoU 50%) para os treinamentos 4 a 5.
Fonte: Elaborado pela autora.

Observando-se agora os resultados da continuidade dos treinamento, percebe-se que ambos atingiram um platô, indicando que, com base no banco de imagens disponíveis, não houve melhoria na performance. Os treinamentos 4 e 5 atingiram a maior AP dentro dos limites de iterações recomendados por Carata et al. (2019) e Mahalleh, ALQutami e Mahmood (2019), conforme já ilustrado na Figura 32b e confirmado na Figura 33.

Deste forma, a CNN que mostrou-se mais promissora para a etapa de testes foi a resultante do treinamento 4, a qual utilizou como técnica de pré-processamento apenas o *Tiling*, apropriou-se dos pesos convolucionais pré-treinados disponibilizados pelos desenvolvedores da Darknet Bochkovskiy, Wang e Liao (2020b) e *batch size* 64, com 32 subdivisões.

6.5 Teste

A etapa de teste se desenvolveu com base nos pesos convolucionais resultantes do treinamento 4, pois este se mostrou mais promissor na etapa de validação. Na Figura 34

são exemplificados os aspectos para os quais as imagens foram aferidas, contendo em 34a, um exemplo de Falso Positivo (FP) e Verdadeiro Positivo (VP), e em 34b um exemplo de Falso Negativo (FN).

(a) FP e VP, respectivamente (b) FN

Figura 34 – Exemplos de FP, VP e FN
Fonte: Elaborado pela autora

Percebe-se que em 34a o objeto detectado com precisão de 97% não é uma rampa, logo trata-se de um FP, e o objeto detectado com 60% de precisão é uma rampa, logo um VP. Já na Figura 34b nenhum objeto foi detectado, contudo, há uma rampa de acessibilidade no passeio público, identificando-se portanto um exemplo de FN.

Outro ponto relevante a ser investigado se refere aos padrões de rampas que a CNN foi capaz de detectar. É esta característica que confere à rede potencialidade de ser empregada em diversos municípios brasileiros, os quais possuem diversos designs de rampas de acessibilidade. Na Figura 35 verifica-se alguns dos padrões mais frequentes observados, que são as rampas com pintura em amarelo (Figura 35a), as rampas sem pintura com pavimento tátil amarelo (Figura 35b), as rampas com pintura em azul (Figura 35c) e as rampas com pintura em azul e piso tátil amarelo (Figura 35d).

Outras rampas frequentes são aquelas que não possuem diferenciação em relação ao piso do passeio público em que está localizada. Estas rampas são muito frequentes nos municípios de pequeno porte e estão ilustradas na Figura 36.

De forma geral a CNN resultante do treinamento 4 foi capaz de detectar as rampas de diversos designs, o que pode ser visto como uma vantagem ao considerar que a mesma pode ter aplicações em vários municípios e/ou em regiões diferentes de um mesmo município. Cabe mencionar também alguns casos específicos ilustrados na Figura 37.

Na Figura 37a tem-se o exemplo de duas rampas em estado inadequado de conservação, sendo que uma delas foi detectada com precisão de 90% (VP) e a outra não foi detectada (FN). Na Figura 37b observa-se um exemplo de detecção bem sucedida (VP) na qual a rampa estar parcialmente obstruída por um veículo. Por outro lado, na Figura 37c

(a) Rampa amarela

(b) Rampa com piso tátil

(c) Rampa azul

(d) Rampa azul com piso tátil

Figura 35 – Exemplos de padrões de rampas frequentes
Fonte: Elaborado pela autora

(a) Contínuo

(b) Paralelepípedo

Figura 36 – Exemplos de rampas sem diferenciação no piso
Fonte: Elaborado pela autora

tem-se um exemplo em que não houve detecção (FN) em caso de obstrução parcial por veículo. Por fim, na Figura 37d são ilustradas duas detecções equivocadas (FP), pois se referem à rampas de acesso à veículos e não à rampas exclusivas para usuários de cadeiras de rodas. Nesta mesma figura tem-se um caso em que houve sobreposição de detecções, sendo uma com precisão de 69% e a outra com precisão de 46%.

Nesta etapa final, todas as imagens do conjunto de teste foram avaliadas no sentido de identificar VPs, FPs e FNs. Além disso, foram computadas as precisões dos VPs e FPs e as dimensões (em pixels quadrados) das detecções. Na Tabela 5 são reunidas as informações estatísticas dos testes.

Tabela 5 – Síntese dos resultados dos testes (geral)

	Total de rampas		**Precisão**					
		(%)	**Min**	**Q1/4**	**Md**	**Me**	**Q3/4**	**Max**
VP	204	77,57	0,26	0,77	0,94	0,85	0,98	1,00
FP	21	7,98	0,26	0,31	0,43	0,50	0,65	0,97
FN	38	14,45	N/A	N/A	N/A	N/A	N/A	N/A
Total	263	100	162	688	1.125	1.293	1.634	5.029
Área*	–	–	152	479	925	1.323	1.716	14.688

*Apenas dos VPs (pixels quadrados).

Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Destaque em cinza para os valores que embasaram a definição do *threshold*.

Fonte: Elaborado pela autora

(a) Rampas em mal estado de conservação

(b) Obstrução parcial com detecção

(c) Obstrução parcial sem detecção

(d) Rampa para acesso de veículos e detecção sobreposta

Figura 37 – Exemplos de detecção em casos específicos
Fonte: Elaborado pela autora

A partir das informações constantes nesta tabela observa-se que a precisão média de detecções corretas (VP) é 85% e a precisão média de detecções incorretas (FP) é de 50%. Verificou-se ainda que a precisão mínima com que uma rampa foi detectada corretamente foi de 26% e a precisão mínima com que uma rampa foi detectada incorretamente também foi de 26%. Estes números podem ser indicativos de um possível *threshold* para priorizar detecções corretas e ignorar as incorretas. Caso opte-se por um *threshold* de 26%, por

exemplo, todas as 204 observações VPs seriam consideradas. Por outro lado, todas as 21 observações FPs também seriam.

Neste caso, a análise dos quartis e medianas podem levar à escolha de um *threshold* que harmonize o total de detecções corretas e incorretas. De acordo com a Tabela 5 (vide valores destacados em cinza), a mediana de precisão dos VPs foi de 94% e de FPs foi de 43%, indicando que metade das detecções corretas se deram com precisões acima de 94% e metade das detecções incorretas se deram com precisões a partir de 43%. Prosseguindo com análises semelhantes, verificou-se que 75% das detecções corretas se deram com precisões a partir de 77% (valores superiores ao do primeiro quartil) e que 75% das detecções incorretas se deram com precisões de até 65% (valores inferiores ao terceiro quartil). Consequentemente, um *threshold* entre 65 e 77% maximalizaria a quantidade de VPs e minimizaria a quantidade de FPs.

Tratando-se de FNs, observou-se que o detector final deixou de demarcar 38 rampas, 14% do total de rampas constantes no conjunto de teste. Conforme demonstrado nas Figuras 37a e 37c, as principais razões pelas quais as rampas deixaram de ser identificadas foram obstrução parcial e mau estado de conservação.

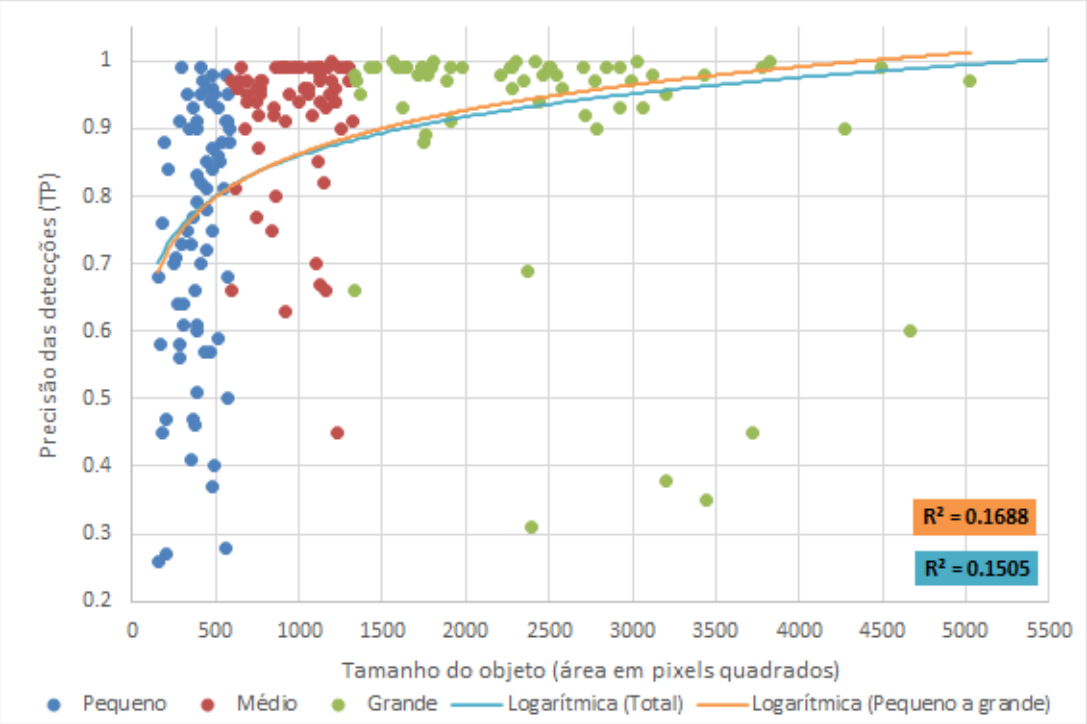
Na Tabela 5 são apresentadas ainda informações quanto ao tamanho dos objetos detectados (ressaltando-se que foram computadas nesta tabela apenas as áreas dos VPs), sendo o tamanho médio dos objetos detectados 1.323 pixels quadrados. Dentre as detecções realizadas, o primeiro quartil são de rampas que ocupam menos de 479 pixels quadrados (até 0,03% do total da imagem), o segundo quartil (mediana) eram de rampas entre 479 e 925 pixels quadrados (até 0,06% do total da imagem), o terceiro quartil de rampas entre 925 e 1.716 pixels quadrados (até 0,12%) e o último quartil eram de rampas até 14.688 pixels quadrados (até 1,06% da imagem).

Em relação ao tamanho das imagens (1.664x832 pixels), estes objetos podem ser considerados pequenos, já que as imagens do GSV são consideravelmente grandes. Para as próximas análises, os objetos detectados serão subdivididos em grupos de tamanhos, tomando-se como base para esta subdivisão a média (1.323 pixels quadrados): pequenos (até 585 pixels quadrados), médios (entre 585 e 1.323 pixels quadrados), grandes (entre 1.323 e 6.882 pixels quadrados) e muito grandes (acima de 6.882 pixels quadrados).

Esta subdivisão foi proposta buscando avaliar a eficiência da rede quanto à detecção de rampas próximas e distantes do ponto de captura da panorâmica – assumindo-se que a classe rampa é representada por uma quantidade de pixels, intuitivamente, os objetos próximos são maiores e os mais distantes são menores.

Portanto, as precisões das detecções e seus respectivos tamanhos foram então plotados no gráfico da Figura 38. Como os objetos de tamanho muito grande foram apenas 2 – com precisões de 0,72 e 0,92, para objetos de 8.385 e 14.688 pixels quadrados

respectivamente – estas duas observações não foram incluídas na referida figura (para viabilizar a visualização). Além destes, foram detectados 74 objetos pequenos, 67 médios e 61 grandes.



Nota: a linha de tendência Linear (Pequeno a Grande) se refere apenas aos dados exibidos no gráfico, já a linha de tendência Linear (total) inclui duas observações muito grandes, as quais não estão ilustradas no gráfico.

Figura 38 – Precisão das detecções para diferentes tamanhos de objetos detectados.
Fonte: Elaborado pela autora.

De forma geral, apesar da ocorrência de algumas detecções com baixa precisão, esta característica se repete para todos os tamanhos de objeto. Uma análise visual preliminar indica que houve maior ocorrência de detecções com baixa precisão entre os objetos pequenos - o que pode ser explicado pela maior dificuldade em identificá-los ou devido ao maior número de observações nesta categoria. Por outro lado, uma análise das curvas de tendência logarítmica - modelo de melhor ajuste aos dados - e dos valores de R^2 evidenciam que não há uma forte correlação entre precisão e tamanho do objeto (valores de R^2 estão mais próximos de 0 do que de 1). Em outras palavras, a rede treinada é capaz de detectar objetos de tamanhos variados (próximos e distantes).

Também é possível constatar a partir da Figura 38, que dentre os objetos de tamanho médio e grande, a ocorrência de precisões acima de 65% é mais significativa (94%) e acima de 77% também é representativa (88%). Dentre os objetos pequenos, cai para 68% o quantitativo de objetos detectados com precisão maior do que 65% e para 50% o quantitativo com precisão maior do que 77%.

Considerando estas análises, verifica-se que o detector de rampas teve melhor performance para objetos a partir de 585 pixels quadrados em uma imagem de 1.381.120 pixels quadrados, o que representa 0,04% da superfície da imagem.

Uma outra linha de investigação refere-se aos estratos populacionais, uma vez que foram adquiridas 10 imagens de cada estrato populacional, logo 50 imagens. Os resultados dos testes estão detalhados nas Tabelas 6 a 10, sendo que cada tabela se refere à um estrato populacional e contém os dados de detecção para cada uma das 263 rampas constantes nas imagens, além de um resumo estatístico das observações.

Tabela 6 – Resultados dos testes para estrato populacional até 20.000 habitantes

Pano ID 15717				Pano ID 15719				Pano ID 15721				Pano ID 15722				Pano ID 15724			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,88			585			Sim	N/A	0,84			210	0,70			406	0,82			407
0,90			676			Sim	N/A	0,95			495	0,90			336			Sim	N/A
0,96			2.275	0,85			1.116	0,99			901	0,91			390	0,99			1.800
1,00			1.190	0,61			384	0,77			741	0,96			765	0,88			1.752
				0,99			1.586	0,96			741	0,99			2.494				
				0,87			760												
					0,27		240												

Pano ID 58946				Pano ID 58948				Pano ID 62371				Pano ID 358757				Pano ID 583886			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,93			516	0,95			408	0,83			392	0,91			1.320	0,81			546
0,94			689			Sim	N/A	0,95			325		0,26		108			Sim	N/A
0,98			2.465	0,45			180	0,59			516		0,36		4.949	0,27			207
1,00			1.804			Sim	N/A			Sim	N/A					0,26			152
				0,98			2.548											Sim	N/A
				0,73			290									0,73			352
																0,35			3.444
																		Sim	N/A

Resumo estatístico

	Total de rampas	Precisão					
		Min	Q1/4	Md	Me	Q3/4	Max
VP	38	0,26	0,78	0,90	0,82	0,96	1,00
FP	3	0,26	0,26	0,27	0,30	0,31	0,36
FN	9	N/A	N/A	N/A	N/A	N/A	N/A
Área*		152	390	630	952	1.287	3.444

*Apenas dos VPs (pixels quadrados). Pano ID=identificação da rampa, Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Fonte: Elaborado pela autora

Tabela 7 – Resultados dos testes para estrato populacional entre 20.000 e 50.000 habitantes

Pano ID 383472				Pano ID 76556				Pano ID 116133				Pano ID 1477599				Pano ID 2147456			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
		Sim	N/A	0,96			624	0,97			2.993	0,66			1.334	0,66			1.161
0,99			1.064			Sim	N/A		0,27		1.180			Sim	N/A		0,87		330
0,95			765	0,87			476	0,58			288			Sim	N/A	0,28			561
		Sim	N/A	0,47			360	0,64			308	0,90			392	0,98			2.204
		Sim	N/A	0,99			2.277		0,68		351	0,84			480	0,38			3196
	0,97		1.925	0,97			2.346	0,92			14.688	0,90			2.784				
0,60			4.662											Sim	N/A				
0,71			264																
	0,43		1.035																
		Sim	N/A																

Pano ID 632256				Pano ID 116132				Pano ID 1003287				Pano ID 118675				Pano ID 131776			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,94			464	0,85			525	0,9			4.272	0,99			989	0,56			286
0,94			1.000	0,96			2.573					0,31			2.394		0,89		1.505
				0,99			4.488												

Resumo estatístico

	Total de rampas	Precisão					
		Min	Q1/4	Md	Me	Q3/4	Max
VP	30	0,28	0,64	0,90	0,79	0,96	0,99
FP	6	0,27	0,49	0,77	0,68	0,88	0,97
FN	8	N/A	N/A	N/A	N/A	N/A	N/A
Área*		264	477	1.032	2.007	2.528	14.688

*Apenas dos VPs (pixels quadrados). Pano ID=identificação da rampa, Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Fonte: Elaborado pela autora

Tabela 8 – Resultados dos testes para estrato populacional entre 50.000 e 100.000 habitantes

Pano ID 5091				Pano ID 345201				Pano ID 46413				Pano ID 345236				Pano ID 159187			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,51			390	0,99			2.262	0,72			448	0,93			1.160	0,98			476
		Sim	N/A	0,98			1.770	0,91			286	0,63			918			Sim	N/A
	0,27		153	0,94			1.122	0,99			300	0,93			2.920	0,47			207
		Sim	N/A	0,95			946	0,61			308	0,99			3.784			Sim	N/A
0,88			189	0,99			968	0,98			3.124	0,98			1.710			Sim	N/A
		Sim	N/A	0,99			900	0,75			840	0,99			920	0,76			180
		Sim	N/A	0,97			1.350					0,91			576	0,97			765
0,96			480			Sim	N/A									0,91			560
0,99			893													0,88			532
0,97			420																

Pano ID 46409				Pano ID 345228				Pano ID 345234				Pano ID 128225				Pano ID 351917			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,70			247	0,99			1.296	0,92			2.720			Sim	N/A	0,99			406
		Sim	N/A	0,96			1.222	1,00			2.304	0,90			578	0,75			480
0,94			2.436	0,91			1.914					0,99			2.840	1,00			3.822
1,00			2.412	0,97			2.769					0,99			1.624	0,72			8.385
		Sim	N/A	0,98			3.432					0,81			441		0,42**		3.182
0,58			168	0,99			2.508					0,82			1.152				

Resumo estatístico

	Total de rampas	Precisão					
		Min	Q1/4	Md	Me	Q3/4	Max
VP	52	0,47	0,86	0,95	0,89	0,99	1,00
FP	2	0,27	0,31	0,34	0,34	0,38	0,42
FN	11	N/A	N/A	N/A	N/A	N/A	N/A
Área*		168	469	933	1.440	2.272	8.385

*Apenas dos VPs (pixels quadrados).**Detecção sobreposta. Pano ID=identificação da rampa, Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Fonte: Elaborado pela autora

Tabela 9 – Resultados dos testes para estrato populacional entre 100.000 e 500.000 habitantes

Pano ID 1815621				Pano ID 18286				Pano ID 18303				Pano ID 18305				Pano ID 18307			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,64			275	0,97			598	0,92			752	0,94			1.222	0,92			850
0,95			1.188	0,96			448	0,97			777		0,65		336			Sim	N/A
0,95			576	0,57			432		0,42		231		0,69		5.461	0,37			484
0,66			598	0,45			1.225		0,65		200		0,46		1.890	0,77			364
	0,46		364	0,99			1.457			Sim	N/A	0,95			1.058	0,98			560
0,97			1.298	0,57			465									0,95			1.368
0,93			846														0,53		656
																0,93			3.060
Pano ID 18315				Pano ID 191707				Pano ID 191708				Pano ID 339752				Pano ID 18280			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,75			325	0,79			390	0,40			496	0,97			1.881	0,81			615
	0,36		187	0,6			384	0,96			1.060	0,99			1.127	0,91			920
		Sim	N/A			Sim	N/A												
0,99			2.706	0,99			2.926												
					0,31		198												

Resumo estatístico

	Total de rampas	Precisão					
		Min	Q1/4	Md	Me	Q3/4	Max
VP	33	0,37	0,75	0,93	0,83	0,97	0,99
FP	9	0,31	0,42	0,46	0,50	0,65	0,69
FN	4	N/A	N/A	N/A	N/A	N/A	N/A
Área*		275	484	777	991	1.222	3.060

*Apenas dos VPs (pixels quadrados). Pano ID=identificação da rampa, Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Fonte: Elaborado pela autora

Tabela 10 – Resultados dos testes para estrato populacional maior que 500.000 habitantes

Pano ID 1009804				Pano ID 1011465				Pano ID 1011519				Pano ID 1085717				Pano ID 191376			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,99			1.166	0,99			1.740	0,99			864	0,97			1.125	0,68			162
0,96			1.035	0,90			1.248	0,68			574	0,93			1.620	0,46			374
0,99			1.786	0,95			690	0,99			1.242	0,97			1.196			Sim	N/A
1,00			3.034	0,99			658	0,99			1.122	0,99			1.980	0,86			513
1,00			1.566	0,41			350	0,99			1.100	0,99			1.647	0,50			574
0,98			1.334	0,69			2.368	0,94			1.218	0,92			1.080	0,97			686
0,99			1.647					0,99			931	0,96			1.035	0,94			742
		Sim	N/A					0,99			1.008	0,98			1.125	0,66			380
																		Sim	N/A
																0,93			360
																		Sim	N/A
Pano ID 10754				Pano ID 23337				Pano ID 53812				Pano ID 72615				Pano ID 1009798			
VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área	VP	FP	é FN?	Área
0,89			1.764	0,78			444	0,99			1.914			Sim	N/A	0,99			1.430
0,67			1.125	0,45			3.720	0,97			5.029	0,99			1.736	0,99			1.276
	0,26		1.140	0,95			3.201					0,70			1.104			Sim	N/A
												0,97			645	0,80			855
												0,85			442				

Resumo estatístico

	Total de rampas	Precisão					
		Min	Q1/4	Md	Me	Q3/4	Max
VP	51	0,41	0,85	0,97	0,88	0,99	1,00
FP	1	0,26	0,26	0,26	0,26	0,26	0,26
FN	6	N/A	N/A	N/A	N/A	N/A	N/A
Área*		162	688	1.125	1.293	1.634	5.029

*Apenas dos VPs (pixels quadrados). Pano ID=identificação da rampa, Min=mínima, Q1/4=primeiro quartil, Md=mediana, Me=media, Q3/4=terceiro quartil, Max=máxima.

Fonte: Elaborado pela autora

Uma comparação quantitativa entre estas tabelas indicou que uma maior proporção de VPs, os quais representam as detecções corretas, foi identificada nos municípios com mais de 500.000 habitantes (88% do total de rampas deste estrato). Atribuindo maior credibilidade à esta característica, cabe destacar também que neste estrato populacional a média da precisão das detecções corretas é de 88% (a segunda maior), destacando-se também a precisão de 85% no primeiro quartil (também a segunda maior) – indicando que 75% das detecções se deram com precisões acima deste valor. Como consequência, este estrato populacional apresentou proporcionalmente menor ocorrência de detecções incorretas (FPs e FNs), destacando-se os FPs com apenas 2% do total de rampas deste estrato.

Analisando por esta mesma ótica, o estrato populacional de pior desempenho da CNN foi entre 20.000 e 50.000 habitantes, com apenas 68% do total de rampas detectadas corretamente. A precisão média das detecções VPs neste estrato foi a menor se comparada aos outros (79%) e a precisão do primeiro quartil também foi a menor (64%). Ao mesmo tempo, a precisão média com que os FPs foram detectados foi a maior dentre todos os estratos populacionais, 68%, simbolizando que, nos padrões de rampa inseridos neste estrato, o detector indica com uma precisão alta que um objeto é uma rampa, mesmo não sendo.

Conforme proposto nos objetivos deste trabalho, a base de imagens rotuladas contemplando panorâmicas de diversos municípios brasileiros viabilizou o treinamento de uma CNN capaz de identificar rampas de acessibilidade nos passeios públicos em imagens do GSV. Os diversos treinamentos realizados fomentaram discussões relevantes sobre as técnicas de pré-processamento mais adequadas para este banco de imagens, assim como os resultados apresentados nas etapas de validação e teste permitiram conhecer o desempenho do detector.

Almejando tornar a base de imagens pública para que outros usuários possam utilizá-la, as panorâmicas rotuladas foram indexadas na plataforma Github, no *weblink*:

<https://github.com/tatianeolivatto/ACR-Street-View.git>

e também na plataforma Zenodo, sob DOI:

10.5281/zenodo.5256106 <https://zenodo.org/record/5256106>

Juntamente com as panorâmicas rotuladas, foram anexados os pesos convolucionais resultantes dos treinamentos 3 e 4, por se tratarem das redes que utilizaram, respectivamente, o banco de imagens original e o banco de imagens pré-processadas com (*Tiling*). Foram também incluídas as instruções para download das panorâmicas do GSV na intenção

de facilitar o processo para aqueles que desejem implementar o detector de rampas ou até mesmo enriquecer o banco de imagens.

7 Considerações Finais

Atualmente é inevitável pensar novas soluções para as cidades sem o emprego de novas tecnologias, o que não é diferente no âmbito da mobilidade urbana. A reduzida acessibilidade urbana ainda é uma questão que acomete muitos municípios brasileiros e que, por questões éticas e legais, precisa de uma abordagem adequada e urgente.

Aliado a isso, observa-se por parte das autoridades e agentes de planejamento o desconhecimento da real situação das infraestruturas urbanas de acessibilidade, o que inclui as rampas para usuários de cadeiras de rodas em passeios públicos. Frente a este cenário desanimador, o emprego de inteligência artificial, especificamente aprendizagem de máquina e visão computacional, confirmou-se como uma alternativa viável para subsidiar soluções de acessibilidade urbana.

Dada a complexidade da tarefa de detecção de rampas de acessibilidade em passeios públicos, houve a necessidade de se compreender os conceitos envolvidos neste processo, iniciando-se pelo Aprendizado de Máquina e Aprendizagem Profunda, passando por Redes Neurais e Visão Computacional, até encontrar a YOLOv4 como rede neural com características que atendessem à demanda inicial.

O método proposto para a composição do banco de panorâmicas do GSV atendeu ao objetivo de inclusão de variados padrões de rampas existentes nos municípios brasileiros, ainda sim, sem tornar inviável sua estruturação. A partir daí, a tarefa que demandou mais esforço foi a rotulação manual das rampas de acessibilidade, um procedimento custoso em termos de tempo, além de estar sujeito a erros humanos. Por outro lado, uma vez concluído, torna a identificação de rampas em calçadas uma tarefa automatizada. Além disso, os testes acerca dos erros humanos comprovaram que a variação de rotulação entre usuários observada não alterará as detecções da CNN.

Uma das etapas que mais influenciaram na precisão final do detector foi a de pré-processamento, sendo a técnica de *Tiling* a que mais gerou impacto positivo. O uso dos pesos pré-treinados e as configurações de *batch size* 64 e subdivisões 32 se mostraram mais apropriadas.

Os testes indicaram que, a fim de harmonizar detecções corretas e incorretas, um *threshold* conveniente deve variar entre 65% e 77% no caso dos pesos convolucionais em questão. Levando em consideração o tamanho proporcional dos objetos em relação à dimensão das imagens, um *threshold* que se aproxime de 77% é mais apto à detecção de objetos médios e grandes (rampas mais próximas), enquanto que ao se aproximar de 65% potencializa a detecção de objetos pequenos (rampas mais distantes).

Os testes também comprovaram que a rede é eficaz na detecção de padrões diversificados de rampas, porém, algumas detecções incoerentes indicam limitações: algumas rampas em péssimo estado de conservação não foram detectadas, bem como alguns casos de oclusão parcial, e algumas rampas de acesso para veículos foram detectadas como rampas de acessibilidade. Constatou-se também que a melhor performance da CNN se deu dentre os padrões de rampas constantes nos municípios de estrato populacional acima de 500.000 habitantes.

Os resultados da etapa de teste, somados às porcentagens de IoU e AP da etapa de validação, confirmam que o detector resultante do treinamento 4, impulsionado principalmente pela técnica de *Tiling*, é promissor para a identificação de rampas de acessibilidade em imagens do GSV.

Consequentemente, apesar das dificuldades atravessadas no desenvolvimento do trabalho – como limitação de memória e tempo disponível para processamento no Google Colab – considera-se que o principal objetivo proposto foi alcançado, que foi construir um detector capaz de identificar rampas de acessibilidade em passeios públicos em imagens do GSV. Conclui-se também que o objetivo de disponibilizar publicamente um banco de panorâmicas do GSV com rótulos de rampas de acessibilidade de municípios brasileiros também foi atingido.

7.1 Trabalhos Futuros

Para trabalhos futuros sugere-se a ampliação do banco de imagens rotuladas para incluir outras classes que contemple, por exemplo, o estado de conservação das rampas ou casos de rampas obstruídas. Com base nos resultados apresentados, pode-se extrapolar a aplicação do método proposto para outros focos, como por exemplo, identificação de sinalização viária e hidrantes.

Outra linha de pesquisa seria apropriar-se da propriedade de *geotagging* das panorâmicas do GSV e efetivar o mapeamento das rampas para usuários de cadeiras de rodas a partir da obtenção das coordenadas geográficas das mesmas. Desta forma, podem ser exploradas aplicabilidades que vão desde a gestão pública desta e outras infraestruturas ao desenvolvimento de aplicativos de roteirização personalizados que abranjam este público alvo.

Referências

ABNT. *NBR 9050*: Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos. Rio de Janeiro: Associação Brasileira de Normas Técnicas, 2001. 97 p. Citado 2 vezes nas páginas 16 e 17.

AHMETOVIC, D. et al. Zebrarecognizer: Efficient and precise localization of pedestrian crossings. In: *Proceedings - International Conference on Pattern Recognition*. Stockholm, Sweden: IEEE, 2014. p. 2566–2571. Citado 2 vezes nas páginas 51 e 54.

AHMETOVIC, D. et al. Mind your crossings: Mining GIS imagery for crosswalk localization. *ACM transactions on accessible computing*, v. 9, n. 4, p. 11, apr 2017. ISSN 1936-7228. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28757907https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5531764/>. Citado 2 vezes nas páginas 51 e 54.

ALEXANDROVA, S.; TATLOCK, Z.; CAKMAK, M. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In: . [S.l.: s.n.], 2015. v. 2015, p. 5537–5544. Citado na página 67.

ALHAFNI, B. et al. *Mapping Areas using Computer Vision Algorithms and Drones*. 2019. 7 p. Disponível em: <https://arxiv.org/pdf/1901.00211.pdf>. Acesso em: 26/05/2020. Citado na página 21.

AMORIN, J. G. A. *Visão Computacional: Métricas - Mean Average Precision*. s/d. Disponível em: <http://www.lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/visao-computacionalmetricasmean-average-precision/>. Acesso em: 09/04/2021. Citado na página 72.

ANDREOPOULOS, A.; TSOTSOS, J. K. 50 Years of object recognition: Directions forward. *Computer Vision and Image Understanding*, v. 117, n. 8, p. 827–891, 2013. ISSN 1077-3142. Disponível em: <http://www.sciencedirect.com/science/article/pii/S107731421300091X>. Citado 2 vezes nas páginas 41 e 42.

APPLE. *Performing Convolution Operations*. vImage Programming Guide. 2016. Disponível em: <https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>. Acesso em: 23/04/2020. Citado na página 39.

ARAI, K.; KAPOOR, S. *Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1*. Springer International Publishing, 2019. (Advances in Intelligent Systems and Computing). ISBN 9783030177959. Disponível em: <https://books.google.com.br/books?id=T3WUDwAAQBAJ>. Citado na página 51.

BALLARD, D. H.; HINTON, G. E.; SEJNOWSKI, T. J. Parallel visual computation. *Nature*, v. 306, n. 5938, p. 21–26, 1983. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/306021a0>. Citado na página 40.

BENGIO, Y. Learning Deep Architectures for AI. *Foundations*, v. 2, p. 1–55, 2009. Citado 2 vezes nas páginas 37 e 38.

BISONG, E. Google colaboratory. In: ____. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. New York, USA: Apress, 2019. p. 59–64. ISBN 978-1-4842-4470-8. Citado na página 50.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. Citado 4 vezes nas páginas 48, 49, 56 e 63.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. *Yolo v4, v3 and v2 for Windows and Linux*. 2020. Disponível em: <https://github.com/AlexeyAB/darknet>. Acesso em: 21/02/2020. Citado 6 vezes nas páginas 68, 69, 70, 71, 79 e 82.

BRAGA, A. d. P.; CARVALHO, A. P. d. L. F.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e Aplicações*. Rio de Janeiro: LTC Editora, 2000. 262 p. ISBN 9788521615644. Disponível em: <https://books.google.com.br/books?id=R-p1GwAACAAJ>. Citado 3 vezes nas páginas 34, 35 e 36.

BRASIL. Lei nº 6.766, de 19 de dezembro de 1979. *Lex*: coletânea de legislação: edição federal, Brasília, DF, 1979. Disponível em: <http://www.planalto.gov.br/ccivil_03/LEIS/L6766.htm>. Acesso em: 15/01/2021. Citado na página 31.

BRASIL. Constituição (1988). *Constituição da República Federativa do Brasil*. Brasília, DF: Senado, 1988. Citado 2 vezes nas páginas 16 e 32.

BRASIL. Lei nº 10.257, de 10 de julho de 2001. *Lex*: coletânea de legislação: edição federal, Brasília, DF, 2001. Disponível em: <http://www.planalto.gov.br/ccivil_03/LEIS/LEIS_2001/L10257.htm>. Acesso em: 15/01/2021. Citado 4 vezes nas páginas 17, 30, 32 e 64.

BRASIL. Decreto nº 5.296, de 2 de dezembro de 2004. *Lex*: coletânea de legislação: edição federal, Brasília, DF, 2004. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 15/05/2020. Citado na página 17.

BRASIL. Lei nº 12.587, de 03 de janeiro de 2012. *Lex*: coletânea de legislação: edição federal, Brasília, DF, 2012. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12587.htm>. Acesso em: 15/01/2021. Citado 4 vezes nas páginas 18, 30, 32 e 64.

BRESSON, G. et al. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, v. 2, n. 3, p. 194–220, 2017. Citado na página 52.

CALVO, M. C. M. et al. Estratificação de municípios brasileiros para avaliação de desempenho em saúde. *Epidemiologia e Serviços de Saúde*, v. 25, n. 4, p. 767–776, 2016. ISSN 2237-9622. Citado na página 25.

CAPINERI, C. et al. *European Handbook of Crowdsourced Geographic Information*. London: Ubiquity Press, 2016. 464 p. Citado na página 20.

CARATA, S. et al. Complete Visualisation, Network Modeling and Training, Web Based Tool, for the Yolo Deep Neural Network Model in the Darknet Framework. In: *International Conference on Intelligent Computer Communication and Processing*. Cluj-Napoca, Romania: IEEE, 2019. p. 517–523. Citado 3 vezes nas páginas 69, 73 e 82.

CARDONE, G. et al. The participact mobile crowd sensing living lab: The testbed for smart cities. *IEEE Communications Magazine*, v. 52, n. 10, p. 78–85, 2014. ISSN 0163-6804 VO - 52. Citado na página 19.

CARNEIRO, T. et al. Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, v. 6, p. 61677–61685, 2018. ISSN 2169-3536. Citado na página 50.

CHEN, L. et al. Estimating pedestrian volume using Street View images: A large-scale validation test. *Computers, Environment and Urban Systems*, v. 81, p. 101481, 2020. ISSN 0198-9715. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0198971519304351>. Citado 2 vezes nas páginas 21 e 54.

CHEN, W.; ZHONG, X.; ZHANG, J. Optimization research and defect object detection of aeroengine blade boss based on YOLOv4. *Journal of Physics: Conference Series*, IOP Publishing, v. 1746, p. 012076, jan 2021. Disponível em: <https://doi.org/10.1088/1742-6596/1746/1/012076>. Citado na página 49.

CORRÊA, J. et al. Linear combination of forecasts with numerical adjustment via MINIMAX non-linear programming. *Revista Gestão da Produção Operações e Sistemas*, v. 11, p. 79–96, 2016. Citado na página 36.

DATA SCIENCE ACADEMY. As 10 principais arquiteturas de redes neurais. In: ____. *Deep Learning Book*. Brasília, DF: Data Science Academy, 2019. cap. 10. Disponível em: <http://deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/>. Acesso em: 22/04/2020. Citado 2 vezes nas páginas 38 e 47.

DATA SCIENCE ACADEMY. Definindo o tamanho do mini-batch. In: ____. *Deep Learning Book*. Brasília, DF: Data Science Academy, 2021. cap. 29. Disponível em: <https://www.deeplearningbook.com.br/definindo-o-tamanho-do-mini-batch/>. Acesso em: 01/05/2021. Citado na página 70.

DATA SCIENCE ACADEMY. O efeito do batch size no treinamento de redes neurais artificiais. In: ____. *Deep Learning Book*. Brasília, DF: Data Science Academy, 2021. cap. 37. Disponível em: <https://www.deeplearningbook.com.br/o-efeito-do-batch-size-no-treinamento-de-redes-neurais-artificiais/>. Acesso em: 01/05/2021. Citado na página 70.

DE JESUS, K. et al. Predicting centre of mass horizontal speed in low to severe swimming intensities with linear and non-linear models. *Journal of Sports Sciences*, v. 37, n. 13, p. 1512–1520, 2019. Citado na página 36.

DERAKHSHANI, M. et al. Assisted excitation of activations: A learning technique to improve object detectors. In: *CVPR*. Long Beach, USA: IEEE, 2019. p. 10. Citado na página 48.

DIAS, L. M. d. S. et al. Predição de classes de solo por mineração de dados em área da bacia sedimentar do São Francisco. *Pesquisa Agropecuária Brasileira*, scielo, v. 51, p. 1396 – 1404, 09 2016. ISSN 0100-204X. Citado na página 21.

DOLLAR, P. et al. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 34, n. 4, p. 743–761, 2012. Citado na página 52.

DRUCK, S. et al. *Análise espacial de dados geográficos*. 3rd. ed. Brasília: Embrapa, 2004. ISBN 85-7383-260-6. Citado na página 25.

DUAN, K. et al. Centernet: Keypoint triplets for object detection. *CoRR*, abs/1904.08189, 2019. Disponível em: <http://arxiv.org/abs/1904.08189>. Citado na página 47.

ELWOOD, S.; GOODCHILD, M. F.; SUI, D. Z. Researching Volunteered Geographic Information: Spatial Data, Geographic Research, and New Social Practice. *Annals of the Association of American Geographers*, v. 102, n. 3, p. 571–590, 2012. Citado na página 19.

ENZWEILER, M.; GAVRILA, D. M. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 12, p. 2179–2195, 2009. Citado na página 52.

ERHAN, D. et al. Scalable object detection using deep neural networks. *CoRR*, abs/1312.2249, 2014. Disponível em: <http://arxiv.org/abs/1312.2249>. Citado na página 45.

FARIA, E. L. *Redes Neurais Convolucionais e Máquinas de Aprendizado extremo aplicadas ao mercado financeiro brasileiro*. 147 f. Tese (Doutorado em Engenharia Civil) — COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2018. Citado 2 vezes nas páginas 39 e 40.

FERNANDES, M. A. C. *Redes Neurais Artificiais Aplicadas à Detecção Inteligente de Sinais*. 100 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Norte. Natal, 1999. Citado 3 vezes nas páginas 35, 36 e 37.

FRANCO, C. R. *Inteligência Artificial*. Londrina: Uniasselvi, 2014. 168 p. ISBN 978-85-68075-77-7. Citado 3 vezes nas páginas 33, 34 e 35.

GERHARDT, T.; SILVEIRA, D. *Métodos de Pesquisa*. PLAGEDER, 2009. (Série Educação a Distância - UFRGS). ISBN 9788538600718. Disponível em: <https://books.google.com.br/books?id=dRuzRyEIzmkC>. Acesso em: 27/05/2020. Citado na página 60.

GERONIMO, D. et al. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 7, p. 1239–1258, 2010. Citado na página 52.

GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11 2013. Citado na página 43.

GIRSHICK, R. B. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. Disponível em: <http://arxiv.org/abs/1504.08083>. Citado 2 vezes nas páginas 43 e 44.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Acesso em: 23/04/2020. Citado 5 vezes nas páginas 38, 39, 40, 41 e 42.

GOOGLE MAPS PLATFORM. *Map and Tile Coordinates*. 2020. Disponível em: <https://developers.google.com/maps/documentation/javascript/coordinates>. Acesso em: 01/07/2020. Citado na página 56.

GOOGLE MAPS PLATFORM. *Street View Static API*. 2021. Disponível em: <https://developers.google.com/maps/documentation/streetview/intro>. Acesso em: 22/04/2021. Citado 3 vezes nas páginas 61, 62 e 64.

GRAPROHAB. *Manual de Aprovação de Projetos Habitacionais*. 3rd. ed. São Paulo: Secretaria de Habitação do Estado de São Paulo, 2019. 138 p. Disponível em: <http://www.habitacao.sp.gov.br/icone/detalhe.aspx?Id=72>. Citado na página 31.

GUIMARÃES RAFAELLA OLIVEIRA, A. H. N. C.; SANTOS, B. J. R. dos. Verificação da acessibilidade nas calçadas do setor central de goiânia, go. *Multi-Science Journal*, v. 1, n. 2, p. 83 – 91, 2018. ISSN 2359-6902. Citado 2 vezes nas páginas 17 e 32.

HARA, K.; FROEHLICH, J. E. Characterizing and visualizing physical world accessibility at scale using crowdsourcing, computer vision, and machine learning. *SIGACCESS Access. Comput.*, Association for Computing Machinery, New York, NY, USA, n. 113, p. 13–21, nov. 2015. ISSN 1558-2337. Disponível em: <https://doi.org/10.1145/2850440.2850442>. Citado 5 vezes nas páginas 18, 21, 51, 54 e 57.

HARA, K. et al. Tohme: Detecting Curb Ramps in Google Street View Using Crowdsourcing, Computer Vision, and Machine Learning. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2014. (UIST '14), p. 189–204. ISBN 9781450330695. Disponível em: <https://doi.org/10.1145/2642918.2647403>. Citado 2 vezes nas páginas 51 e 54.

HE, K.; GIRSHICK, R.; DOLLAR, P. Rethinking imagenet pre-training. In: *Proceedings of the International Conference on Computer Vision*. Seoul, Korea: IEEE, 2019. p. 4918–4927. Citado na página 68.

HE, K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 37, 06 2014. Citado 2 vezes nas páginas 43 e 44.

HOIEM, D.; EFROS, A. A.; HEBERT, M. Geometric context from a single image. In: *International Conference on Computer Vision*. Beijing, China: IEEE, 2005. v. 1, p. 654–661. Citado na página 59.

HOIEM, D.; EFROS, A. A.; HEBERT, M. Putting objects in perspective. In: *Computer Society Conference on Computer Vision and Pattern Recognition*. New York, USA: IEEE, 2006. v. 2, p. 2137–2144. ISSN 1063-6919. Citado na página 59.

HORAUD, R. et al. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, v. 47, n. 1, p. 33–44, 1989. Citado na página 59.

HUBEL, D. H.; WIESEL, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, v. 148, n. 3, p. 574–591, oct 1959. ISSN 0022-3751. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/14403679https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/>. Citado na página 40.

HUMPHREYS, L.; LIAO, T. Mobile geotagging: Reexamining our interactions with urban space. *Journal of Computer-Mediated Communication*, v. 16, n. 3, p. 407–423, 04 2011. ISSN 1083-6101. Disponível em: <https://doi.org/10.1111/j.1083-6101.2011.01548.x>. Citado 2 vezes nas páginas 56 e 57.

IBGE. *Censo Demográfico 2010*. Instituto Brasileiro de Geografia e Estatística, 2010. Disponível em: <https://www.ibge.gov.br/estatisticas>. Acesso em: 15/05/2020. Citado 10 vezes nas páginas 16, 17, 18, 23, 24, 25, 28, 29, 63 e 74.

IBGE. *Mapas Interativos Brasil 1 por 1: Rampa para cadeirante -23.5634396,-46.6603714*. Instituto Brasileiro de Geografia e Estatística, 2010. Disponível em: <http://mapasinterativos.ibge.gov.br/atlas_ge/brasil1por1.html>. Acesso em: 25/05/2020. Citado na página 20.

IBGE. *Perfil dos municípios brasileiros: 2017*. Instituto Brasileiro de Geografia e Estatística, 2017. Disponível em: <https://agenciadenoticias.ibge.gov.br/media/com_mediaibge/arquivos/496bb4fbf305cca806aaa167aa4f6dc8.pdf>. Acesso em: 15/05/2020. Citado 2 vezes nas páginas 17 e 23.

ISMAGILOVA, E. et al. Smart cities: Advances in research — an information systems perspective. *International Journal of Information Management*, v. 47, p. 88–100, 2019. Citado 3 vezes nas páginas 18, 20 e 21.

JANAI, J. et al. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *CoRR*, abs/1704.05519, 2017. Disponível em: <http://arxiv.org/abs/1704.05519>. Citado na página 51.

JUNIOR, L. C. M.; COLOVAN, J. A. U. Aplicação de Redes Neurais Profundas para detecção e classificação de plantas daninhas e seu estado da arte. *REGRAD - Revista Eletrônica de Graduação do UNIVEM*, v. 11, n. 1, p. 391–403, 2018. ISSN 1984-7866. Disponível em: <file:///C:/Users/Tatiane/Downloads/2638-85-5678-1-10-20180828.pdf>. Citado na página 40.

KRYLOV, V. A.; KENNY, E.; DAHYOT, R. Automatic Discovery and Geotagging of Objects from Street View Imagery. *Remote Sensing*, v. 10, n. 5, p. 661–681, 2018. Disponível em: <https://www.mdpi.com/2072-4292/10/5/661{#}ci>. Citado 3 vezes nas páginas 56, 57 e 59.

KUMAR, V. et al. Multiple Object Detection in 360° Videos for Robust Tracking. In: ____. *Pattern Recognition and Machine Intelligence - Part II*. Tezpur, India: [s.n.], 2019. p. 499–506. ISBN 978-3-030-34871-7. Citado 2 vezes nas páginas 55 e 57.

LAINA, I. et al. Deeper depth prediction with fully convolutional residual networks. *Computing Research Repository*, p. 239–248, 2016. Citado na página 59.

LAW, H.; DENG, J. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, p. 14, 08 2019. Citado na página 47.

LAW, S.; PAIGE, B.; RUSSELL, C. Take a look around. *ACM Transactions on Intelligent Systems and Technology*, Association for Computing Machinery (ACM), v. 10, n. 5, p. 1–19, Nov 2019. ISSN 2157-6912. Disponível em: <http://dx.doi.org/10.1145/3342240>. Citado na página 54.

LAWRENCE, J. et al. Comparing tensorflow deep learning performance using cpus, gpus, local pcs and cloud. In: *Proceedings of Student-Faculty Research Day*. New York, NY, USA: [s.n.], 2017. p. C1–1–C1–7. Disponível em: <https://academicworks.cuny.edu/bx_pubs/50/>. Citado na página 50.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/nature14539>. Citado 3 vezes nas páginas 37, 38 e 52.

LEITE, F. P. A. A promoção da acessibilidade para as pessoas com deficiência: a observância das normas e do desenho universal. *Âmbito Jurídico*, v. 95, n. 1, 10 2011. Citado na página 30.

LETCHFORD, A.; ZARZELLI, A.; BERRIEL, R. *Google Street View Panorama Image Downloader*. 2018. Disponível em: <https://github.com/robolyst/streetview>. Acesso em: 16/03/2021. Citado na página 61.

LI, F.-F.; JOHNSON, J.; YEUNG, S. *Lecture 11:Detection and Segmentation*. Universidade Stanford. 2017. Disponível em: <http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf>. Acesso em: 25/04/2020. Citado na página 41.

LI, X.-j.; RATTI, C.; SEIFERLING, I. Mapping urban landscapes along streets using google street view. In: *International Cartographic Conference*. Cham, Suíça: Springer, 2017. p. 341–356. ISBN 978-3-319-57335-9. Citado na página 54.

LI, X.-j. et al. Assessing street-level urban greenery using google street view and a modified green view index. *Urban Forestry & Urban Greening*, v. 14, n. 3, p. 675–685, 06 2015. Citado na página 54.

LIN, T. et al. Focal loss for dense object detection. *ICCV*, 2017. Disponível em: <http://arxiv.org/abs/1708.02002>. Citado na página 47.

LIN, T. et al. Microsoft COCO: common objects in context. *Computing Research Repository*, abs/1405.0312, 2014. Disponível em: <http://arxiv.org/abs/1405.0312>. Citado na página 64.

LIU, L. et al. Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, v. 128, n. 2, p. 261–318, 2020. ISSN 1573-1405. Disponível em: <https://doi.org/10.1007/s11263-019-01247-4>. Citado 8 vezes nas páginas 21, 38, 41, 42, 45, 46, 50 e 52.

LIU, W. et al. Ssd: Single shot multibox detector. In: LEIBE, B. et al. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 21–37. ISBN 978-3-319-46448-0. Citado na página 46.

LéVY, J. Os novos espaços da mobilidade. *GEOgraphia*, v. 3, n. 6, 2001. Citado na página 16.

MACHADO, M. H.; LIMA, J. P. Avaliação multicritério da acessibilidade de pessoas com mobilidade reduzida: um estudo na região central de itajubá (mg). *URBE. Revista Brasileira de Gestão Urbana*, scielo, v. 7, p. 368 – 382, 12 2015. ISSN 2175-3369. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2175-33692015000300368&nrm=iso>. Citado na página 16.

MAHALLEH, V. B. S.; ALQUTAMI, T. A.; MAHMOOD, I. A. YOLO-Based Valve Type Recognition and Localization. In: *International Conference on Industrial Engineering and Applications*. Tokyo, Japan: [s.n.], 2019. p. 37–40. Citado 5 vezes nas páginas 69, 70, 73, 79 e 82.

MAIA, A. G.; QUADROS, W. J. d. Tipologia municipal de classes sociocupacionais: uma nova dimensão para análise das desigualdades territoriais no brasil. *Revista de Economia e Sociologia Rural*, scielo, v. 47, n. 2, p. 389 – 418, 06 2009. ISSN 0103-2003. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-20032009000200004&nrm=iso>. Citado na página 30.

MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2. Citado na página 33.

MOBASHERI, A. et al. Enrichment of OpenStreetMap Data Completeness with Sidewalk Geometries Using Data Mining Techniques. *Sensors (Basel, Switzerland)*, v. 18, n. 2, feb 2018. ISSN 1424-8220 (Electronic). Citado na página 19.

MOBILIZE BRASIL. *Relatório da Campanha Calçadas do Brasil*. Mobilize Mobilidade Urbana Sustentável, 2013. Disponível em: <https://www.mobilize.org.br/midias/pesquisas/relatorio-calcadas-do-brasil---jan-2013.pdf>. Acesso em: 25/05/2020. Citado na página 20.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. 2. ed. New York, USA: MIT Press, 2018. ISBN 0262039400. Citado 2 vezes nas páginas 33 e 34.

MONGEON, P.; PAUL-HUS, A. The journal coverage of Web of Science and Scopus: a comparative analysis. *Scientometrics*, v. 106, n. 1, p. 213–228, jan 2016. ISSN 1588-2861. Disponível em: <https://doi.org/10.1007/s11192-015-1765-5>. Citado na página 50.

MONTERO, A. S. et al. Framework for natural landmark-based robot localization. In: *Ninth Conference on Computer and Robot Vision*. Toronto, Canadá: IEEE, 2012. p. 131–138. Citado na página 59.

NAJIBI, M.; RASTEGARI, M.; DAVIS, L. S. G-CNN: an iterative grid based object detector. *CoRR*, abs/1512.07729, 2015. Disponível em: <http://arxiv.org/abs/1512.07729>. Citado na página 46.

NDONHONG, V.; BAO, A.; GERMAIN, O. Wellbore schematics to structured data using artificial intelligence tools. In: *Offshore Technology Conference*. Houston, USA: Offshore Technology Conference, 2019. p. 18. Citado na página 45.

NEIS, P.; ZIELSTRA, D. Generation of a tailored routing network for disabled people based on collaboratively collected geodata. *Applied Geography*, v. 47, p. 70–77, 2014. Citado 2 vezes nas páginas 19 e 20.

NELSON, J. *The Importance of Blur as an Image Augmentation Technique*. 2020. Disponível em: <https://blog.roboflow.com/using-blur-in-computer-vision-preprocessing/>. Acesso em: 20/08/2021. Citado na página 67.

NELSON, J.; SOLAWETZ, J. *Responding to the Controversy about YOLOv5*. 2020. Disponível em: <https://blog.roboflow.com/yolov4-versus-yolov5/>. Acesso em: 20/04/2021. Citado na página 48.

NESKOROZHENYI, R. *Yolo-Tiling: Tile (Slice) YOLO Dataset for Small Objects Detection*. 2021. Disponível em: <https://github.com/slanj/yolo-tiling>. Acesso em: 20/04/2021. Citado na página 67.

NETO, C. A. A.; ROLT, C. R. de; ALPERSTEDT, G. D. Accessibility and Technology in Smart City Construction/Acessibilidade e Tecnologia na Construcao da Cidade Inteligente. *RAC - Revista de Administracao Contemporanea*, v. 22, p. 291–310, may 2018. ISSN 14156555. Disponível em: <https://link.gale.com/apps/doc/A536388793/AONE?u=capes{&}sid=AONE{&}xid=03>. Citado na página 18.

NILSSON, N. J. *Introduction to Machine Learning: An Early Draft of a Proposed Textbook*. 1998. 179 p. Disponível em: <https://ai.stanford.edu/{~}nilsson/MLBOOK.p>. Acesso em: 22/04/2020. Citado 2 vezes nas páginas 34 e 36.

NVIDIA. *Introduction to NVIDIA GPU Cloud*. 2018. Disponível em: <https://docs.nvidia.com/ngc/pdf/ngc-introduction.pdf>. Citado na página 50.

NVIDIA. *Develop, Optimize and Deploy GPU-accelerated Apps*. 2020. Disponível em: <https://developer.nvidia.com/cuda-toolkit>. Citado na página 50.

NVIDIA. *NVIDIA cuDNN*. 2020. Disponível em: <https://developer.nvidia.com/cudnn>. Citado na página 50.

ONU. *Good Practice of Accessible Urban Development*. Organização das Nações Unidas, 2016. Disponível em: <https://www.un.org/disabilities/documents/desa/good_practices_in_accessible_urban_development_october2016.pdf>. Acesso em: 10/05/2019. Citado na página 16.

OSM. *OpenStreetMap stats report*. OpenStreetMap, 2020. Disponível em: <https://www.openstreetmap.org/stats/data_stats.html>. Acesso em: 11/05/2020. Citado na página 19.

OSMSURROUND. *OSM Quality Assurance Editor -23.5634396,-46.6603714*. 2020. Disponível em: <http://editor.osmsurround.org/>. Acesso em: 25/05/2020. Citado na página 20.

OSRM. *Open Source Routing Machine Project*. Open Source Routing Machine, 2020. Disponível em: <http://project-osrm.org/>. Acesso em: 12/05/2020. Citado na página 19.

O'GARA, S.; MCGUINNESS, K. Comparing data augmentation strategies for deep image classification. In: *Irish Machine Vision Image Processing*. Dublin, Irlanda: [s.n.], 2019. Citado na página 67.

PATHAK, A. R.; PANDEY, M.; RAUTARAY, S. Application of Deep Learning for Object Detection. *Procedia Computer Science*, v. 132, p. 1706–1717, 2018. ISSN 1877-0509. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1877050918308767>. Citado na página 42.

PINHEIRO, P. H. O.; COLLOBERT, R.; DOLLÁR, P. Learning to segment object candidates. *CoRR*, abs/1506.06204, 2015. Disponível em: <http://arxiv.org/abs/1506.06204>. Citado na página 45.

PINTO, P. d. C. C. *Implementation of Faster R-CNN Applied to the Datasets COCO and PASCAL VOC*. 69 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2019. Citado na página 71.

PITTS, W.; MCCULLOCH, W. S. How we know universals the perception of auditory and visual forms. *The bulletin of mathematical biophysics*, v. 9, n. 3, p. 127–147, 1947. ISSN 1522-9602. Disponível em: <https://doi.org/10.1007/BF02478291>. Citado na página 34.

POUYANFAR, S. et al. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, v. 51, n. 5, 2018. Citado na página 52.

RAY, S. *Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video, Electronic and Digital Imaging*. Focal Press, 2002. ISBN 9780240515403. Disponível em: <https://books.google.com.br/books?id=cuzYl4hx-B8C>. Citado na página 56.

REDMON, J. et al. You Only Look Once: Unified, Real-Time Object Detection. In: *Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, 2015. p. 779–788. Disponível em: <http://arxiv.org/abs/1506.02640>. Citado 2 vezes nas páginas 46 e 47.

REDMON, J.; FARHADI, A. Yolo9000: Better, faster, stronger. In: *Conference on Computer Vision and Pattern Recognition*. Honolulu, USA: IEEE, 2017. p. 6517–6525. Citado na página 47.

REDMON, J.; FARHADI, A. YOLOv3: An Incremental Improvement. Washington, p. 6, 2018. Disponível em: <https://pjreddie.com/media/files/papers/YOLOv3.pdf>. Citado 2 vezes nas páginas 47 e 68.

REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Disponível em: <http://arxiv.org/abs/1506.01497>. Citado na página 44.

REZENDE, R. P. d. *Mapeamento e Gestão de Sistemas de Infraestrutura Urbana: Metodologia Aplicada em Sistemas Informacionais*. 83 p. Dissertação (Mestrado em Engenharia Civil) — Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2019. Citado na página 21.

RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, v. 115, p. 211–252, 2015. Disponível em: <https://link.springer.com/article/10.1007/s11263-015-0816-y>. Citado 3 vezes nas páginas 42, 63 e 65.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. USA: Prentice Hall Press, 2009. ISBN 0136042597. Citado na página 37.

SADEGHI, H.; VALAEE, S.; SHIRANI, S. Semi-supervised logo-based indoor localization using smartphone cameras. In: *International Symposium on Personal, Indoor and Mobile Radio Communications*. Washington, USA: IEEE, 2014. v. 2014-June, p. 2024–2028. Citado na página 59.

SADEGHI, H.; VALAEE, S.; SHIRANI, S. A weighted knn epipolar geometry-based approach for vision-based indoor localization using smartphone cameras. In: *Proceedings of the Sensor Array and Multichannel Signal Processing Workshop*. Coruna, Spain: IEEE, 2014. p. 37–40. Citado na página 59.

SADEGHI, H.; VALAEE, S.; SHIRANI, S. Ocrapose: An indoor positioning system using smartphone/tablet cameras and ocr-aided stereo feature matching. In: *International Conference on Acoustics, Speech and Signal Processing*. Brisbane, Austrália: IEEE, 2015. p. 1473–1477. Citado na página 59.

SADEGHI, H.; VALAEE, S.; SHIRANI, S. 2dtripnp: A robust two-dimensional method for fine visual localization using google streetview database. *IEEE Transactions on Vehicular Technology*, v. 66, n. 6, p. 4678–4690, jun 2017. ISSN 1939-9359. Citado na página 59.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 3, n. 3, p. 210–229, 1959. Citado na página 33.

SERMANET, P. et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. *International Conference on Learning Representations (ICLR) (Banff)*, 12 2013. Citado na página 45.

SHAPIRO, A. Street-level: Google street view's abstraction by datafication. *New Media Society*, SAGE Publications, London, England, v. 20, n. 3, p. 1201–1219, 2018. ISSN 1461-4448. Citado na página 63.

SHEN, S. *Introdução ao aprendizado profundo*. 2018. Disponível em: <https://medium.com/@syshen/>. Acesso em: 23/04/2020. Citado na página 40.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, v. 6, n. 1, p. 60, 2019. ISSN 2196-1115. Disponível em: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0#Sec3>. Citado na página 67.

SILVA FILHO, A. M. Desafios e tecnologias para cidades do futuro. *Revista Espaço Acadêmico*, Universidade Estadual de Maringá, v. 11, n. 132, p. 75–78, 2012. ISSN 1519-6186. Disponível em: <https://doaj.org/article/4b765caa691549b1a77523f694f34722>. Citado na página 18.

SILVA, J. B. d. et al. Wayfinding em aplicativos de recomendação de rota: coerência com mapas cognitivos. In: *Anais do 15º Ergodesign & Usihc*. São Paulo: Blucher, 2015. v. 2, p. 1161–1173. Citado na página 19.

SILVEIRA, F. T. J. e Rogério Leandro Lima da. Crescimento demográfico e urbanização em municípios de porte médio: alterações na dinâmica urbana regional do rio grande do sul. *Revista Brasileira de Gestão e Desenvolvimento Regional*, v. 16, n. 3, 2020. ISSN 1809-239X. Disponível em: <https://www.rbgdr.net/revista/index.php/rbgdr/article/view/5873>. Citado na página 30.

SOUZA, L. *Pessoas com deficiência física criticam falta de acessibilidade em SP*. São Paulo: Agência Brasil, 2019. Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2019-09/pessoas-com-deficiencia-fisica-criticam-falta-de-acessibilidade-em-sp>. Acesso em: 12/12/2020. Citado 3 vezes nas páginas 18, 23 e 64.

SUN, H. et al. Surrounding moving obstacle detection for autonomous driving using stereo vision. *International Journal of Advanced Robotic Systems*, v. 10, n. 6, p. 261, 2013. Disponível em: <https://doi.org/10.5772/56603>. Citado na página 57.

SUN, Z.; BEBIS, G.; MILLER, R. On-road vehicle detection: a review. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 28, n. 5, p. 694–711, 2006. Citado na página 52.

SZEGEDY, C.; TOSHEV, A.; ERHAN, D. Deep neural networks for object detection. In: . [S.l.: s.n.], 2013. p. 1–9. Citado na página 45.

SZELISKI, R. Recognition. In: ____. *Computer Vision: Algorithms and Applications*. London: Springer London, 2011. p. 575–640. ISBN 978-1-84882-935-0. Citado na página 42.

TASHIEV, I. *XmlToTxt*. 2020. Disponível em: <https://github.com/Isabek/XmlToTxt>. Acesso em: 01/07/2020. Citado na página 65.

TESLA. *TESLA V100 Performance Guide: Deep Learning and HPC Applications*. 2017. Disponível em: <https://h20195.www2.hpe.com/v2/getdocument.aspx?docname=a00040438enw#>. Citado na página 50.

TORII, A.; SIVIC, J.; PAJDLA, T. Visual localization by linear combination of image descriptors. In: *Proceedings of the International Conference on Computer Vision*. Barcelona: IEEE, 2011. p. 102–109. Citado na página 59.

TSAI, V.; CHANG, C.-T. Three-dimensional positioning from Google street view panoramas. *IET Image Process.*, v. 7, n. 3, p. 229–239, 2013. Disponível em: <https://www.semanticscholar.org/paper/Three-dimensional-positioning-from-Google-street-Tsai-Chang/70f5c09aec1199e4a85993421f2e184c503fbf0b>. Citado 2 vezes nas páginas 57 e 59.

TSAI, V. J. D.; CHANG, C. FEATURE POSITIONING ON GOOGLE STREET VIEW PANORAMAS. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Melbourne, Australia: [s.n.], 2012. p. 305–309. Citado na página 59.

TZUTA, L. *LabelImg*. 2018. Disponível em: <https://github.com/tzutalin/labelImg>. Acesso em: 16/03/2021. Citado na página 65.

UCAR, A.; DEMIR, Y.; GUZELIS, C. Object recognition and detection with deep learning for autonomous driving applications. *SIMULATION*, v. 93, p. 003754971770993, 06 2017. Citado na página 51.

VAHTRA, R.; ANBARJAFARI, G. *Parking Space Monitoring and ID Based Car Tracking*. 51 p. Dissertação (Mestrado em Ciência da Computação) — University of Tartu. Tastu, Estônia, 2019. Citado 2 vezes nas páginas 54 e 68.

WANG, J.; ZHA, H.; CIPOLLA, R. Coarse-to-fine vision-based localization by indexing scale-invariant features. *Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE, v. 36, n. 2, p. 413–422, 2006. Citado na página 59.

WANG, L.; GUO, B.; YANG, Q. Smart city development with urban transfer learning. *Computer*, v. 51, n. 12, p. 32–41, 2018. Citado 2 vezes nas páginas 20 e 21.

WANGENHEIM, A. V. *Deep Learning: Detecção de Objetos em Imagens*. 2018. Disponível em: <http://www.lapix.ufsc.br/ensino/visao/visao-computacionaldeep-learning/deteccao-de-objetos-em-imagens/{#}Classificadores{_}de{_}Regioes{_}associados{_}a{_}Extratores{_}de{_}Caracter>. Citado 3 vezes nas páginas 43, 44 e 46.

WANLI, M. I. N. et al. People logistics in smart cities. *Communications of the ACM*, v. 61, n. 11, p. 54–59, 2018. ISSN 00010782. Citado 2 vezes nas páginas 21 e 53.

WEISS, M. C.; BERNARDES, R. C.; CONSONI, F. L. Cidades inteligentes como nova prática para o gerenciamento dos serviços e infraestruturas urbanos: a experiência da cidade de porto alegre. *URBE. Revista Brasileira de Gestão Urbana*, scielo, v. 7, p. 310 – 324, 12 2015. ISSN 2175-3369. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S2175-33692015000300310&nrm=iso>. Citado na página 21.

WELD, G. et al. Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery. In: *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. New York, NY, USA: Association for Computing Machinery, 2019. (ASSETS '19), p. 196–209. ISBN 9781450366762. Disponível em: <https://doi.org/10.1145/3308561.3353798>. Citado 2 vezes nas páginas 51 e 57.

WELZEL, A.; REISDORF, P.; WANIELIK, G. Improving urban vehicle localization with traffic sign recognition. In: *Proceedings of the Conference on Intelligent Transportation Systems*. Las Palmas, Spain: IEEE, 2015. p. 2728–2732. Citado na página 59.

WU, X.; SAHOO, D.; HOI, S. C. H. Recent advances in deep learning for object detection. *Neurocomputing*, 2020. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231220301430>. Citado 5 vezes nas páginas 46, 47, 52, 71 e 73.

XIAO, J. et al. Recognizing scene viewpoint using panoramic place representation. In: *Conference on Computer Vision and Pattern Recognition*. Providence, USA: IEEE, 2012. p. 2695–2702. Citado na página 59.

YANG, Y. et al. Pose estimation based on four coplanar point correspondences. In: *International Conference on Fuzzy Systems and Knowledge Discovery*. China: IEEE, 2009. v. 5, p. 410–414. Citado na página 59.

YAO, Y. et al. Towards automatic construction of diverse, high-quality image datasets. *IEEE Transactions on Knowledge and Data Engineering*, v. 32, n. 6, p. 1199–1211, 2020. Citado 2 vezes nas páginas 63 e 64.

YOO, D. et al. Attentionnet: Aggregating weak directions for accurate object detection. *CoRR*, abs/1506.07704, 2015. Disponível em: <http://arxiv.org/abs/1506.07704>. Citado na página 46.

YU, L. et al. Monocular urban localization using street view. In: *International Conference on Control, Automation, Robotics and Vision*. Phuket, Thailândia: IEEE, 2016. p. 1–6. Citado na página 59.

ZAWORSKI, R. *Data Augmentation Techniques and Pitfalls for Small Datasets*. 2018. Disponível em: <https://snow.dog/blog/data-augmentation-for-small-datasets>. Acesso em: 21/08/2021. Citado na página 67.

ZHANG, F.; LI, C.; YANG, F. Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. *Sensors*, v. 19, p. 21, 01 2019. Citado na página 48.

ZHAO, Z.-Q. et al. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, v. 30, n. 11, p. 3212–3232, 2019. Citado 8 vezes nas páginas 34, 35, 38, 43, 44, 45, 47 e 52.

ZHOU, X.; WANG, D.; KRÄHENBÜHL, P. Objects as points. *CoRR*, abs/1904.07850, 2019. Disponível em: <http://arxiv.org/abs/1904.07850>. Citado na página 47.

ZIPF, A. et al. Crowdsourcing for individual needs – the case of routing and navigation for mobility-impaired persons. In: ____. *European Handbook of Crowdsourced Geographic Information*. London: Ubiquity Press, 2016. p. 325–337. ISBN 978-1-909188-80-8. Citado na página 20.

Ünel, F. ; Özkalayci, B. O.; Çiğla, C. The power of tiling for small object detection. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2019. p. 582–591. Citado 2 vezes nas páginas 66 e 67.

APÊNDICE A – *Script* Python para aquisição de imagens do GSV

```
"""
Arquivo principal da ferramenta Google Street View Panorama
Image Downloader, modificado para baixar multiplas imagens.
"""

#Nome do arquivo: usage_multiplescoordinates.py
#Criado por: Tatiane Ferreira Olivatto
#Criado em: 14 de dezembro de 2020.

import streetview
#Disponivel em: https://github.com/robolyst/streetview
import matplotlib.pyplot as plt
import pandas as pd
import shutil

dados = pd.read_excel('*.xlsx')

panoidList = []
lat\_pano = []
lon\_pano = []
data = []

for index,row in dados.iterrows():
    lati = row['lat']
    long =  row['long']

    Str_pano = str(int(row['id']))

    try:
        panoids = streetview.panoids(lat=lati, lon=long)
        panoid = panoids[-1]['panoid']
        panorama = streetview.download_panorama_v3(
        panoid, zoom=2, disp=False)
        plt.imsave('*/pano'+Str_pano+'.png', panorama)
        panoidList.append(panoid)
```

```
            lat_pano.append(panoids[-1]['lat'])
            lon_pano.append(panoids[-1]['lon'])
            data.append(panoids[-1]['year'])
        except:
            panoidList.append(0)
            lat_pano.append(0)
            lon_pano.append(0)
            data.append(0)

dados['panoid'] = panoidList
dados['lat_pano'] = lat_pano
dados['lon_pano'] = lon_pano
dados['year'] = data
dados.to_excel('*_pano.xlsx', index=False)
```

ANEXO A – *Script* Python para criação dos arquivos *train* e *test*

```
#generate_train.py
import os

image_files = []
os.chdir(os.path.join("data", "obj"))
for filename in os.listdir(os.getcwd()):
    if filename.endswith(".jpg"):
        image_files.append("data/obj/" + filename)
os.chdir("..")
with open("train.txt", "w") as outfile:
    for image in image_files:
        outfile.write(image)
        outfile.write("n")
    outfile.close()
os.chdir("..")

#generate_test.py
import os

image_files = []
os.chdir(os.path.join("data", "test"))
for filename in os.listdir(os.getcwd()):
    if filename.endswith(".jpg"):
        image_files.append("data/test/" + filename)
os.chdir("..")
with open("test.txt", "w") as outfile:
    for image in image_files:
        outfile.write(image)
        outfile.write("n")
    outfile.close()
os.chdir("..")
```