

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Geociências
Programa de Pós-Graduação em Análise e Modelagem de Sistemas Ambientais

Antônio Campos de Abreu Filho

**DETECÇÃO DE PAINÉIS FOTOVOLTAICOS EM ORTOFOTOS UTILIZANDO
REDES NEURAS PROFUNDAS**

Belo Horizonte

2021

Antônio Campos de Abreu Filho

Detecção de painéis fotovoltaicos em ortofotos utilizando redes neurais profundas

Versão Final

Dissertação apresentada ao Programa de Pós Graduação em Análise e Modelagem de Sistemas Ambientais da Universidade Federal de Minas Gerais como requisito parcial à obtenção do título de Mestre em Análise e Modelagem de Sistemas Ambientais.

Orientador: Prof. Dr. Marcelo Antonio Nero

Coorientador: Prof. Dr. Plínio Temba

Belo Horizonte

2021

A162d
2021

Abreu Filho, Antônio Campos de.

Detecção de painéis fotovoltaicos em ortofotos utilizando redes neurais profundas [manuscrito] / Antônio Campos de Abreu Filho. – 2021.
114 f., enc.: il. (principalmente color.)

Orientador: Marcelo Antonio Nero.

Coorientador: Plínio da Costa Temba.

Dissertação (mestrado) – Universidade Federal de Minas Gerais, Departamento de Cartografia, 2021.

Bibliografia: f. 101-114.

1. Modelagem de dados – Aspectos ambientais – Teses. 2. Energia solar – Teses. 3. Geração de energia fotovoltaica – Belo Horizonte (MG) – Teses. 4. Geoprocessamento – Teses. I. Nero, Marcelo Antonio. II. Temba, Plínio da Costa. III. Universidade Federal de Minas Gerais. Departamento de Cartografia. IV. Título.

CDU: 911.2:519.6(815.1)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ANÁLISE E MODELAGEM DE SISTEMAS AMBIENTAIS

FOLHA DE APROVAÇÃO

DETECÇÃO DE PAINÉIS FOTOVOLTAICOS EM ORTOFOTOS UTILIZANDO REDES NEURAS PROFUNDAS

ANTÔNIO CAMPOS DE ABREU FILHO

Dissertação submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ANÁLISE E MODELAGEM DE SISTEMAS AMBIENTAIS, como requisito para obtenção do grau de Mestre em ANÁLISE E MODELAGEM DE SISTEMAS AMBIENTAIS, área de concentração ANÁLISE, MODELAGEM E GESTÃO DE SISTEMAS AMBIENTAIS.

Dr. Marcelo Antonio Nero - Orientador
IGC/UFMG

Dr. Plínio da Costa Temba - Coorientador
IGC/UFMG

Dra. Karla Albuquerque de Vasconcelos Borges
PRODABEL/PBH

Dr. Jefersson Alex dos Santos
DCC/UFMG

Dr. Marcos Antônio Timbó Elmiro
IGC/UFMG

Belo Horizonte, 01 de abril de 2021.



Documento assinado eletronicamente por **Jefersson Alex dos Santos, Membro**, em 01/04/2021, às 17:07, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcos Antonio Timbo Elmiro, Professor do Magistério Superior**, em 01/04/2021, às 17:09, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Plínio da Costa Temba, Professor do Magistério Superior**, em 01/04/2021, às 17:15, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Marcelo Antonio Nero, Professor do Magistério Superior**, em 01/04/2021, às 17:20, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Karla Albuquerque de Vasconcelos Borges, Usuário Externo**, em 05/04/2021, às 11:15, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0636321** e o código CRC **8E6BE57D**.

AGRADECIMENTOS

Aos meus pais, Antônio e Rita, pela paciência e confiança em minhas decisões e, principalmente, pela valorização da educação em nosso lar, muito obrigado!

À minha segunda mãe e meu xodó, Tia Bebel, obrigado pelo amor incondicional que encontro em você!

Às irmãs Guilhermina e Helena, não conseguiria colocar em palavras a gratidão de tê-las como irmãs e melhores amigas. Este trabalho é resultado do apoio e inspiração que encontro em vocês, obrigado!

Aos meus irmãos de consideração, Leonardo Dinali e Guilherme Miranda, muito obrigado pela parceria de sempre.

À Marina Mol, minha companheira, por sua lealdade e amor, muito obrigado.

Ao Professor Marcelo Antonio Nero, obrigado pela orientação, pela liberdade dada no desenvolvimento desta pesquisa e pelo resgate do mestrado que teria sido abandonado, definitivamente, se não fosse sua preciosa intervenção.

Ao Professor Plínio Temba, obrigado por todo o suporte, sua contribuição foi de suma importância para a conclusão desta pesquisa.

Aos membros da banca examinadora, Professora Karla Albuquerque, Professor Timbó e Professor Jefersson, obrigado pelo interesse, disponibilidade e considerações imprescindíveis a este estudo.

Ao Rodrigo Leitão e à Ana Karl, agradeço pela parceria e confiança construídas nestes anos de trabalho conjunto.

Aos colegas de programa, em especial aos queridos André Leal, Beckembauer Araújo, Charles Fonseca, Daniel Salim, Guilherme Gandra, Marcelo Tenenwurcel, Nino Camini e Rafael Tarantino, agradeço às sugestões, discussões, provocações, insistências e críticas que deram forma à este trabalho.

Ao Argemiro Teixeira, minha gratidão por seu profissionalismo e pela brilhante condução da disciplina de Modelagem de Sistemas Ambientais em plena pandemia.

Ao Alfredo Fiuza por seu suporte logístico, extremamente oportuno no curso do mestrado, muito obrigado!

Por fim, agradeço à Alexandra Elbakyan, pela criação da plataforma que permitiu que esta pesquisa tivesse tantas referências de qualidade e que colabora para o acesso à informação e crescimento de toda comunidade científica global.

RESUMO

O mapeamento de informações espaciais tem ganhado cada vez mais relevância em aplicações de planejamento urbano e gerenciamento das cidades. As análises e classificações automatizadas de imagens têm sido um problema em várias áreas do conhecimento e a comunidade do sensoriamento remoto têm demonstrado particular interesse nas aplicações de técnicas que utilizam redes neurais profundas nos problemas de classificação de uso e ocupação do solo. Para o treinamento destas redes, é necessário uma extensa rotulagem para os objetos que se pretende identificar. A segmentação de instâncias, em especial, requer uma rotulagem ainda mais complexa e trabalhosa. O mundo demanda energia e a “Lei de Swanson” prevê que módulos fotovoltaicos alcançarão a paridade de custo por *watt* produzido com outras formas de produção energética em poucos anos. Comparada com outras alternativas, a geração fotovoltaica em telhados possui maior aceitação por utilizar um espaço ocioso das construções urbanas. A compreensão do fenômeno da expansão das instalações fotovoltaicas em áreas urbanas é um importante indicador da evolução de segurança energética das cidades e sua detecção a partir das imagens aéreas constitui uma importante forma de entendê-lo. Este trabalho objetivou treinar uma rede neural profunda utilizando bases de dados de livre acesso do Rio de Janeiro/RJ, Brasília/DF e Campinas/SP. Com o modelo treinado, foi feita uma varredura em ortofotos cobrindo toda a extensão do município de Belo Horizonte/MG em resolução espacial de 7 cm/pixel e foram segmentadas pixel a pixel, todas as placas fotovoltaicas e sistemas de aquecimento solar da área. Foram utilizados 1719 dados rotulados manualmente como amostras de treinamento e a inferência do modelo de segmentação identificou 3655 placas fotovoltaicas e 26369 feições de aquecimento solar na área de estudo. Como métricas de validação do modelo, foram obtidos os valores de precisão média global ($AP=0,18$), *recall* médio global ($AR=0,36$), precisão média da classe de placas fotovoltaicas ($AP_{UFV}=0,30$), precisão média da classe de sistemas de aquecimento solar ($AP_{SAS}=0,16$).

Palavras-chave: energia solar fotovoltaica, aprendizado profundo, segmentação de instâncias, geoprocessamento.

ABSTRACT

Mapping spatial information has increasingly gained relevance in urban planning and city management applications. The automated analysis and classification of images have been a problem in several areas of knowledge and the remote sensing community has shown particular interest in the application of techniques that use deep neural networks in the land use segmentation task. To train these networks, extensive labeling is required for the objects to be identified. Instance segmentation, in particular, requires even more complex and laborious labeling. The world demands energy and "Swanson's Law" predicts that photovoltaic modules will achieve grid parity in a few years. Compared to other alternatives, rooftop photovoltaic generation is more widely accepted for using idle spaces in urban buildings. Studying how photovoltaic installations evolve in urban areas is an important indicator of the evolution of energy security in cities and its detection based on aerial images is an important way of understanding it. This work aimed to train a deep neural network using publicly available data from Rio de Janeiro/RJ, Brasília/DF and Campinas/SP. Using the trained model, an orthophoto, covering the entire area of Belo Horizonte/MG, was segmented, on a 7 cm/pixel spatial resolution, evaluating all photovoltaic modules and solar heating systems in the study area. 1719 manually labeled data were used as training samples and the inference step identified 3655 photovoltaic modules and 26369 solar heating systems in the area. As validation metrics of the model, the following values were obtained: global average precision ($AP=0.18$), global average recall ($AR=0,36$), average precision of solar panel's class ($AP_{UFV}=0,30$) and average precision of solar heating systems class ($AP_{SAS}=0,16$).

Keywords: photovoltaic solar energy, deep learning, instance segmentation, geoprocessing.

LISTA DE ILUSTRAÇÕES

- Figura 1 - Capacidade cumulativa instalada fotovoltaica (em GW, ao final do ano)
- Figura 2 - Custo de produção/watt de módulos FV (em dólares) comparado à capacidade cumulativa instalada (em MW)
- Figura 3 - Representação esquemática de uma imagem em formato RGB
- Figura 4 - Ortofotos RGB com resolução espacial de 7 cm/pixel (à esquerda) e 1m/pixel (à direita)
- Figura 5 - Exemplo esquemático de uma rede neural
- Figura 6 - Camadas empilhadas de aplicações de convoluções, normalizações e pooling dos dados de entrada, contendo a probabilidade de classificação
- Figura 7 - Filtro de convolução 3x3 aplicado à imagem de entrada
- Figura 8 - Camada de normalização ReLU
- Figura 9 - Max pooling com tamanho de janela 2 e stride 2
- Figura 10 - Camada FC com 12 neurônios aplicada na camada normalizada
- Figura 11 - Mapas de ativação extraídos a partir de filtros em diferentes níveis
- Figura 12 - Exemplo da tarefa de classificação de imagens
- Figura 13 - Exemplos da tarefa de detecção de objetos
- Figura 14 - Arquitetura de R-CNN
- Figura 15 - Arquitetura de Fast R-CNN
- Figura 16 - Representação esquemática da arquitetura da Faster R-CNN
- Figura 17 - Arquitetura da RPN
- Figura 18 - Desenho esquemático da métrica IoU
- Figura 19 - Exemplos da tarefa de segmentação de instâncias
- Figura 20 - Desenho esquemático da arquitetura de Mask R-CNN para segmentação de instâncias
- Figura 21 - Arquitetura da FPN
- Figura 22 - Arquitetura da rede head
- Figura 23 - Redimensionamento da máscara predita de 28x28 pixels (à esquerda) para o tamanho da imagem original (à direita)
- Figura 24 - RoI Align
- Figura 25 - Amostra aumentada para 21 novas amostras

Figura 26 - Fotografia de SAS em primeiro plano e UFV em segundo plano

Figura 27 - Exemplo de diferenças intraclasse SAS (destacadas em azul anil) e semelhança de ambas classes SAS e UFV

Figura 28 - Fotografia aérea produzida por drone (acima) comparada a ortofoto do Google (abaixo) no mesmo local

Figura 29 - Localização da área de estudo - Datum: WGS84 - Sistema de Coordenadas Geográficas

Figura 30 - Exemplo de retorno de uma requisição de imagem de *Maps* URLs (à esquerda) e MSA (à direita)

Figura 31 - Exemplos de qualidade das imagens disponibilizadas para os dez municípios mais populosos do Brasil, exceto Belo Horizonte/MG com escala de 1:500

Figura 32 - Fluxograma da metodologia do trabalho

Figura 33 - Distribuição espacial dos pontos UFV (em vermelho) extraídos de ANEEL em (a) Rio de Janeiro/RJ, (b) Brasília/DF e (c) Campinas/SP

Figura 34 - Exemplo do procedimento de anotação dos dados de referência

Figura 35 - Exemplo da geração de pontos aleatórios em torno das anotações

Figura 36 - Exemplo de conversão de anotações em formato vetorial para anotações formato COCO

Figura 37 - Aplicação dos sete procedimentos de aumento de dados em uma imagem das amostras

Figura 38 - Processo de criação do grid vetorial de $0,01^{\circ} \times 0,01^{\circ}$ para composição do mosaico de ortofotos

Figura 39 - Exemplo da geração de um dos componentes do mosaico raster a partir de dados extraídos do *Maps* URLs

Figura 40 - Área de cobertura das janelas deslizantes (azul) na área de estudo e extração de imagem de 600x600 a partir de uma janela deslizante

Figura 41 - Localização das anotações UFV e SAS no Rio de Janeiro/RJ (a), Brasília/DF (b) e Campinas/SP (c)

Figura 42 - Evolução da acurácia e erro durante as iterações do treinamento

Figura 43 - Exemplo de problemas topológicos (a) e (c) corrigidos por pós-processamento (b) e (d)

Figura 44 - Distribuição espacial das UFV detectadas na área de estudo

Figura 45 - Distribuição espacial dos SAS detectadas na área de estudo

Figura 46 - Exemplos de ambiguidade dos dados

Figura 47 - UFV detectados no Estádio Mineirão com falsos positivos

Figura 48 - Locais com grandes composições de módulos fotovoltaicos corretamente segmentados

Figura 49 - Feições grandes de módulos fotovoltaicos corretamente segmentadas com diversos falsos positivos

Figura 50 - Distorções nas ortofotos onde edificações foram incorretamente classificadas como UFV

Figura 51 - Subjetividade dos valores de referência

Figura 52 - Exemplos de falsos positivos preditos pelo modelo na área de estudo

Figura 53 - Exemplo de classificação ambígua na borda de uma janela deslizante

LISTA DE TABELAS

Tabela 1 - Lista dos dez municípios mais populosos do Brasil

Tabela 2 - Resumo dos dados extraídos de ANEEL contendo pontos UFV

Tabela 3 - Número de feições anotadas para treinamento

Tabela 4 - Métricas de validação do modelo treinado para diferentes tamanhos de objetos

Tabela 5 - Número de feições encontradas na área de estudo por classe e etapa de processamento

LISTA DE QUADROS

Quadro 1 - Especificações de *hardware* do ambiente de desenvolvimento

Quadro 2 - Técnicas de aumento de dados aplicadas à todas as imagens

LISTA DE EQUAÇÕES

Equação 1 - Interseção-sobre-União (IoU)

Equação 2 - Acurácia

Equação 3 - Precisão

Equação 4 - *Recall*

LISTA DE SÍMBOLOS E ABREVIATURAS

.gpkg	Geopackage
.pth	Arquivo do PyTorch
2D	Duas Dimensões
3D	Três Dimensões
AGI	<i>Artificial General Intelligence</i> (em português: Inteligência Artificial Geral)
AM	Aprendizado de Máquina
ANEEL	Agência Nacional de Energia Elétrica
AP	<i>Average Precision</i> (em português: Precisão Média)
API	<i>Application Programming Interface</i> (em português: Interface de programação de aplicações)
AR	<i>Average Recall</i> (em português: Recall Médio)
CNN	<i>Convolutional Neural Networks</i> (em português: Redes Neurais Convolucionais)
COCO	<i>Creating Common Objects in Context</i> (em português: Criando objetos comuns em contexto)
COP21	21ª Conferência das Partes
CPU	<i>Central Processing Unit</i> (em português: Unidade Central de Processamento)
EPSG	<i>European Petroleum Survey Group</i> (em português: Grupo de Estados Unidos)
EUA	Estados Unidos
FAIR	<i>Facebook Artificial Intelligence Research</i> (em português: Pesquisas em Inteligência Artificial do Facebook)
Fast R-CNN	<i>Fast Region-based Convolutional Neural Networks</i> (em português: Redes Neurais Convolucionais Rápidas baseada em regiões)
Faster R-CNN	<i>Faster Region-based Convolutional Neural Networks</i> (em português: Redes Neurais Convolucionais Mais Rápidas baseada em regiões)
FC	<i>Fully Connected Layer</i> (em português: Camada Totalmente Conectada)

FN	<i>False Negative</i> (em português: Falso Negativo)
FP	<i>False Positive</i> (em português: Falso Positivo)
FPN	<i>Feature Pyramid Networks</i> (em português: Redes de Pirâmides de Recursos)
FV	Fotovoltaico(a)
GAN	<i>Generative Adversarial Network</i> (em português: Redes Adversariais Generativas)
GB	<i>Gigabyte</i>
GDAL	<i>Geospatial Data Abstraction Library</i>
GeoJSON	<i>Geographic JavaScript Object Notation</i> (em português: Notação de Objetos Geográficos JavaScript)
GeoTIFF	<i>Georeferenced Tagged Image File Format</i>
GMP	<i>Google Maps Platform</i> (em português: Plataforma Google de Mapas)
GPU	<i>Graphics Processing Unit</i> (em português: Unidade de Processamento Gráfico)
GW	<i>Gigawatt</i>
HTTP	<i>Hypertext Transfer Protocol</i> (em português: Protocolo de transferência de hipertexto)
IA	Inteligência Artificial
IEF-MG	Instituto Estadual de Florestas de Minas Gerais
IoU	Interseção-sobre-União
IP	<i>Internet Protocol</i> (em português: protocolo de internet)
JPEG	<i>Joint Photographic Experts Group</i>
JSON	<i>JavaScript Object Notation</i> (em português: Notação de Objetos JavaScript)
LiDAR	<i>Light Detection and Ranging</i>
Mask R-CNN	<i>Mask Region-based Convolutional Neural Networks</i> (em português: Redes Neurais Convolucionais de Máscaras baseada em regiões)
MB	<i>Megabytes</i>
MSA	<i>Maps Static API</i> (em português: API de mapas estáticos)
OGR	<i>OpenGIS Simple Features Reference Implementation</i>

ONU	Organização das Nações Unidas
PNG	<i>Portable Network Graphics</i>
R-CNN	<i>Region-based Convolutional Neural Networks</i> (em português: Redes Neurais Convolucionais baseada em regiões)
RAM	<i>Random Access Memory</i> (em português: Memória de Acesso Aleatório)
ReLU	<i>Rectified Linear Unit</i> (em português: Unidade Linear Retificada)
RGB	<i>Red, Green, Blue</i> (em português: Vermelho, Verde, Azul)
RNA	Redes Neurais Artificiais
Roi	<i>Region Of Interest</i> (em português: regiões de interesse)
Roi Align	<i>Region Of Interest Align</i> (em português: alinhamento das regiões de interesse)
RPN	<i>Region Proposal Network</i> (em português: Rede de proposta de regiões)
SAS	Sistema de Aquecimento Solar
SEDE-MG	Subsecretaria de Ciência, Tecnologia e Inovação do Governo do Estado de Minas Gerais
SIG	Sistemas de Informações Geográficas
SVM	Máquina de Vetor de Suporte
tanh	Função Tangente Hiperbólica
TIFF	<i>Tagged Image File Format</i>
TN	<i>True Negative</i> (em português: Verdadeiro Negativo)
TP	<i>True Positive</i> (em português: Verdadeiro Positivo)
UFV	Centrais Geradoras Fotovoltaicas
UNFCCC	Convenção-Quadro das Nações Unidas sobre Mudança do Clima
VANT	Veículo Aéreo Não Tripulado
VC	Visão Computacional
WGS84	<i>World Geodetic System 84</i> (em português: Sistema Geodésico Mundial 84)

SUMÁRIO

1. INTRODUÇÃO	15
1.1 Contextualização	15
1.2 Justificativa	22
1.3 Estrutura	23
2. OBJETIVOS	24
2.1 Objetivo Geral	24
2.2 Objetivos Específicos	24
3. FUNDAMENTAÇÃO TEÓRICA	25
3.1 Imagens digitais	25
3.2 Visão Computacional	28
3.3 Inteligência Artificial e Redes Neurais	28
3.4 Segmentação de Instâncias	46
3.5 Aumento de dados	53
3.6 Sistemas fotovoltaicos e sistemas de aquecimento solar	54
4. MATERIAIS E MÉTODOS	59
4.1 Área de estudo	59
4.2 Hardware utilizado	60
4.3 Softwares utilizados	61
4.4 Dados utilizados	62
4.5 Metodologia	68
5. RESULTADOS	82
5.1 Preparação de dados	82
5.2 Treinamento e validação	83
5.3 Inferência	85
6. DISCUSSÕES	89
7. CONSIDERAÇÕES FINAIS	98
REFERÊNCIAS BIBLIOGRÁFICAS	101

1. INTRODUÇÃO

1.1 Contextualização

O mapeamento de informações espaciais é um processo fundamental e de grande importância nos estudos ambientais feitos por órgãos públicos e privados em todo o mundo, de maneira a apoiar as tomadas de decisões informadas pelas partes interessadas. Neste aspecto, é indispensável que estas informações sejam captadas com rapidez e precisão, considerando que as alterações ambientais no mundo vem ocorrendo de maneira cada vez mais veloz, especialmente nas áreas urbanas (MOHANRAJAN, 2020). Portanto, as informações contidas em imagens orbitais (imagens de satélite) ou em imagens aéreas provenientes de processos fotogramétricos (feitas por veículos aéreos não tripulados (VANTs) ou aerofotogrametria convencional) se mostram como uma das melhores formas de monitorar estas alterações em grandes escalas, tanto temporais quanto espaciais, e, dispô-las em mapas temáticos que possam auxiliar os tomadores de decisão e analistas no entendimento da realidade e dos fenômenos que ocorrem em suas áreas. Exemplos recentes compreendem os trabalhos de Mello et al. (2017), Silva (2020) e Nero et al. (2020) e Vargas (2021).

Estes mapas temáticos possuem informações como vegetação, uso e cobertura do solo, manchas urbanas, edificações e etc., elementos estes que podem ser extraídos a partir de classificações de imagens orbitais (USMAN, 2013; MOHANRAJAN, 2020), ortofotos ou dados de *Light Detection And Ranging* (LiDAR), a exemplo do que foi realizado por Faria (2017).

Ainda que estas imagens orbitais e aéreas estejam amplamente disponíveis, inclusive em bancos de dados abertos ao público, o processamento e extração de características e informações das mesmas não acontece no mesmo passo que sua disponibilização. Um exemplo desta dificuldade é o desafio 16, lançado pelo Instituto Estadual de Florestas de Minas Gerais (IEF-MG) para o programa de aceleração de startups SEED, realizado pela Subsecretaria de Ciência, Tecnologia e Inovação do governo do estado de Minas Gerais (SEDE-MG). Neste desafio, o IEF ressalta como parte das dificuldades que o órgão enfrenta, o tempo gasto por seus técnicos para

identificar áreas de desmatamento de vegetação nativa em todo o estado, já que esta análise é feita utilizando as imagens do satélite Sentinel-2 em inspeção visual por seu corpo técnico. No edital¹ onde descreve este problema, o IEF afirma que atualmente um único técnico leva um dia inteiro de trabalho para analisar uma área de 12 km² (SEED, 2021) em um Estado que possui uma área total de 586528 km². Portanto, para executar a varredura em todo o território estadual, em um único dia de trabalho seriam necessários aproximadamente 48.877 técnicos trabalhando, simultaneamente. Cabe ressaltar que este é um monitoramento que é de natureza diária e contínua do órgão.

Apesar das técnicas de classificação de imagens serem bastante utilizadas para interpretação da realidade, sua aplicação em áreas urbanas é limitada muito por conta da heterogeneidade dos dados de superfície, sendo os objetos com respostas espectrais muito distintas que geram confusão nos processos de classificação (FARIA, 2017).

Neste sentido, a área de visão computacional (VC) tem ganhado cada vez mais relevância em aplicações de mapeamento, planejamento urbano, gerenciamento das cidades, mobilidade e outras. Junto a isso, novas técnicas de aprendizado profundo (ramo da inteligência artificial) têm trazido melhorias relevantes para o estado da arte da VC, principalmente, na classificação de imagens orbitais e aéreas (ABDURAHMANOV, 2020) com técnicas de redes neurais convolucionais de aprendizado profundo.

As análises e classificações automatizadas de imagens têm sido um problema em várias áreas do conhecimento e a comunidade do sensoriamento remoto têm demonstrado particular interesse nas aplicações destas técnicas nos problemas de classificação de uso e ocupação do solo (BHUIYAN, 2020). Este interesse advém das possibilidades de automatização destas classificações de formas que não eram possíveis antes. Isso devido à inexistência destas técnicas até poucos anos atrás ou mesmo pelo poder computacional que a execução destas técnicas demanda. Ainda, tem-se que até recentemente, tais possibilidades de automação não eram ofertadas

¹ O edital do processo seletivo do "Seed 2021 - Edição Especial" está disponível em <<http://seed.mg.gov.br/wp-content/uploads/2016/11/Edital-consolidado-07042021.pdf>>

em custos acessíveis, permanecendo, portanto, fora do alcance dos pesquisadores do tema.

Para treinar as redes neurais de aprendizado profundo na área de VC, é necessário realizar extensas rotulagens (anotações de dados) para cada objeto que se deseja identificar em imagens. Estas anotações servem como valores de referência para o aprendizado da rede neural. Quanto maior a qualidade e o número de anotações, melhores e mais acurados ficam as classificações das redes.

Em especial, a segmentação de instâncias (uma das várias técnicas de classificação de imagens), requer rotulagens ainda mais complexas e laboriosas, e, para bases de dados espaciais, ainda não há muitos dados públicos disponíveis (CARVALHO et al., 2020). A maioria dos dados públicos disponíveis para a construção destas redes são para fotografias de paisagens, fotos de cenas urbanas, imagens médicas e outras. Além dos dados de treinamento, também não há ambientes de desenvolvimento adequados e robustos para uso de arquivos vetoriais - amplamente usado como padrão na indústria dos Sistemas de Informações Geográficas (SIG) - como anotações de treino destas redes. Portanto, a adaptação das ferramentas de aprendizado profundo já existentes em outras áreas, para ambientes SIG é um desafio e uma oportunidade.

Conforme exposto, existe um problema de mapeamentos contínuos, ágeis e escaláveis e o setor energético é uma destas áreas em que a geração de informações de qualidade são importantes para os processos decisórios em escalas locais, regionais e globais.

As sociedades modernas usam energia em processos industriais, transporte, agricultura, aquecimento, alimentação, trabalho e em quase todo tipo de atividade. Considerando-se que a população mundial continuará crescendo pelas próximas décadas com estimativa da Organização das Nações Unidas de uma população de 10,9 bilhões no ano de 2100 (ONU, 2019), além de que à medida em que a população mundial se desenvolve e enriquece, passa a utilizar mais recursos energéticos. Assim, infere-se que o consumo e a demanda por energia siga esta tendência e seja cada vez maior (SOYTAS et al., 2003). Historicamente, a matriz energética mundial tem sido composta por combustíveis fósseis (carvão, petróleo e gás natural), e, atualmente, ainda provê 84% da energia consumida mundialmente

(BP, 2019). Esta composição da matriz energética traz impactos consideráveis ao ambiente, já que os combustíveis fósseis produzem dióxido de carbono (CO₂) e outros gases de efeito estufa, fator determinante nas mudanças climáticas globais.

Os impactos ambientais deste consumo crescente têm se tornado o centro do debate global. Em dezembro de 2015 aconteceu a 21ª Conferência das Partes (COP21) da Convenção-Quadro das Nações Unidas sobre Mudança do Clima (UNFCCC). Durante a COP21 foi realizado o Acordo de Paris, tratado que foi um marco no esforço internacional no combate às mudanças climáticas causadas pelo homem, devido às emissões de gases de efeito estufa em suas atividades. O acordo global obteve a assinatura de 194 nações e da União Europeia, incluindo os principais emissores de gases de efeito estufa - China, Estados Unidos e Índia (ONU, 2016) - e representou um sério movimento em direção à tendência da descarbonização das economias mundiais.

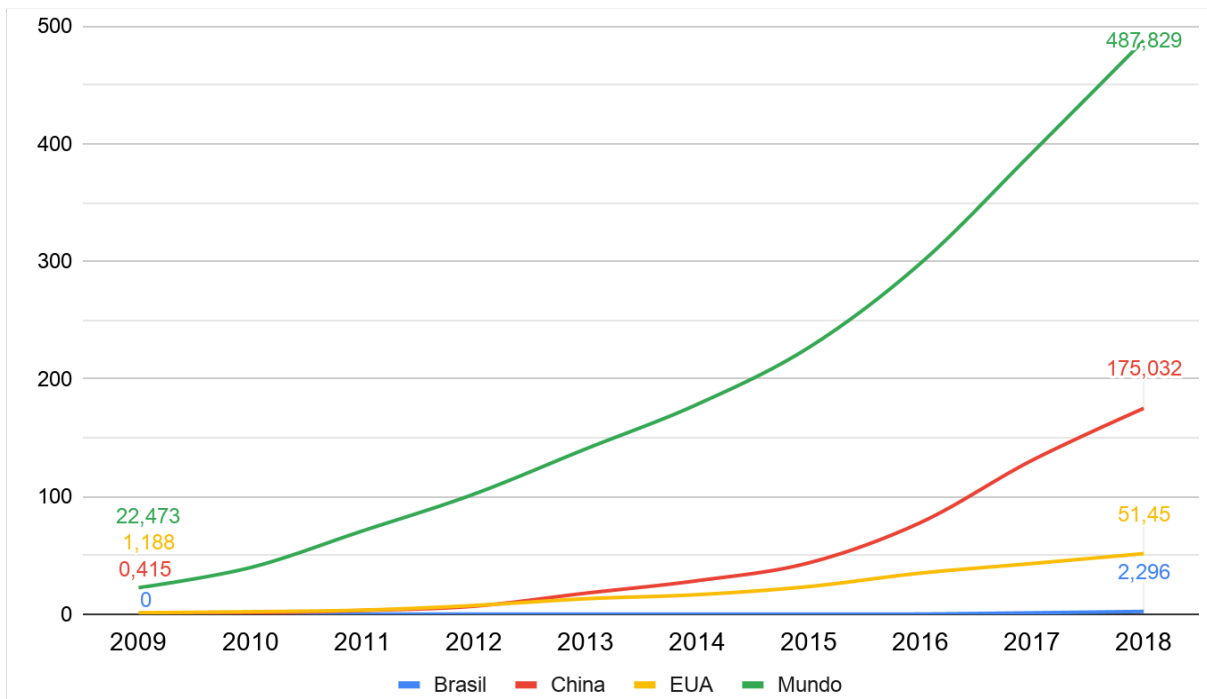
Reconhecendo o potencial de mercado que as tecnologias limpas e de baixo carbono possuem, muitos países já vêm impulsionando o pioneirismo e estabelecendo a vanguarda destas indústrias, com destaque para a produção de energia solar e eólica (KERN et al., 2016). A China, particularmente, deixou de ser um agente passivo na economia das energias limpas para liderar este mercado globalmente (KERN et al., 2016). Assim, no ano de 2006 este país aprovou a “Lei de Energia Renovável”, o que acelerou o processo de desenvolvimento de tecnologias de geração solar e eólica, tanto nacionalmente como globalmente (TU et al., 2019). Mais recentemente, com seu 13º Plano Quinquenal, o governo chinês especificou que pretende desenvolver e avançar as principais tecnologias como por meio da geração solar fotovoltaica de ponta, geração eólica de alta eficiência, *smart-grids*² e novos tipos de dispositivos de armazenamento de energia (CHINA, 2016, p. 67).

Em termos globais, essa mudança em direção à utilização das energias renováveis tem sido significativa e, a tendência é que isto continue. Os dados de crescimento deste setor dos últimos 10 anos, demonstram que a perspectiva para os próximos anos é que essa tendência crescente permaneça acontecendo. Conforme os dados apresentados por BP (2019), representados no gráfico da Figura 1, a

² Sistema de energia inteligente que regula o fluxo de eletricidade e informações bidirecionalmente, tanto entre unidades produtoras quanto consumidores, para controlar a atividade da rede (EERE, 2018).

capacidade cumulativa instalada de geração solar fotovoltaica no mundo saltou de 22,47 GW de capacidade instalada no ano de 2009 para 487,82 GW no ano de 2018, um aumento de 2070,73% em um período de 10 anos - crescimento de 32,58% ao ano, em média.

Figura 1 - Capacidade cumulativa instalada fotovoltaica (em GW, ao final do ano)

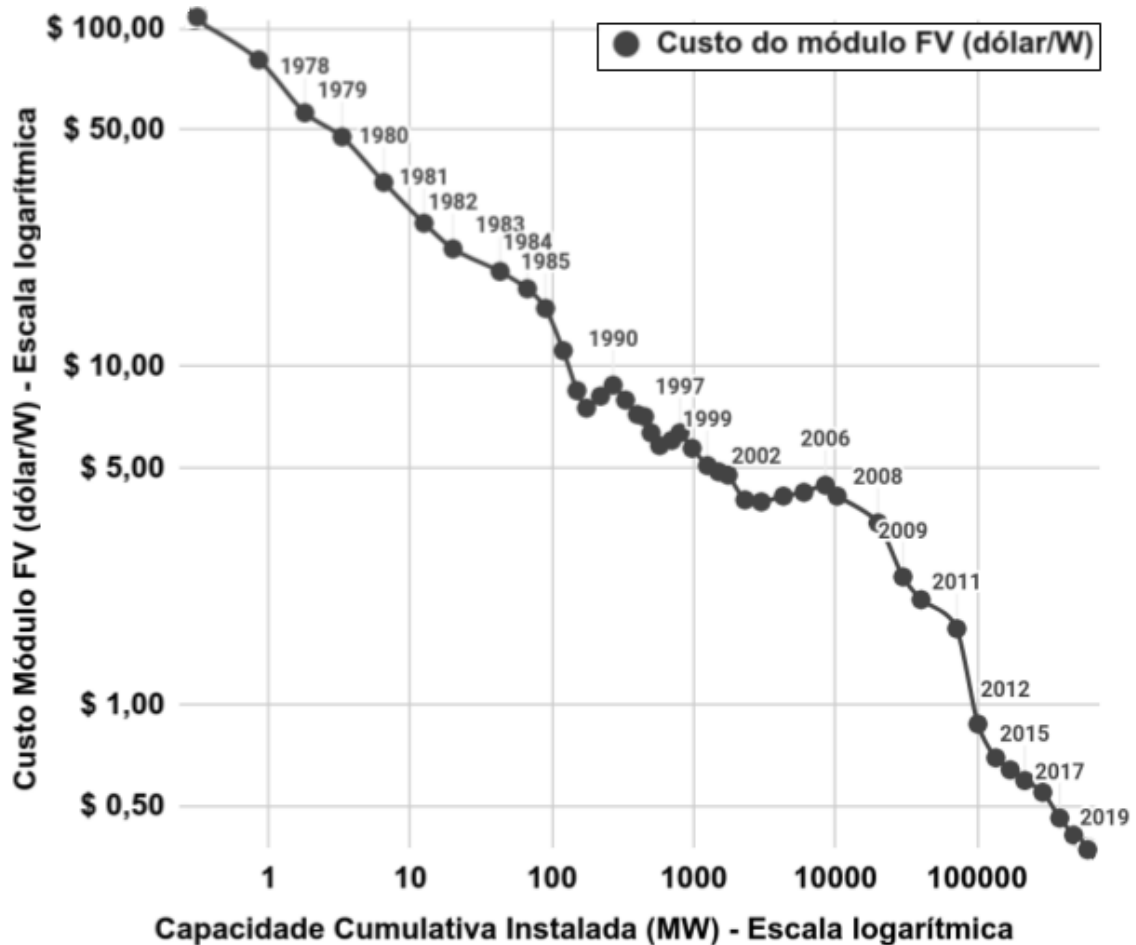


Fonte: British Petroleum (2019)

Neste mesmo período o Brasil saiu de 0 GW no ano de 2009 para 2,29 GW em 2018, capacidade equivalente a 1% da capacidade instalada da China, que possuía 175,03 GW de capacidade cumulativa instalada no final de 2018. Os dados comparativos do Brasil em relação ao mundo demonstram que o país ainda tem um mercado enorme a explorar na área de geração de energia solar. Isso porque o país tem a maior parte de seu território em baixas latitudes, próximas à linha do equador e isso torna toda sua área potencialmente atrativa para geração fotovoltaica.

Essa aceleração na adoção da geração fotovoltaica se explica pelas reduções exponenciais de preços na produção dos módulos fotovoltaicos (Figura 2) mas também como consequência da mudança dos investimentos de projetos massivos de grande escala de produção de energia solar (HAUKKALA, 2015).

Figura 2 - Custo de produção/watt de módulos FV (em dólares) comparado à capacidade cumulativa instalada (em MW)



Fonte: IRENA (2020)

Os sistemas fotovoltaicos convertem luz solar em eletricidade e possuem algumas vantagens frente a outros sistemas, uma vez que é um sistema modulado, pode ser escalado para grandes áreas mas também para áreas muito pequenas como telhados residenciais (IEA, 2018). Outra vantagem dos sistemas fotovoltaicos é que podendo estar perto do consumidor final, permite fácil manutenção e tem um menor custo, principalmente, relativos à distribuição e transmissão com maior retorno financeiro do que outras opções de geração de energia (XIN-GANG, 2019).

Em termos da eficiência e custo na produção de Centrais Geradoras Fotovoltaicas (UFV)³ cita-se a “Lei de Swanson”, uma observação que corresponde

³ No curso do texto, usou-se o termo “Central Geradora Fotovoltaica” (UFV) como sinônimo de “sistemas fotovoltaicos” por se tratar da terminologia utilizada pela Resolução Normativa nº 876 de 10 de março de 2020 da Agência Nacional de Energia Elétrica (ANEEL, 2020).

à curva de aprendizagem elaborada por Swanson (2006) em que o autor afirma que os módulos fotovoltaicos têm seus preços reduzidos em 20% a cada vez que a capacidade cumulativa instalada dobra (Figura 2) e que esta tecnologia alcançaria a paridade de custo por Watt produzido com outras formas de produção energética, como o gás natural em alguns anos após sua publicação.

Segundo Mitidieri (2017), diferente de fontes que dependem de ocorrências naturais que definem sua localização (como, por exemplo, as grandes quedas d'água que determinam as posições das hidrelétricas), a energia solar é um recurso energético que pode ser aproveitado em todos os pontos do planeta, sendo posicionado de acordo com as condições meteorológicas, climáticas e latitude das localidades instaladas. Isto permite uma grande flexibilidade desta modalidade de produção, que pode ser desenvolvida tanto em grandes usinas centralizadas, quanto próximas dos pontos de consumo, na forma de pequenas plantas distribuídas.

Para tanto, pode-se realizar estudos de modo a definir as áreas ou locais mais adequados para a instalação de UFV pela utilização de imagens de alta resolução, obtidas por VANTs e com a geração das áreas de melhor adequabilidade e potencialidade de instalação, tal como apresentado em Nero et al. (2020).

Neste sentido, Yan et al. (2019) demonstraram que com o atual estado da arte dos módulos fotovoltaicos, sem subsídios governamentais, esta modalidade de geração já alcançou esta paridade na China, se comparados os preços da geração fotovoltaica distribuída e os preços da energia elétrica fornecidos pela rede.

Partain e Fraasand (2010) afirmaram que se o crescimento das instalações solares continuarem a dobrar a cada dois anos, este tipo de geração alcançaria números suficientes para cobrir a produção mundial de eletricidade no ano de 2030. Naam (2011) afirma que dada a tendência do barateamento dos custos da geração fotovoltaica, por volta do ano de 2030, a energia elétrica oriunda do carvão, por exemplo, terá o dobro do custo da energia de fontes solares.

Com os preços dos módulos fotovoltaicos diminuindo exponencialmente, a tendência é que o incentivo econômico para a adoção desta tecnologia aumente da mesma maneira. Um aspecto relevante no acesso e consumo de energia elétrica é o custo da distribuição/transmissão e seu peso na composição do preço repassado ao consumidor final. Nos Estados Unidos (EUA), segundo o órgão responsável pelas

informações de energia do país (EIA, 2019), a distribuição e transmissão de energia representam 40% do preço da energia, enquanto no Brasil corresponde a aproximadamente 20% (ANEEL, 2017). Sabe-se que os telhados são os melhores locais para a instalação de painéis FV em edificações (KOUHESTANI, 2018) e estes módulos em telhados não demandam grande estrutura de distribuição e transmissão e, portanto, podem reduzir consideravelmente os custos com a geração de energia, além de possibilitarem uma menor demanda na rede elétrica.

Com a diminuição nos preços de instalação e dos equipamentos, o investimento em tecnologias que produzem energia solar tem sido cada vez mais atraente para o setor de produção energética.

Hoje mais de 50% da população mundial vive em ambientes urbanos e espera-se que este número aumente para 70% no ano de 2050. Em dez anos, estima-se que 75% do consumo energético global virá das cidades, e, portanto, iniciativas que incluam novas abordagens como UFV em ambientes urbanos são essenciais (HUANG, 2019). Comparada com outras alternativas, a geração fotovoltaica em telhados possui maior aceitação, em parte, por não apresentar disputas no uso da terra (MAINZER, 2018) e utilizar um espaço ocioso das construções urbanas.

1.2 Justificativa

A compreensão do fenômeno da presença e expansão das instalações de UFV, em especial em áreas urbanas, é um importante indicador da evolução de segurança energética das cidades, e, conforme demonstrado na seção 1.1, este fenômeno evolui exponencialmente e sua detecção a partir de imagens aéreas constitui uma importante forma de entendê-lo.

As redes neurais profundas podem ser utilizadas como uma solução em mapeamentos temáticos automatizados e nesta pesquisa isto é demonstrado utilizando o exemplo de UFV como alvos a serem detectados. O desenvolvimento de aplicações que usem estas técnicas pode melhorar a precisão, aumentar a escala e poupar os recursos das instituições que fazem uso destas ferramentas frequentemente.

A possibilidade da automatização destas detecções em áreas extensas, em volumes de trabalho que não seriam possíveis para humanos executarem manualmente em termos produtivos é um dos pontos que este trabalho se faz importante.

1.3 Estrutura

Este trabalho está estruturado da seguinte forma:

O capítulo 1 apresenta uma breve contextualização do problema da classificação e segmentação automatizada de imagens. Apresenta também o cenário energético mundial e a importância da detecção de UFV em imagens urbanas.

O capítulo 2 apresenta os objetivos gerais e específicos desta pesquisa.

O capítulo 3 traz o arcabouço da literatura científica usada no âmbito deste trabalho, expondo aspectos conceituais das técnicas usadas.

Já o capítulo 4 corresponde à metodologia desenvolvida nesta pesquisa, descrevendo as etapas de treinamento, validação e inferência do modelo de segmentação de imagens.

No capítulo 5 são expostos os resultados encontrados na execução da metodologia proposta e o capítulo 6 discute-os.

Por fim, no capítulo 7 o autor expõe algumas considerações sobre o trabalho realizado e perspectivas de futuros trabalhos.

2. OBJETIVOS

2.1 Objetivo Geral

O objetivo principal deste trabalho foi a criação e execução de uma metodologia de segmentação de imagens aéreas de alta resolução (7 cm/pixel) em toda a extensão do município de Belo Horizonte/MG buscando duas classes-alvo: sistemas fotovoltaicos (UFV) e sistemas de aquecimento solar (SAS).

2.2 Objetivos Específicos

- Criar um método replicável de segmentação de imagens para outros alvos além de UFV/SAS;
- Analisar e discutir a precisão de modelos de aprendizado profundo, principalmente em feições de classes distintas, mas que possuam características (geometria, textura, distribuição espacial) semelhantes como UFV e SAS;
- Expor a atual situação das instalações de UFV em todas edificações do município de Belo Horizonte e gerar uma base de dados contendo os polígonos segmentados de UFV e SAS.
- Obter como produto do método de segmentação arquivos vetoriais georreferenciados em formatos do padrão da indústria do geoprocessamento (*geopackage* e *shapefile*) para facilitar a manipulação das informações em ambientes SIG;
- Publicar os dados obtidos em uma plataforma de mapas na internet.

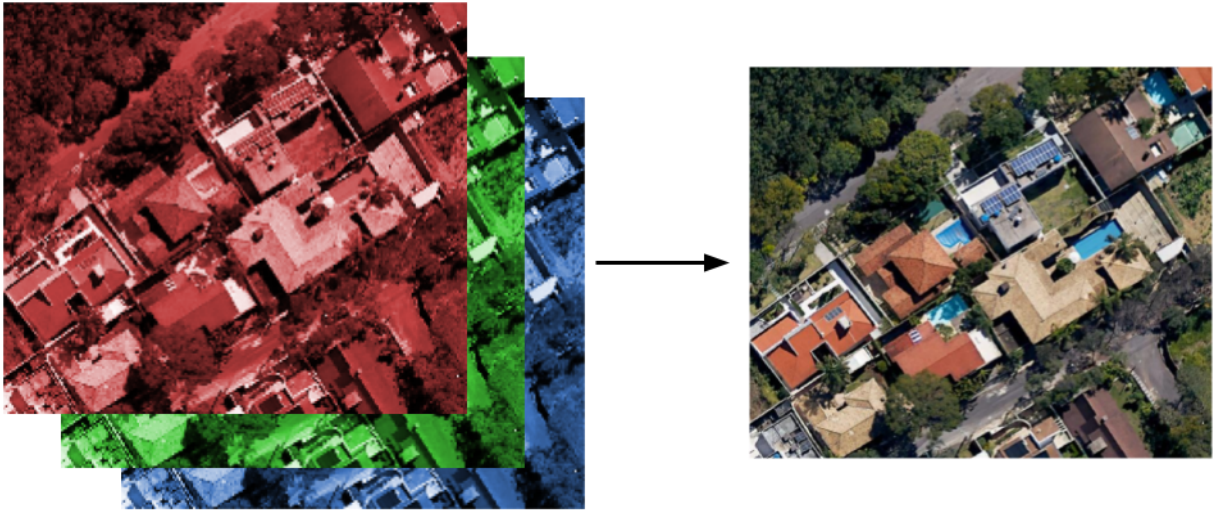
3. FUNDAMENTAÇÃO TEÓRICA

3.1 Imagens digitais

Uma imagem digital é a representação discreta de dados espaciais e de intensidade que dão formas e cores, respectivamente, a estes dados. A representação discreta é dada pela menor unidade das imagens, chamada de “pixel”. As imagens digitais podem ter diferentes disposições, as mais comumente utilizadas em ambientes digitais são as bandas visíveis do espectro eletromagnético com imagens de formato RGB (do inglês *red*, *green* e *blue* - vermelha, verde e azul), usadas em fotografias, *displays* de equipamentos eletrônicos como televisores, celulares e outros.

Imagens RGB são dados matriciais de duas dimensões (2D) com três planos distintos, que são planos equivalentes à respostas espectrais da luz destas três bandas, respectivamente (SOLOMON e BRECKON, 2011) e que possuem valores de pixels representando a intensidade de luz em escala variável de 0 a 255, ao considerar-se resolução radiométrica de 8 *bits*. As cores primárias, em cada um dos planos, são então misturadas para que formem as cores que vemos em imagens. Existem outras formas de representações de imagens digitais, além das imagens RGB, como imagens binárias (apenas um plano 2D com valores de pixels variando entre 0 ou 1), imagens da banda pancromática, segundo a qual há fusionamento das três bandas do visível em uma banda, mas que não estão no escopo deste trabalho e portanto não serão aqui descritas. Na literatura utilizam-se termos variados para referir a estes planos bidimensionais, por vezes são chamados de “bandas”, “canais” ou “espectros”. A Figura 3 demonstra uma representação esquemática de uma imagem digital no formato RGB e sua composição por três bandas.

Figura 3 - Representação esquemática de uma imagem em formato RGB



Fonte: Adaptado de Google Earth (2018)

Além de pixels e bandas, a resolução de uma imagem digital define o poder representativo da informação contida na imagem. Existem diferentes tipos de resoluções para as imagens digitais. Estas resoluções podem ser resolução espacial (número de linhas e colunas) que define o número de pixels da imagem, resolução temporal (número de imagens em determinado tempo t) (SOLOMON e BRECKON, 2011).

Ortofotos digitais são fotografias aéreas que recebem transformações de correção em perspectiva ortogonal para que pareçam que foram verticalmente tomadas a partir de uma distância infinita (ACSM, 1994).

A resolução espacial é dada pelo tamanho que um único pixel representa na realidade, por exemplo, em uma imagem com resolução espacial de 1 metro, significa que cada pixel destas imagens representam uma área de 1x1 metro (1 m^2) no terreno (SATSUMMIT, 2017). A Figura 4 mostra imagens de uma ortofoto com resolução espacial de 7 cm/pixel e a outra imagem do mesmo local com resolução espacial de 1 m/pixel.

Figura 4 - Ortofotos RGB com resolução espacial de 7 cm/pixel (à esquerda) e 1m/pixel (à direita)



Fonte: Google Earth (2018)⁴

A resolução temporal para imagens de satélites ou ortofotos digitais, refere-se à qual frequência de tempo que uma mesma área tem dados disponibilizados. Comumente a literatura da área de sensoriamento remoto também chama a resolução temporal de “tempo de revisita” (SATSUMMIT, 2017).

A resolução espectral que representa o número de bandas de uma imagem é outro tipo de resolução de imagens aplicadas às imagens de satélites. Satélites modernos, em geral, coletam dados dos espectros para além do que é visível ao olho humano como os espectros do infravermelho, ultravioleta e também oriundos de sensores de radar e *lasers*. Estas bandas são importantes porque trazem informações para além daquelas que são visíveis a olho nu e aumentam a possibilidade das análises (SATSUMMIT, 2017).

Por fim, é possível citar também a resolução radiométrica de uma imagem, que diz respeito à amplitude de dados que um único pixel pode possuir. Uma imagem binária, por exemplo, tem resolução de 1 bit, ou seja, a informação varia entre 0 ou 1 (2 valores diferentes) por pixel. Uma imagem RGB, possui uma resolução de 2^8 bits/pixel/banda, ou seja, a informação pode variar de 0 a 255 (256 valores diferentes) em cada pixel de uma de suas bandas.

⁴ Imagem original (esq.) redimensionada para 1m/pixel (dir.) pelo autor para fins didáticos.

3.2 Visão Computacional

A VC é o campo que estuda como computadores podem extrair descrições explícitas e com significados de imagens digitais (BALLARD et al., 1982). Portanto, a VC busca desenvolver métodos automáticos e algoritmos que possam executar essa extração para servir aos propósitos de seus estudiosos (AMERIKANOS, 2020). A VC tem sido a vanguarda do aprendizado de máquina. Estes algoritmos utilizam redes neurais artificiais (RNA) para executar tarefas que demandam um alto volume de trabalho, em níveis que humanos não são capazes de executar (LIVINGSTONE, 2019).

Para que se extraíam dados geoespaciais e, posteriormente, utilizá-los em SIG, geralmente faz-se uso de fotografias aéreas, imagens orbitais e para criar ou atualizar bancos de dados geoespaciais (DOUCETTE et al., 2004). Dentro dos sistemas de VC que utilizam redes neurais convolucionais, existem cinco principais tarefas: classificação de imagem, detecção de objetos, segmentação semântica, segmentação de instâncias e segmentação panóptica.

3.3 Inteligência Artificial e Redes Neurais

Inteligência artificial (IA) é a definição mais ampla da área do conhecimento que trata de sistemas que simulam, de certo modo, a inteligência humana e que são usados para otimizar as tarefas que têm sido feitas por seres humanos historicamente, como tomada de decisão, classificação de elementos em seus respectivos conjuntos, reconhecimento facial, tradução de idiomas e outros (KAVLAKOGLU, 2020).

Dentro dessa área do conhecimento, Goertzel (2019) separa a IA em duas categorias, a Inteligência Artificial Geral (AGI, do inglês *Artificial General Intelligence*) que é o sistema que pode aprender a desempenhar diferentes tarefas e habilidades ao lidar com situações em contextos completamente diferentes daqueles em que foi programada e treinada. A outra categoria é a Inteligência Artificial Especializada (em inglês *Narrow AI*), que é o sistema que possui inteligência aplicada a um domínio

específico para o qual foi treinado (GOERTZEL, 2019) como jogar xadrez, prever condições climáticas ou classificar imagens.

Por muito tempo, o campo da IA esteve vinculado às ficções científicas, mas nas últimas décadas isto tem mudado rapidamente e aplicações de IA estão cada vez mais integradas ao cotidiano no mundo inteiro. Isto é explicado pelo fato da disponibilidade de quantidades de dados e capacidade de processamento computacional cada vez maiores (KAVLAKOGLU, 2020).

Dentro do campo da IA, existem subconjuntos que desempenham tarefas diferentes, alguns deles são o Processamento de Linguagem Natural, Sistemas Especializados, Aprendizado de Máquina e outros.

Aprendizado de Máquina (AM), ou em inglês *Machine Learning*, é uma ramificação da IA que busca extrair padrões inteligíveis e executar tarefas da mesma forma que um humano faria, manualmente.

Amerikanos (2020) aponta que geralmente os algoritmos de AM estão dentro de quatro principais categorias, são elas: aprendizado supervisionado, aprendizado não-supervisionado, aprendizado auto-supervisionado e aprendizado por reforço. O mesmo autor aponta que o aprendizado supervisionado, é a categoria mais usada das quatro listadas e baseia-se no uso de dados de treinamento, com anotações manuais feitas por humanos para que o modelo aprenda os padrões destas anotações de referência e gere modelos de predição, aplicados à dados os quais o modelo não teve contato durante o treinamento. O aprendizado supervisionado é utilizado principalmente em tarefas de classificação, regressão e segmentação. É importante notar que a maior parte das aplicações de IA na área de processamento digital de imagens utiliza a classificação supervisionada (O'SHEA et al. 2015).

Entre os algoritmos no campo do AM estão a regressão linear, regressão logística, máquina de vetor de suporte (SVM), redes neurais artificiais e outros.

A modelagem preditiva é a abordagem em que, a partir de dados históricos prévios de algum fenômeno, se cria um modelo para previsões com dados posteriores em que não se tem a resposta. Portanto, pode ser descrita como um mapeamento de uma função de aproximação (f) com variáveis de entrada (X) para variáveis resposta (y) (BROWNLEE, 2017).

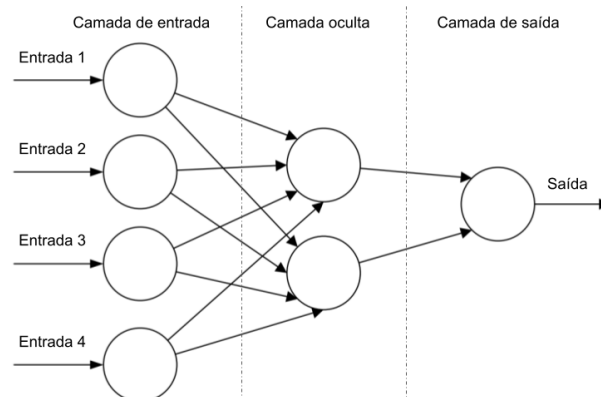
Geralmente, a literatura divide os tipos de funções de aproximação em duas categorias: tarefas de regressão e tarefas de classificação. As tarefas de regressão são aquelas em que se prediz uma variável quantitativa contínua a partir de dados de entrada de mesma natureza, por exemplo a predição da altura de uma pessoa a partir da variável de entrada peso. As tarefas de classificação são aquelas em que se prediz uma classe discreta a partir das variáveis de entrada (BROWNLEE, 2017). Um exemplo de tarefa de classificação é dado por um modelo que recebe uma imagem de um animal como dado de entrada e realiza a classificação deste animal em alguma classe, como “cachorro” ou “gato”.

Em geral, os modelos de classificação estabelecem além da classe predita, um valor de probabilidade numérico (em termos percentuais, de 0 a 100%) daquele dado de entrada pertencer de fato àquela classe que o modelo apontou.

As RNA foram uma abordagem proposta por McCulloch et al. (1944), como uma maneira de ensinar às máquinas a aprenderem a desempenhar tarefas de modelagens preditivas, tanto de regressão quanto de classificação. Este aprendizado acontece com o fornecimento pelos operadores das RNA de dados de treinamento para que estas redes analisem os exemplos dos padrões que se deseja reconhecer (HARDESTY, 2017).

A Figura 5 mostra um esquema de uma rede neural simples do tipo *feedforward* onde há uma camada de entrada, uma camada oculta e uma camada de saída, sendo esta a principal estrutura praticada por trabalhos de RNA na literatura (O'SHEA et al. 2015).

Figura 5 - Exemplo esquemático de uma rede neural



Fonte: Adaptado de O'SHEA et al. (2015)

Na estrutura apontada pela Figura 5, carregam-se os dados na camada de entrada na forma de vetores multidimensionais (no caso de imagens, vetores com dimensões de *altura x largura x bandas*) que têm seus valores distribuídos para a camada escondida. A camada oculta por sua vez tomará decisões a partir dos valores da camada de entrada e avaliará como uma mudança estocástica em si mesma (seus valores de pesos e vieses) prejudica ou melhora o resultado final de classificação, a partir de um valor de referência previamente anotado, e a este processo se dá o nome de aprendizagem (O'SHEA et al. 2015). Redes neurais que possuem três ou mais camadas ocultas são chamadas de redes neurais profundas.

Um problema importante a ser entendido ao desenvolver-se modelos no aprendizado de máquina é o conceito de sobreajuste (no inglês, *overfitting*). Quando um modelo de classificação é treinado, o operador do modelo objetiva que este seja generalizável, ou seja, buscando resposta para a pergunta “o quão bem este modelo lida com exemplos específicos em que nunca teve contato durante a etapa de treinamento?” (BROWNLEE, 2016). Quando um modelo tem uma má performance quando aplicado aos dados de teste, mas também aos dados de treinamento, ele sofre um problema de sub-ajuste. Quando um modelo tem uma má performance nos dados de teste mas uma boa performance nos dados de treino, a este problema se dá o nome de sobreajuste, ou seja, ele só responde bem àqueles dados os quais já teve contato.

Na área de classificação de imagens, existem algumas técnicas para melhorar a capacidade de generalização dos modelos, como técnicas de regularização e aumento de dados. A seção 3.5 detalha conceitualmente a técnica de aumento de dados e como ela diminui os vieses e aumenta o poder de generalização dos modelos de classificação.

As Redes Neurais Convolucionais (CNN, da sigla em inglês *Convolutional Neural Networks*), são algoritmos adaptados das RNA tradicionais para processamento de imagens digitais que aplicam filtros nas mesmas, de modo que possam capturar as dependências espaciais destas imagens e extrair as principais características de baixo-nível (cores e bordas, por exemplo), reduzindo o número de parâmetros e reutilizando os pesos da RNA (SAHA, 2018).

As RNA são consideradas, em geral, modelos “caixa-preta”, ou seja, é difícil para seus operadores explicarem as inter-relações entre as variáveis de entrada com os resultados de suas predições. As CNN, ao contrário, uma vez que são feitas voltadas à classificação de imagens digitais, é possível apontar exatamente onde os neurônios de ativação estão sendo engatilhados ao longo da rede, já que seus mapas de ativação⁵ podem ser visualizados e analisados por seus operadores (BROWNLEE, 2017).

As CNN foram inicialmente inspiradas por uma pesquisa de Hubel et al. (1959), sobre o sistema de visão de mamíferos (gatos e macacos) e concluíram que alguns neurônios específicos do sistema de visão de seus cérebros são ativados em resposta a padrões de bordas em seu campo visual muito mais do que à outros padrões (PEŠEK, 2018).

Isto posto, as CNN emulam estas ativações de neurônios sensíveis às bordas no campo de visão, e, considerando que imagens possuem um certo padrão de ordenamento dos pixels (ou seja, regiões que possuem determinadas feições e autocorrelação espacial), permitindo que os neurônios das CNN sejam ativados somente nestas regiões das imagens. Este é um fato importante, pois permite que nem todas as camadas de neurônios de uma rede neural estejam ligadas entre si, e, portanto, seja possível criar redes menos complexas e computacionalmente mais leves do que seriam redes com camadas totalmente conectadas (FC, do inglês *fully connected layer*) (PEŠEK, 2018).

A arquitetura das CNN é composta de três tipos de camadas, são elas as camadas convolucionais, camadas de *pooling* e camadas FC, exemplificadas na Figura 6. A cada uma dessas camadas, o modelador estabelece seus hiperparâmetros que devem ser definidos de antemão no modelo, ou seja, valores que não são modificados durante o treinamento da rede. Estes hiperparâmetros são o número e dimensão dos filtros das camadas convolucionais, tamanho da janela e *stride* (valor de salto em pixels a cada iteração da convolução) das camadas de *pooling* e número de neurônios das camadas FC. É possível alternar a aplicação de

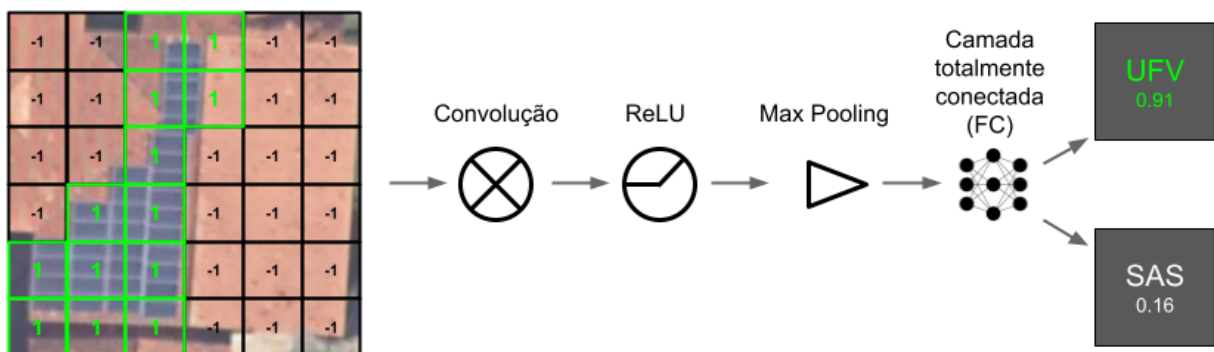
⁵ Mapas de ativação são mapas de características extraídas por uma CNN de uma imagem de entrada. Na literatura, o termo em inglês usado é *feature maps*, mas em português podem ser encontrados termos correlatos como “mapas de características” ou “mapas de recursos”.

cada uma dessas camadas para obter saídas mais refinadas com menos processamento computacional e melhores resultados.

As camadas convolucionais são filtros⁶ deslizantes. Os filtros são aplicados nas imagens de maneira a extrair suas principais características de alto/baixo nível. Essa extração resulta no que a literatura chama de mapas de ativação (do inglês *features map*) e essas redes de extração de características⁷ compõem o que se chama de *backbone* de uma CNN.

A Figura 6 ilustra um exemplo de uma CNN de classificação de imagem (matriz binária de pixels representando a letra “X” - resolução 6x6x1) com as camadas da rede sequenciais gerando as probabilidades desta imagem corresponder a uma classe “X” ou “O”, passando por etapas alternadas de convoluções, ReLU, *pooling* e FC. Cada uma destas etapas está detalhada abaixo.

Figura 6 - Camadas empilhadas de aplicações de convoluções, normalizações e *pooling* dos dados de entrada, contendo a probabilidade de classificação



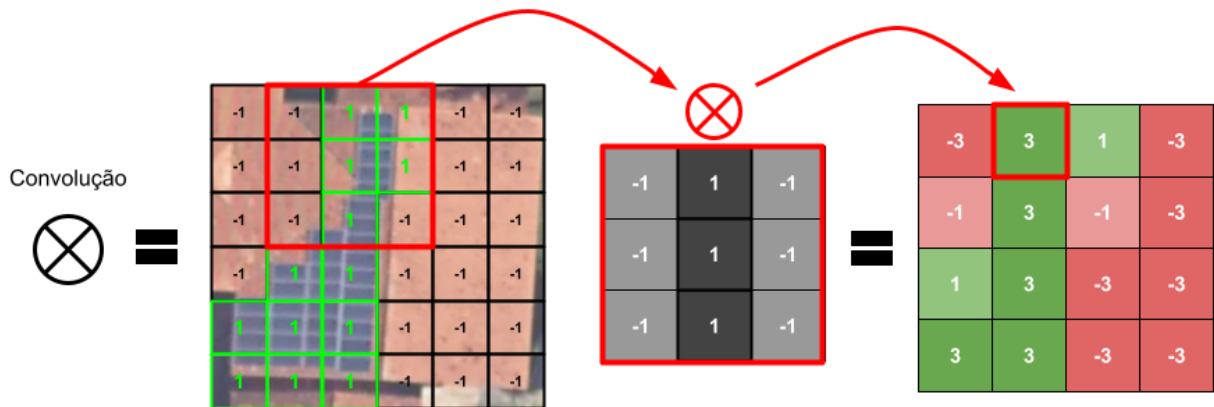
Fonte: Adaptado de ROHRER (2018).

A Figura 7 representa um desenho esquemático da primeira etapa de convolução da rede. Esta etapa de convolução aplica três convoluções de filtros de 3x3x1 aplicadas à imagem de entrada resulta em uma camada filtrada de dimensão 7x7x1.

⁶ Alguns autores da literatura se referem a estes filtros usando o termo “*kernels*”.

⁷ Autores da área de sensoriamento remoto também usam o termo “extração de atributos”.

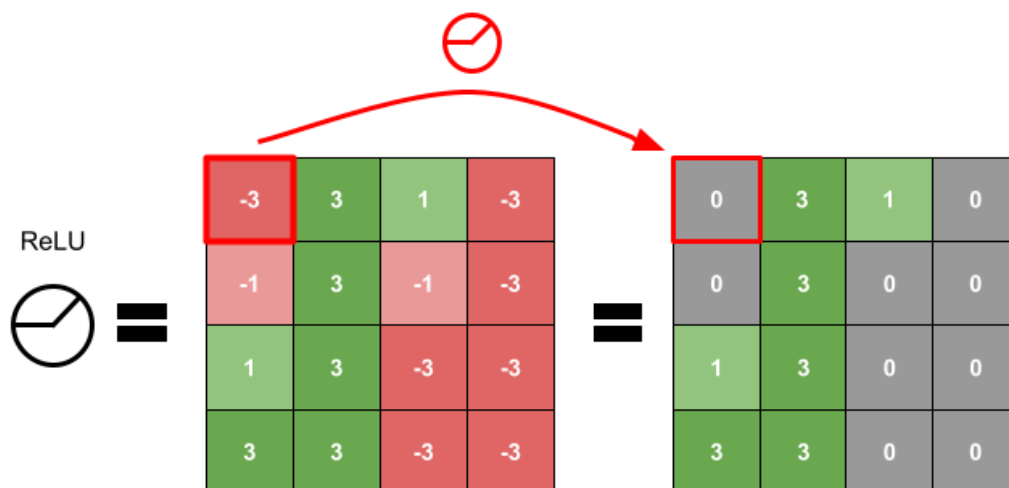
Figura 7 - Filtro de convolução 3x3 aplicado à imagem de entrada



Fonte: Adaptado de ROHRER (2018)

Para evitar que os números calculados pelas operações de convolução fiquem complexos demais, são realizadas operações de normalização dos dados obtidos pelas camadas. Para isso, usam-se funções e uma das mais utilizadas em CNN são as funções de unidade linear retificada (ReLU, do inglês *rectified linear unit*). Este método transforma em zero todos os valores negativos produzidos pelas convoluções, enquanto os valores positivos não são alterados (Figura 8).

Figura 8 - Camada de normalização ReLU



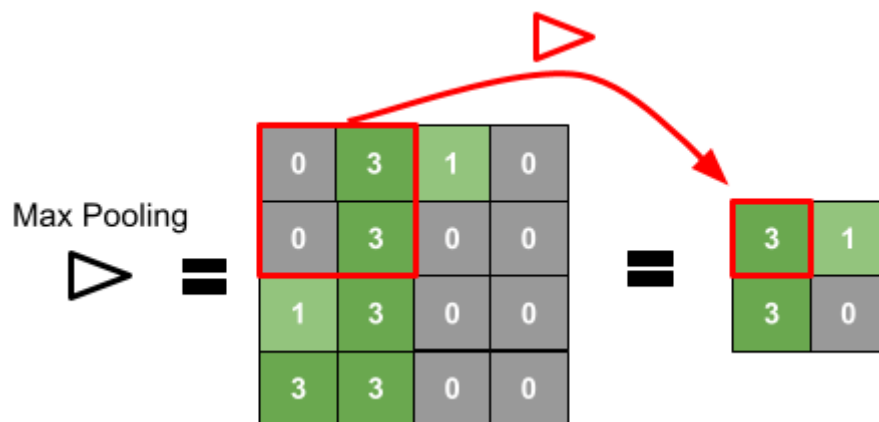
Fonte: Adaptado de ROHRER (2018)

A ReLU é bastante utilizada por conta de seu baixo custo computacional e também por alcançar os melhores resultados se comparada às outras funções de ativação como a função sigmóide e a função tangente hiperbólica (tanh). Outra

vantagem da ReLU é que ela tem a capacidade de resultar em zeros verdadeiros, diferente das outras funções de ativação que aprendem a se aproximar de zero mas não resultam em zeros verdadeiros.

As camadas de *pooling*, basicamente, baixam a resolução espacial dos dados de entrada, o que é útil para diminuir o número de parâmetros de ativação dessa camada da CNN (O'SHEA et al. 2015). As camadas *pooling* possuem dois hiperparâmetros: *stride* - e tamanho de janela que é a dimensão da janela de pixels, a ser percorrida a cada iteração do *stride*. Em geral, na literatura das CNN, usa-se o método do *max pooling* mas há CNN que utilizam o método de *pooling* médio. *Max pooling* é o método o qual a janela percorrida no dado de entrada da camada computa na saída o maior valor vetorial da matriz de entrada. A Figura 9 demonstra a segunda iteração de *max pooling* em uma camada, utilizando *stride* = 2 e tamanho de janela = 2.

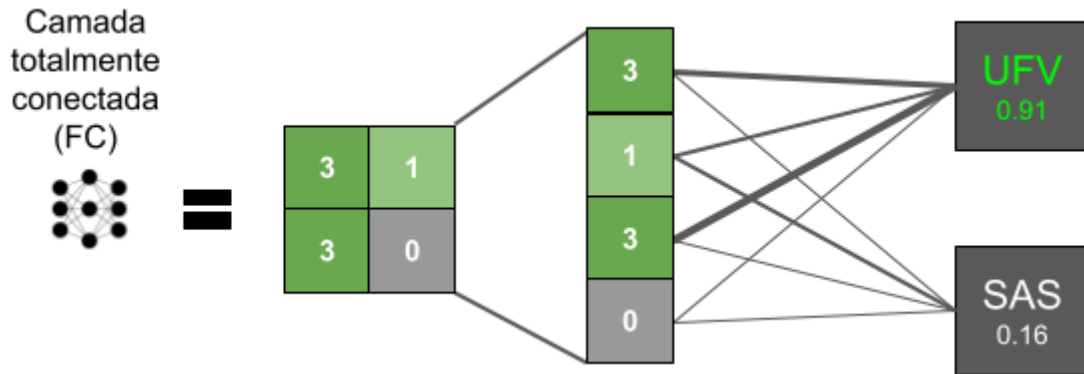
Figura 9 - Max pooling com tamanho de janela 2 e stride 2



Fonte: Adaptado de ROHRER (2018)

A camada FC (Figura 10), aplicada ao final da rede, é composta pela entrada de informações dada pela saída da camada anterior e a saída sendo o número de neurônios que corresponde ao número de classes que seu conjunto de dados possui. A saída da camada FC é dada pela probabilidade da imagem de entrada pertencer a uma classe (O'SHEA et al. 2015).

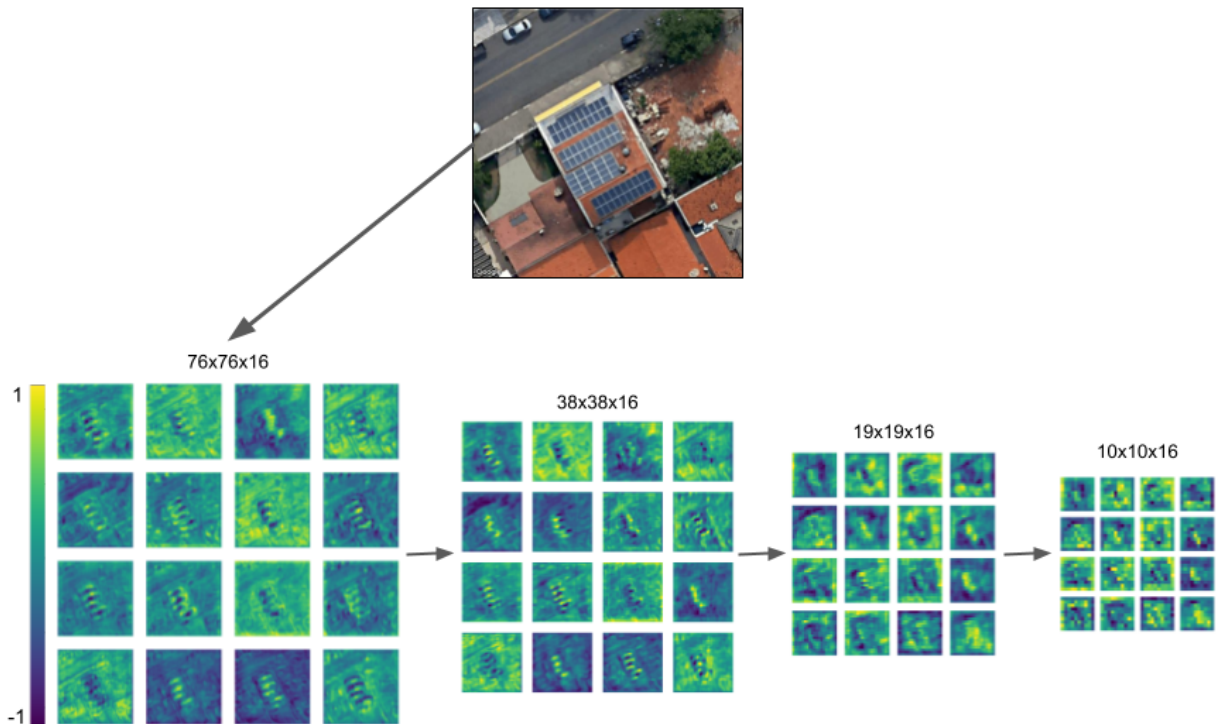
Figura 10 - Camada FC com 12 neurônios aplicada na camada normalizada



Fonte: Adaptado de ROHRER (2018)

Estas camadas da rede neural treinada, resultam em neurônios que são sensibilizados pelas características das imagens fornecidas. Os mapas de ativação (Figura 11) são a representação visual da sensibilidade das várias camadas da rede aos dados da imagem de entrada. Na Figura 11 é possível observar em quais áreas da imagem de entrada possuem maior probabilidade (em amarelo) de pertencer à classe “UFV” em diferentes resoluções (de 76x76 à 10x10 pixels).

Figura 11 - Mapas de ativação extraídos a partir de filtros em diferentes níveis



Fonte: Próprio autor (2021)

Dentro do universo de técnicas do AM, está o aprendizado profundo (em inglês *deep learning*) que são RNAs com múltiplas camadas de neurônios e busca trazer modelos computacionais que possam lidar com representações de dados com níveis de abstração a partir de conceitos simples generalizados e combinados, como bordas, contornos e cores (GOODFELLOW et al., 2016). Estes conceitos generalizados são transformações múltiplas não-lineares e são formados a partir dos dados de treinamento desta rede neural (SONG et al., 2020).

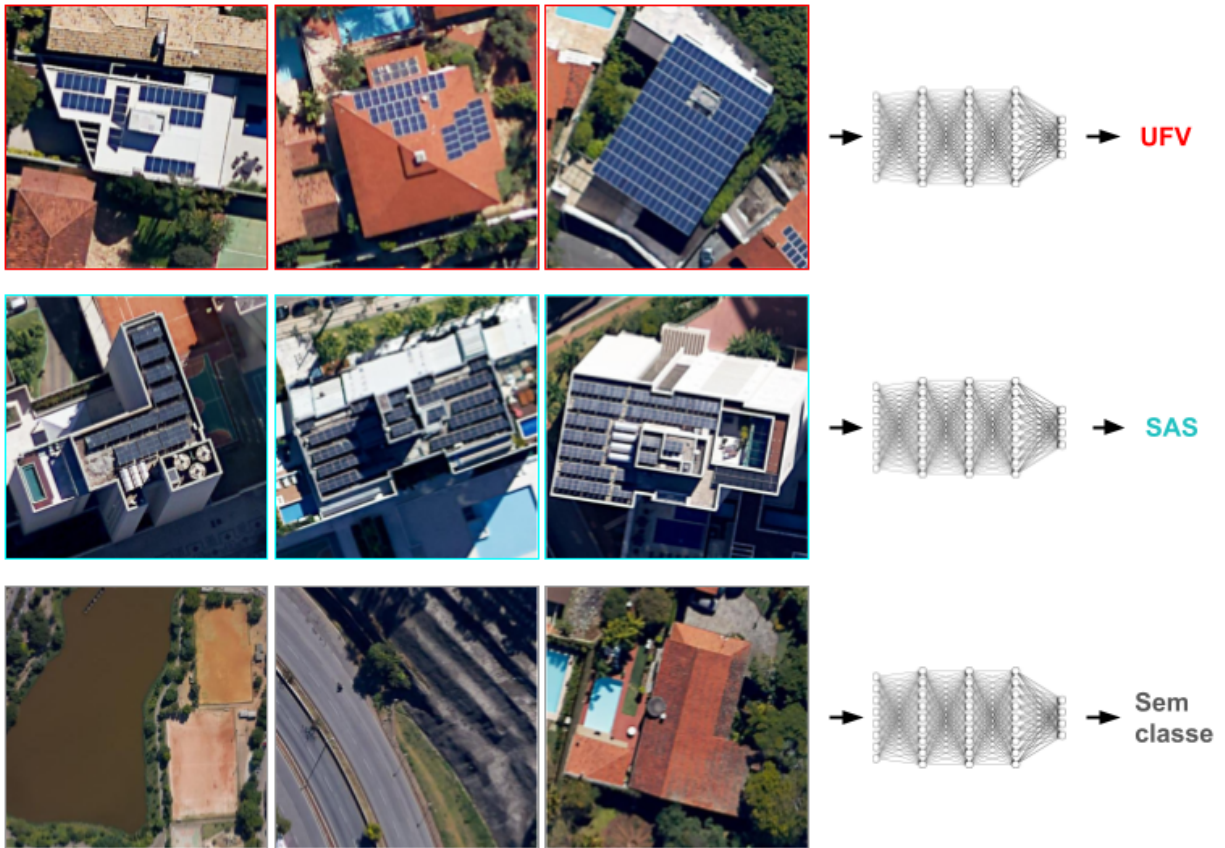
Marques (2016) afirma que técnicas de RNA ficaram robustas e ganharam destaque com o desenvolvimento do algoritmo de *backpropagation* para as RNA de aprendizado profundo. Este algoritmo melhorou consideravelmente o tempo de treinamento e a acurácia das redes profundas, especialmente quando se trata das CNN, amplamente utilizadas no processamento de imagens digitais.

No campo do sensoriamento remoto, o uso do aprendizado profundo tem sido cada vez maior e Song (2020) ressalta que no curso dos últimos anos a combinação de métodos de aprendizado profundo que extraem informações espectrais e espaciais tem sido proposta, e, atualmente é amplamente aceito que esta combinação traz melhorias importantes na performance das classificações.

Os termos “classificação” e “classificação de imagens” foram utilizados no texto, de maneira geral, referindo especificamente à segmentação de instâncias (classificação dos pixels de uma imagem separados por suas respectivas classes e unidades individuais) em imagens digitais. A definição expandida da técnica de segmentação de instâncias está descrita abaixo. Mas há também o conceito da “classificação de imagem” como uma tarefa específica da área de visão computacional, não sendo um sinônimo para o termo usado durante o texto, em geral.

A classificação de imagens é a tarefa que atribui uma classe (rótulo) a uma imagem, sem definir exatamente onde o objeto com a classe está localizado. A Figura 12 demonstra três exemplos de imagens classificadas com a classe “UFV” (destaque vermelho), três imagens classificadas como “SAS” (destaque azul anil) e três imagens onde não foram detectadas nenhuma classe (destaque em cinza).

Figura 12 - Exemplo da tarefa de classificação de imagens

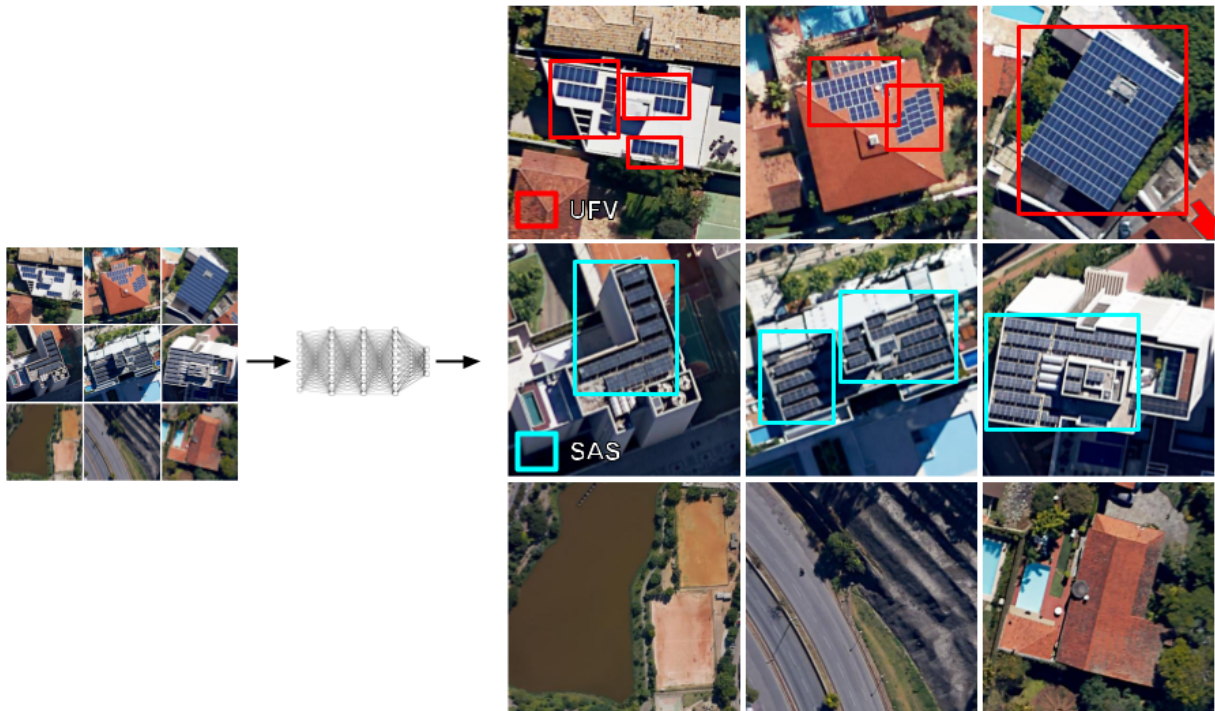


Fonte: Adaptado de Google Earth (2018)

Na classificação de imagem, o modelador define as classes que cada imagem representa em um conjunto de dados, alimenta a CNN com imagens contendo exemplos diferentes dessas classes, e em seguida, a CNN calcula a probabilidade dessas imagens de entrada pertencerem a esta classe. Esta probabilidade pode ser binária (sendo pertencimento verdadeiro ou pertencimento falso) ou por meio de uma função exponencial normalizada, um vetor de probabilidades para cada classe do conjunto de dados (PEŠEK, 2018).

A detecção de objetos se caracteriza pela localização espacial de cada uma das classes identificadas em uma imagem por meio de uma caixa delimitadora contendo as coordenadas do objeto e a sua respectiva classe associada. Brownlee (2019) aponta que a detecção de objetos busca respostas para os problemas de onde estão os objetos em uma imagem (localização), quais seus tamanhos (dimensões) e o que eles são (classificação). A Figura 13 apresenta exemplos de imagens com seus respectivos objetos detectados conforme suas classes.

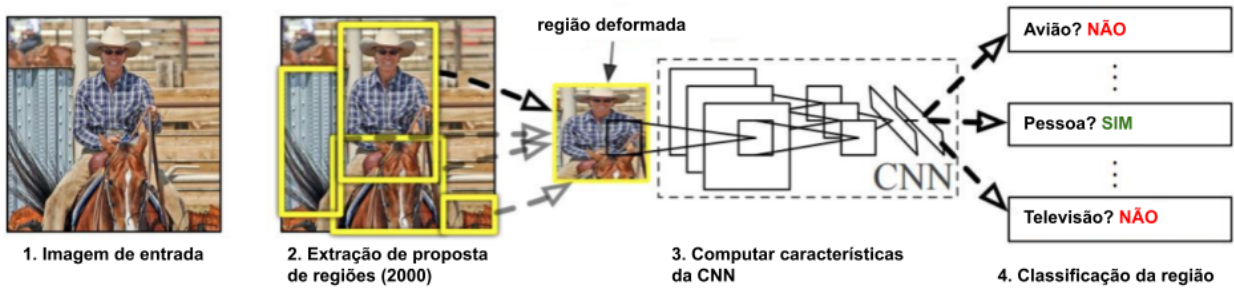
Figura 13 - Exemplos da tarefa de detecção de objetos



Fonte: Adaptado de Google Earth (2018)

As arquiteturas de CNN que resultam em detecção de objetos servem como base para a segmentação semântica e segmentação de instâncias, descritas nas seções seguintes. O *insight* para o desenvolvimento dos algoritmos de detecção de objetos em imagens partiram principalmente com a proposta das R-CNN (*Region-based Convolutional Neural Networks*, do inglês redes neurais convolucionais baseadas em regiões) por Girshick et al. (2014). As R-CNN buscam por 2000 propostas de caixas delimitadoras usando o algoritmo “busca seletiva” (em inglês *selective search*) na imagem, processando objetos de cores, dimensões e texturas semelhantes entre si. Cada uma das regiões propostas pelo algoritmo recebe uma caixa delimitadora que alimenta a R-CNN que usa essas regiões para extrair as características dentro delas usando CNN de aprendizado profundo (BROWNLEE, 2019). Os recursos extraídos alimentam um classificador SVM que irá computar se a região candidata possui um objeto de determinada classe (HERRMANN, 2019). A Figura 14 apresenta o desenho esquemático da arquitetura da R-CNN.

Figura 14 - Arquitetura de R-CNN



Fonte: Adaptado de Girshick et al. (2014)

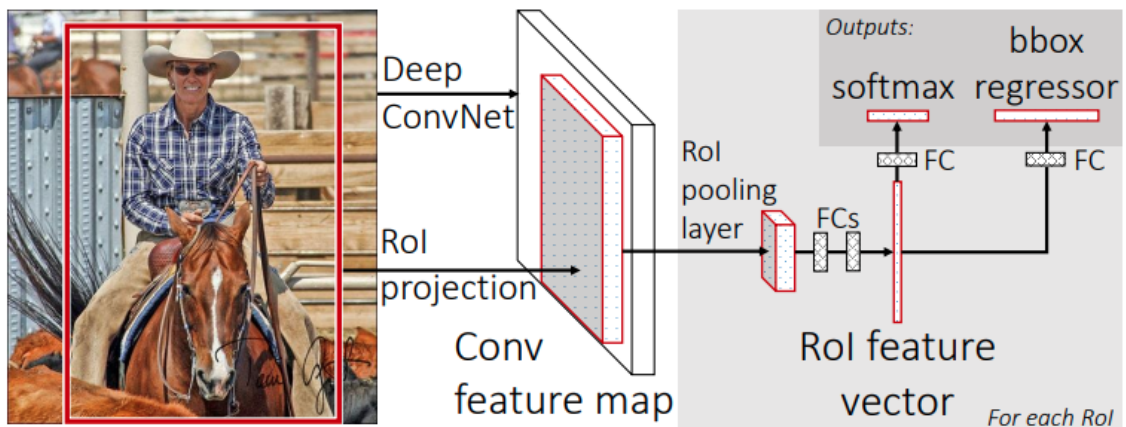
Apesar da R-CNN ter alcançado um desempenho superior a arquiteturas de natureza similar, há vários problemas com esta rede, principalmente, relacionados à performance e tempo gasto no processamento da classificação. Esta lentidão acontece porque cada uma das 2000 regiões propostas precisa ser passada separadamente, pela necessidade de gerar as características das imagens, uma rede para definir as classes de cada região e isso torna o algoritmo lento demais tanto na etapa de treinamento quanto na etapa de inferência do modelo (PEŠEK, 2018).

Logo no ano seguinte da proposta da R-CNN, um de seus criadores propôs soluções para os gargalos desta arquitetura e publicou o trabalho contendo a arquitetura Fast R-CNN, que seria uma forma de classificar objetos de maneira mais eficiente do que a R-CNN (GIRSHICK, 2015). A proposta da Fast R-CNN é trocar o envio de todas as propostas de região como entradas da rede, a imagem de entrada inteira alimenta a rede em um primeiro momento e com a imagem inteira a CNN gera mapas de ativação e a partir daí, estes mapas recebem propostas de regiões. Estas regiões são então transformadas em vetores de tamanho fixo por uma camada de *RoI pooling*, ou seja, são encolhidas para vetores de dimensão 7×7 que posteriormente são transformados em um vetor unidimensional. O *RoI pooling* otimiza a R-CNN pois diminui o número de parâmetros da rede ao remover as sobreposições das regiões propostas. Por fim, cada um desses vetores alimenta uma camada FC que faz a previsão da probabilidade destes vetores pertencerem a cada classe e também estes vetores passam por uma etapa de regressão para que o modelo compute, a partir dos valores de referência do treinamento, o quanto este

vetor precisa ser modificado em termos de dimensão e localização (seus valores de altura, largura, x e y) para corresponder ao valor de referência. Portanto, em resumo, o que a Fast R-CNN otimiza em relação à sua rede predecessora é o fato de não ser necessário computar classificações e regressões (SVM) para 2000 regiões em cada imagem de entrada, mas apenas as regiões propostas e “encolhidas” pelo *Rol pooling* e também unificar os processos de propostas de regiões, classificação e regressão na mesma rede.

A Figura 15 apresenta a arquitetura da Fast R-CNN, proposta por Girshick (2015).

Figura 15 - Arquitetura de Fast R-CNN

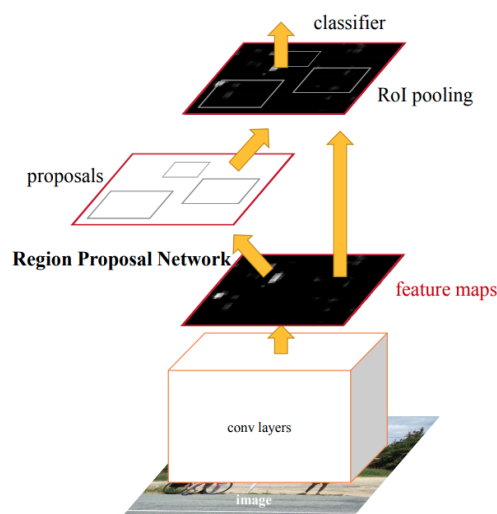


Fonte: Extraído de Girshick (2015)

Mas como a R-CNN e a Fast R-CNN usam a busca seletiva para gerar as regiões propostas, este algoritmo ainda trazia um gargalo para estas redes em termos de performance e tempo de computação.

No ano seguinte, Ren et al. (2016) propuseram a rede Faster R-CNN (Figura 16) que resolveu o gargalo supracitado ao remover a etapa de busca seletiva na proposta de regiões e criar uma rede de proposta de regiões (RPN, do inglês *Region Proposal Network*) que utiliza menos recursos computacionais e acelera o processo de gerar propostas de regiões de interesse.

Figura 16 - Representação esquemática da arquitetura da Faster R-CNN

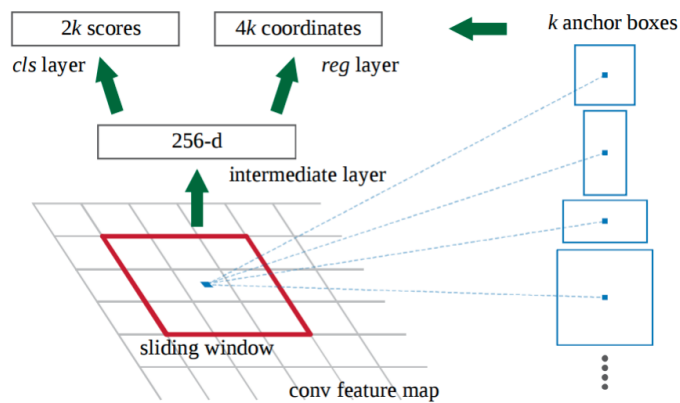


Fonte: Extraído de Ren et al. (2016)

A abordagem da RPN é feita usando âncoras que são caixas delimitadoras de propostas e pontuações, criadas a partir de uma janela deslizante em todo o mapa de ativação. Estas âncoras são criadas em diferentes aspectos e escalas, portanto, possuem dimensões variadas e com isso reduz o número de parâmetros do modelo, tornando-o mais leve (PEŠEK, 2018). A Figura 17 representa o esquema da arquitetura da RPN. Cada uma destas âncoras alimenta neurônios da rede que resultam em uma medida de probabilidade desta janela pertencer a um objeto (*objectness*). Importante destacar que nesta etapa ainda não se estabelece qual a classe desse objeto, mas sim se é um objeto ou não. Estas âncoras também servem de entrada de outros neurônios da rede que fazem a regressão do quanto este objeto detectado precisa ser deslocado para corresponder aos dados dos valores de referência (anotações feitas pelo modelador). A correspondência dos valores de referência em relação às âncoras é feita usando o método *Intersection-over-Union* (IoU)⁸. A rede considera objetos as âncoras que possuem um IoU > 0,5 (mais da metade da área da âncora corresponde a área do valor de referência).

⁸ Alguns autores se referem a esta medida (IoU) como Índice de Jaccard.

Figura 17 - Arquitetura da RPN



Fonte: Extraído de Ren et al. (2016)

Considerando que a segmentação de instâncias é uma abordagem de classificação pixel a pixel, estas métricas são compostas pela taxa de acertos/erros do modelo para cada pixel, comparados os dados de treinamento e predição, onde Verdadeiro Positivo (TP) é elemento corretamente classificado como pertencente à determinada classe; Verdadeiro Negativo (TN) é o elemento corretamente classificado como não pertencente à determinada classe, Falso Positivo (FP) é o elemento erroneamente classificado como pertencente à determinada classe; e Falso Negativo (FN) é o elemento erroneamente classificado como não pertencente à determinada classe.

Para avaliar a performance do modelo de segmentação de instâncias, os principais autores da área usam majoritariamente a métrica de IoU que é uma medida da intersecção das áreas de pixels de referência e áreas de pixels preditas pelo modelo divididos pela união de ambas as áreas (Equação 1).

$$IoU = \frac{TP}{TP + FP + FN} \quad (1)$$

Onde:

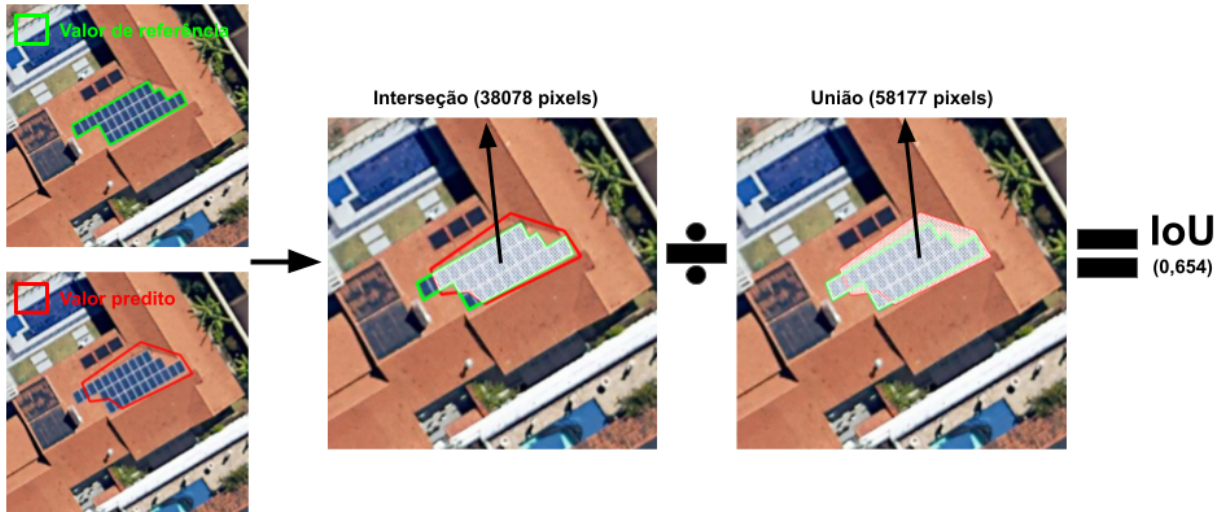
TP = verdadeiro positivo

FP = falso positivo

FN = falso negativo

Uma sobreposição perfeita da área predita pelo modelo e a área anotada como valor de referência resulta em $IoU = 1$, já a sobreposição inexistente da área predita e a anotada resulta em $IoU = 0$. A Figura 18 exemplifica a métrica.

Figura 18 - Desenho esquemático da métrica IoU



Fonte: Adaptado de Google Earth (2018)

Outras métricas utilizadas na validação do modelo são as métricas de acurácia, precisão e *recall*. Então, a partir destas medições, a acurácia (Equação 2) é dada pela soma dos valores corretamente classificados divididos por todos os elementos calculados:

$$Acuracia = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Onde:

TP = verdadeiro positivo

TN = verdadeiro negativo

FP = falso positivo

FN = falso negativo

A medida de precisão (Equação 3) é dada pelos elementos corretamente classificados (TP) dividido pelo número de predições (TP+FP):

$$Precisao = \frac{TP}{TP + FP} \quad (3)$$

Onde:

TP = verdadeiro positivo

TN = verdadeiro negativo

FP = falso positivo

FN = falso negativo

E a medida de *recall* (Equação 4) é dada pelo número de objetos corretamente detectados dividido (TP) pelo número de detecções positivas (TP + FN):

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Onde:

TP = verdadeiro positivo

TN = verdadeiro negativo

FP = falso positivo

FN = falso negativo

A métrica *average precision* (AP) usa o valor médio de 10 IoU calculados em diferentes valores de *threshold*⁹, desde 0,5 até 0,95 em passos de 0,05. Quanto maior o valor de AP (mais próximo à 1) melhor a precisão do modelo. A validação também considera a precisão média em diferentes tamanhos de objetos, sendo AP_{peq} para objetos com área menor que 32² pixels, AP_{méd} para objetos com área maior que 32² pixels e menor que 96² e AP_{gnd} para objetos maiores que 96² pixels (Carvalho et al., 2020). Por fim, a métrica *average recall* (AR) serve para medir o poder do modelo em encontrar verdadeiros positivos (TP) e é calculado da mesma maneira que AP, mas fazendo a média do *recall* em diferentes níveis de tamanhos de objetos e *threshold*.

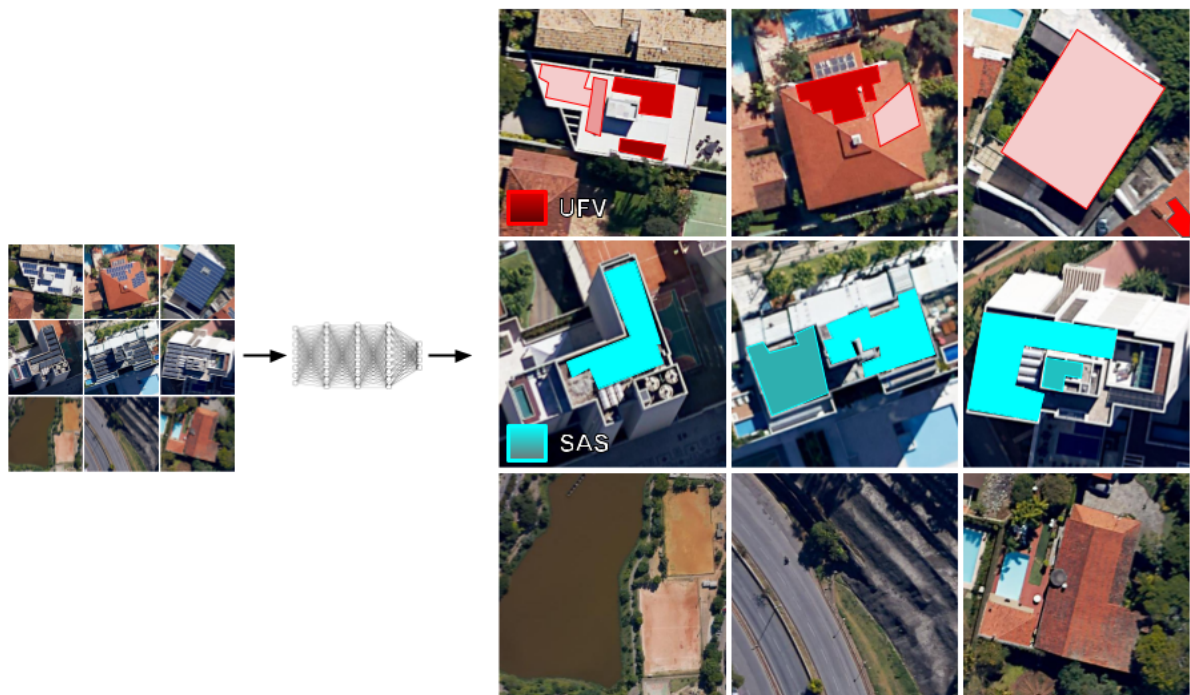
⁹ *Threshold* (limiar) se refere ao valor mínimo de confiança das predições que vai ao conjunto final de resultados. Caso a confiança destas predições seja menor que o *threshold*, a predição é descartada.

3.4 Segmentação de Instâncias

A detecção de objetos é um método que representa uma importante evolução se comparada à classificação de imagem, uma vez que não resulta apenas nas classes de imagens mas também dá a localização espacial e a classe de objetos em uma imagem na forma de caixas delimitadoras ou centróides. Mas a segmentação de instâncias vai um passo além e proporciona uma localização ainda mais refinada que a da detecção de objetos, ao classificar pixel a pixel os objetos de uma imagem, bem como separando seus diferentes entes individualmente (HAFIZ et al., 2020 e FENG, 2020).

A Figura 19 apresenta um desenho esquemático da tarefa de segmentação de instâncias onde cada um dos polígonos coloridos nas cores vermelho e azul anil é uma instância individual das classes UFV e SAS, respectivamente.

Figura 19 - Exemplos da tarefa de segmentação de instâncias



Fonte: Adaptado de Google Earth (2018)

Cada uma das instâncias está classificada a nível de pixel, diferentemente por exemplo da detecção de objetos que apresenta apenas uma caixa delimitadora da localização destes objetos na imagem.

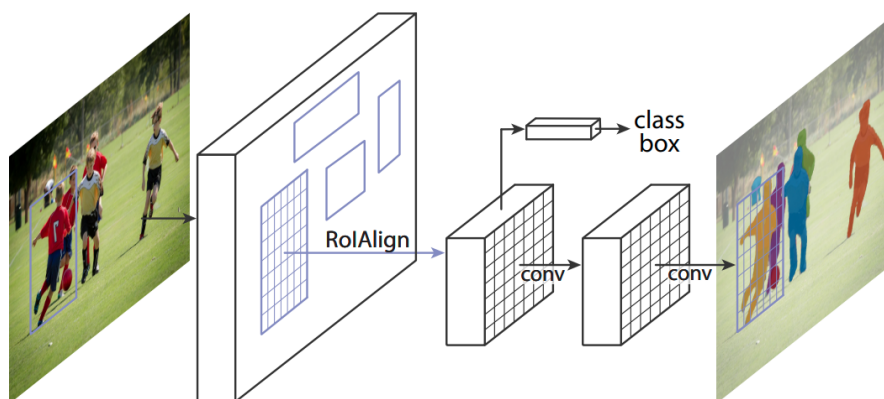
Utilizando a Faster R-CNN como base, o departamento de pesquisa em inteligência artificial do Facebook, *Facebook AI Research* (FAIR) desenvolveu uma extensão da Faster R-CNN que além de detecção de objetos, também executava a segmentação de instâncias. À esta rede foi dado o nome de Mask R-CNN¹⁰, voltada para a segmentação de instâncias em imagens. Desde então, esta técnica tem sido o estado da arte em segmentação de instâncias em diversos domínios.

Os autores da Mask R-CNN resumem sua proposta:

“Mask R-CNN é conceitualmente simples: Faster R-CNN possui duas saídas para cada objeto candidato, um rótulo de classe e uma caixa delimitadora deslocada; a isso adicionamos uma terceira ramificação que resulta em uma saída de máscaras do objeto. Mask R-CNN é portanto uma ideia intuitiva e natural. Mas a saída de máscara adicional é distinta das saídas de classe e caixa delimitadora, exigindo a extração de uma configuração espacial muito mais precisa de um objeto. Em seguida, apresentamos os elementos-chave de Mask R-CNN, incluindo o alinhamento pixel a pixel, que é a principal peça que faltava em Fast/Faster R-CNN.” (He et al., 2017)

Portanto, a arquitetura da Mask R-CNN (Figura 20) é dividida em duas arquiteturas de redes separadas: *backbone* e *head*. A primeira rede, *backbone*, consiste na extração de mapas de ativação e a segunda etapa (*head*) é a rede que objetiva a detecção de objetos em caixas limitadoras (RoI) - da mesma maneira que a Faster R-CNN, usando classificação e regressão para tal - e também uma predição de máscaras aplicada separadamente a cada uma das RoI encontradas (HE et al., 2017).

Figura 20 - Desenho esquemático da arquitetura de Mask R-CNN para segmentação de instâncias

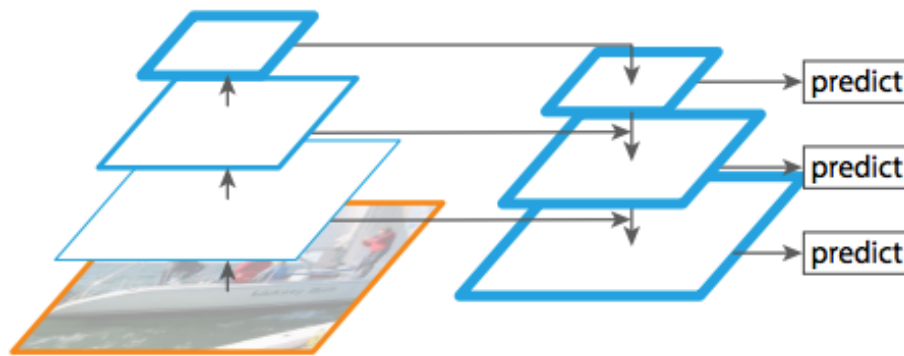


Fonte: Extraído de He et al. (2017)

¹⁰ Publicação “Mask R-CNN” de He et al. (2017) - A implementação do código desta rede está disponível em https://github.com/matterport/Mask_RCNN

Os autores exploraram o uso de uma técnica chamada *Feature Pyramid Network* (FPN) proposta por Lin et al. (2017) na rede *backbone*, para extração dos mapas de ativação de maneira mais efetiva (HE et al., 2017). A FPN consiste em uma arquitetura que extrai os mapas de ativação e alimenta a RPN para detecção de objetos. A RPN, por sua vez, aplica uma janela deslizante nos mapas de ativação supracitados, em diferentes níveis de escala e faz previsões do *objectness*¹¹ de cada janela, bem como da regressão da localização de sua caixa delimitadora (PEŠEK, 2018). A Figura 21 apresenta a arquitetura da FPN.

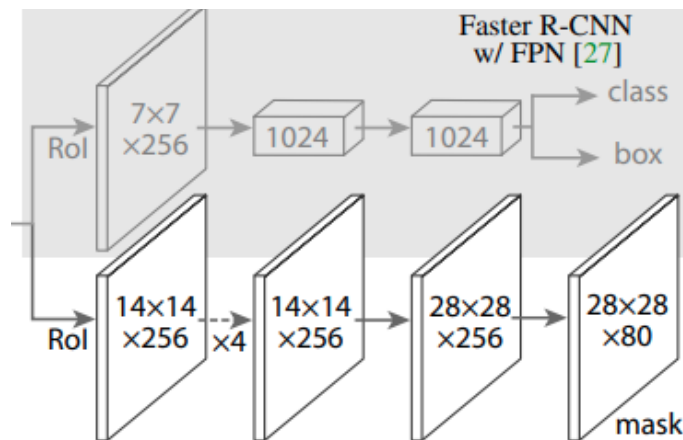
Figura 21 - Arquitetura da FPN



Fonte: Extraído de Pešek (2018)

A segunda etapa é a rede chamada “head” (Figura 22).

Figura 22 - Arquitetura da rede *head*

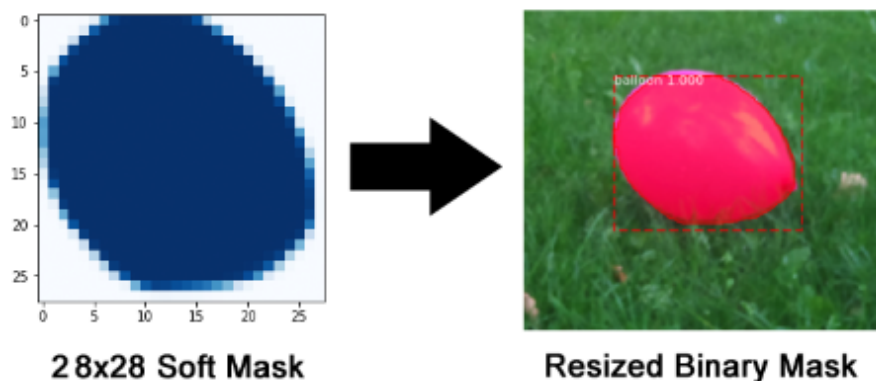


Fonte: Extraído de He et al. (2017)

¹¹ Probabilidade de um elemento pertencer a determinado objeto.

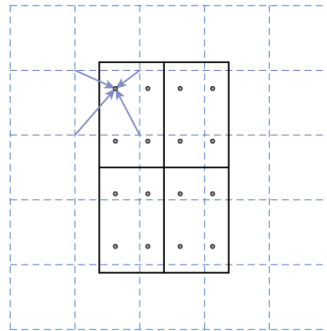
Esta rede é onde foi adicionada a extensão mais relevante proposta em Mask R-CNN, que permitiu a criação da segmentação de instâncias a partir da arquitetura da Faster R-CNN. O que foi modificado foi a adição das camadas FC de predição de máscaras (Figura 16, a parte inferior da figura destaca essas camadas em cor escura) onde cada uma das RoI recebe uma predição binária, pixel a pixel, do pertencimento ou não deste pixel à uma classe (PEŠEK, 2018). Importante ressaltar que essa predição pixel a pixel é feita, por padrão da Mask R-CNN, numa escala de 28x28 pixels e redimensionada para o tamanho da imagem original de entrada (Figura 23).

Figura 23 - Redimensionamento da máscara predita de 28x28 pixels (à esquerda) para o tamanho da imagem original (à direita)



Fonte: Extraído de Abdulla (2018)

A outra novidade proposta em Mask R-CNN foi a substituição do RoI *pooling* da Faster R-CNN pela técnica RoI *Align* (alinhamento das regiões de interesse). Esta alteração objetivou corrigir os desalinhamentos na escala dos pixels já que a RoI *pooling* fazia quantização e arredondamentos baseados na posição de uma caixa delimitadora e não de uma máscara (PEŠEK, 2018). Portanto, para corrigir esse desalinhamento, o RoI *Align* executa uma interpolação bilinear para calcular os valores exatos dos recursos de entrada em quatro locais regularmente amostrados em cada RoI e agrega o resultado usando valores máximos ou médios. A Figura 24 mostra como o RoI *Align* é deslocado usando o procedimento supracitado.

Figura 24 - Rol Align

Fonte: Extraído de He et al. (2017)

He et al. (2017) afirmam que o uso do *Rol Align* traz um enorme impacto nas métricas de performance de segmentação, com um aumento de 10% até 50% na acurácia das máscaras preditoras.

Detectron2 é um *framework* que implementa os principais algoritmos do estado da arte de detecção de objetos, segmentação semântica e segmentação de instâncias em imagens (WU et al., 2019). Este *framework* foi criado pela FAIR e é baseado no PyTorch¹². Entre os algoritmos do estado da arte que estão implementados em Detectron2 estão *DensePose*, *PointRend*, *DeepLab* e Mask R-CNN, que foi proposto pelo próprio FAIR.

O objetivo do Facebook ao lançar o Detectron2 foi:

“[...] oferecer suporte à uma ampla gama de modelos de última geração de detecção e segmentação de objetos disponíveis hoje, mas também atender ao cenário em constante mudança da pesquisa de ponta. A pesquisa inovadora, por definição, envolve a invenção de novos modelos que provavelmente quebrarão premissas de projeto dos modelos já existentes. Isso significa construir um software para oferecer suporte a um conjunto de requisitos não especificáveis, ao mesmo tempo tornando seu uso o mais simples possível. Esperamos desenvolver e refinar continuamente o Detectron2 a serviço desse objetivo. Com a biblioteca agora disponível para a comunidade de aprendizado de máquina em geral, estamos ansiosos para colaborar e aprender com outras pessoas à medida que expandimos os limites do que é possível em sistemas de visão computacional” (FAIR, 2019)

¹² PyTorch é uma plataforma/pacote de computação científica baseado na linguagem de programação Python, que otimiza o uso de placas gráficas (GPU) para a construção de redes de aprendizado profundo. (PYTORCH, 2021).

Ao implementar estes modelos com códigos otimizados, as redes implementadas em Detectron2 podem ser até quatro vezes¹³ mais rápidas que seus códigos originais, apenas com estas otimizações (FAIR, 2019).

Resumidamente, Detectron2 torna a implementação, manipulação e uso das principais técnicas de classificação de imagens mais fácil, usando linguagem Python.

O modelador de uma rede de segmentação de instâncias, precisa fornecer a essa rede os dados de treinamento, validação e teste que ajudem o modelo a entender quais são as características dos objetos e classes que se deseja identificar em imagens. Para isso, é necessário criar anotações dos valores de referência¹⁴.

Grande parte do tempo gasto na criação de modelos de classificação de imagens é direcionado ao tratamento, curadoria e obtenção dos dados que alimentam estes modelos (LABELBOX, 2021).

Em especial, a segmentação de instâncias apresenta um desafio maior do que outros métodos de classificação, por se tratar de uma classificação pixel a pixel, as anotações também são feitas na escala de pixels com detalhamento espacial muito mais refinado, e, portanto, o trabalho é ainda mais complexo e demorado que em outras tarefas como a detecção de objetos.

Carvalho et al. (2020) aponta que a comunidade de visão computacional faz uso extensivo de bancos de dados públicos contendo dados previamente anotados como *ImageNet* (DENG et al., 2009), *Cityscapes* (CORDTS et al., 2016), *Open Images v4* (KUZNETSOVA et al., 2020) e *Creating Common Objects in Context* (COCO) de Lin et al. (2014), já que a criação de novas anotações é trabalhosa e demanda muito tempo. Mas estes bancos de dados contém apenas imagens de cenas naturais, urbanas, fotografias comuns, e a área do sensoriamento remoto carece de extensas fontes de dados anotados. Mas os mesmos autores apontam que ainda que não existam bancos de dados com estas anotações específicas, existem diversos bancos de arquivos de formato vetorial e rasters - amplamente

¹³ Detectron2 comparou sua implementação otimizada de Mask R-CNN com a implementação original e aferiu que a taxa de transferências do *framework* foi de 62 imagens/seg. contra 14 imagens/seg, portanto, 4,42 vezes mais rápida.

¹⁴ Termos correlatos na literatura: “*ground truth*” ou “verdade do terreno”

usados nos ambientes SIG - que podem ser adaptados com o propósito de servirem como anotações.

Um dos procedimentos mais usados na anotação de dados para segmentação de instâncias é a anotação de formato COCO. As anotações neste formato são compostas por pontos com as localizações no plano cartesiano bidimensional da matriz (x e y) de cada vértice de um polígono anotado e sua respectiva classe, em um arquivo de formato JSON¹⁵. Junto à anotação do polígono é necessário fornecer também a imagem referência, à qual a anotação se refere.

Na seção 4.5 especifica-se como foram criadas as anotações dos dados de referência, executando a adaptação de dados vetoriais para formato COCO.

Durante o treinamento de redes de aprendizado profundo, as primeiras camadas executam a extração de características de baixo nível à exemplo do exposto na Figura 11, estas camadas não respondem apenas à bases de dados muito específicas, mas sim às imagens, em geral (ABDURAHMANOV, 2020). Estas características são representadas pelos mapas de ativação, já conceituados anteriormente. Por exemplo, os primeiros mapas de ativação de uma rede neural que foi treinada para classificar imagens de carros ou pedestres podem ser reaproveitados em redes que classificam gatos e cachorros. A esta tarefa se dá o nome de transferência de aprendizado. Esta técnica se justifica principalmente pelo fato de que o treino de CNN treinadas a partir de parâmetros iniciais aleatórios é um processo caro em termos de tempo e recursos computacionais, além de demandar bases de dados contendo milhões de imagens diferentes de treinamento. Portanto, o uso de pesos pré-treinados permite que os operadores pulem uma etapa computacionalmente exaustiva na criação de modelos.

A literatura da área de visão computacional utiliza redes pré-treinadas oriundas, principalmente, de bancos de imagens públicas, voltados à pesquisas científicas, como *ImageNet*, COCO, *Cityscapes*, *Open Images v4*. O banco de imagens *ImageNet*, por exemplo, possui 14.197.122 imagens organizadas em 21.841 subcategorias (650 amostras de cada subcategoria, em média). Além das imagens, estes bancos também possuem as anotações de valores de referências

¹⁵ JSON (*JavaScript Object Notation*) é um formato leve de intercâmbio de dados computacionais. Baseado em um subconjunto da linguagem de programação JavaScript (ECMA, 2017).

para detecção de objetos (caixas delimitadoras) e também de segmentação (anotações vetoriais, pixel a pixel).

Detectron2 implementa por padrão, as redes pré-treinadas em imagens do *ImageNet* e *COCO*.

3.5 Aumento de dados

O aumento de dados¹⁶ é a técnica usada para alterar características de imagens usando transformações com o propósito de aumentar a diversidade dos dados de treinamento sem precisar coletar novos dados, e, com isso, os modelos passam a ter um maior poder de generalização e menor viés (TONG, 2020).

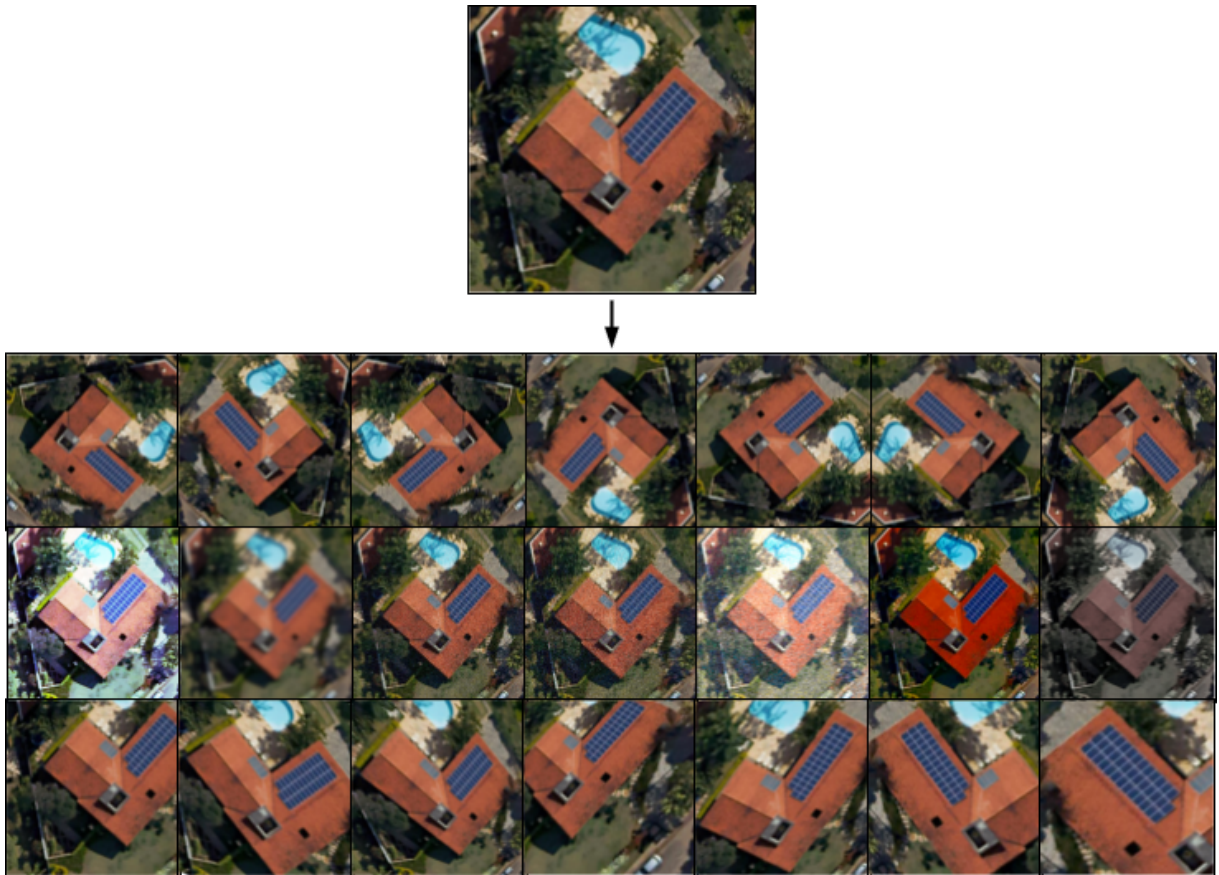
A performance de um modelo de classificação de imagens é dada por seu desempenho relativo à classe menos representativa dos dados de entrada, ou seja, o aumento de dados pode ser utilizado para melhorar a representatividade das classes sub-representadas nos dados de treinamento (ELISHA, 2020).

Entre as transformações que podem ser aplicadas às imagens para o aumento de dados estão as transformações afim, transformações de perspectiva, mudanças no contraste, adição de ruído gaussiano, recorte de regiões das imagens, desfocamento, rotação, espelhamento e outros (TONG, 2020).

A Figura 25 demonstra exemplos das transformações supracitadas a partir de uma imagem de entrada. Nesta figura é possível ver como criar novos dados de treinamento a partir de uma única amostra quando aplicada à técnicas de aumento de dados. É importante ressaltar que apesar de muito parecidas, estas imagens são muito diferentes a nível de valores de pixels e portanto, oferecem muitas informações adicionais à rede sobre o que é a classe alvo e conseqüentemente aumenta muito o poder de generalização do modelo.

¹⁶ A seção 3.5 trata especificamente do aumento de dados de CNN de classificação de imagens. Outros domínios de Aprendizado de Máquina também podem se beneficiar desta técnica, mas a descrição e aplicação destes não está no escopo desta pesquisa.

Figura 25 - Amostra aumentada para 21 novas amostras



Fonte: Adaptado de Google Earth (2018)

3.6 Sistemas fotovoltaicos e sistemas de aquecimento solar

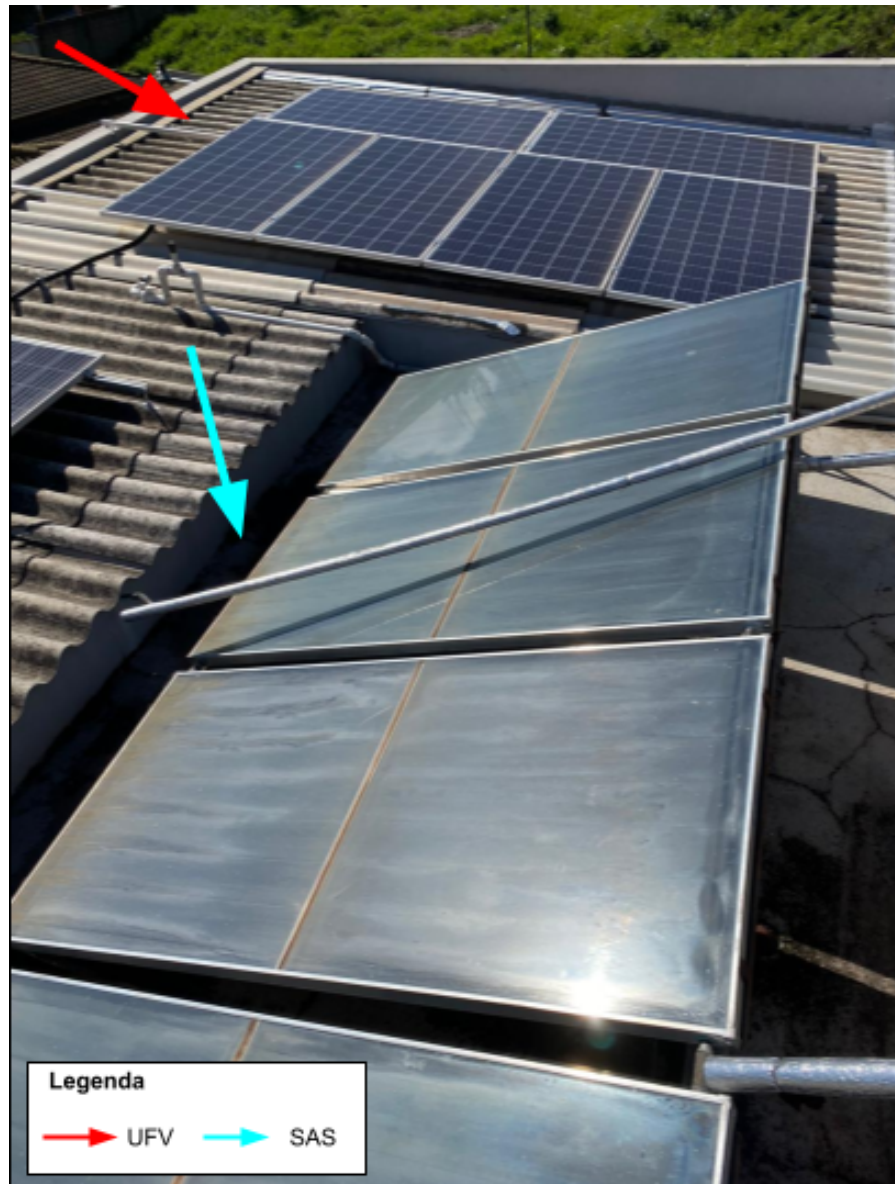
Sistemas fotovoltaicos ou Unidades Geradoras Fotovoltaicas (UFV) são aqueles capazes de converter a luz solar em energia elétrica, utilizando painéis com células contendo semicondutores e aproveitando-se do efeito fotovoltaico. As células fotovoltaicas geralmente são constituídas de silício mas podem ser feitas de outros materiais como a perovskita¹⁷. Já os sistemas de aquecimento solar (SAS), capturam a energia térmica proveniente do sol e as transformam em energia mecânica, geralmente utilizada para aquecer água ou ambientes, sem o uso de energia elétrica (RYLATT et al., 2001). As feições e características dos dois sistemas são relativamente parecidas e não foram encontradas referências na literatura de

¹⁷ Perovskita é o nome de uma estrutura cristalina que pode ser inorgânica ou híbrida, onde alguns componentes são orgânicos, como é o caso das utilizadas em células solares. A família de perovskitas com propriedades fotovoltaicas é composta por um cátion orgânico e um inorgânico, sendo chumbo ou estanho, e um halogênio, sendo iodo, bromo ou cloro (SZOSTAK, 2016).

trabalhos que façam a diferenciação entre cada um dos sistemas na detecção e/ou diferenciação destes objetos separadamente.

A Figura 26 apresenta uma fotografia de um SAS e um UFV em um mesmo telhado residencial.

Figura 26 - Fotografia de SAS em primeiro plano e UFV em segundo plano



Fonte: Fotografia do autor - Data: 20/08/2020

Puttemans et al. (2015) apontaram que a detecção de UFV por meio de ortofotos é um desafio, principalmente pela variância nas cores que estes painéis apresentam (por conta da reflexão especular), bem como na variabilidade dos

formatos que estes objetos podem ter (disposição, rotação e inclinação devido aos diferentes ângulos de inclinação dos telhados). Os autores destacaram também que a baixa resolução é um problema, já que por exemplo em uma imagem de resolução espacial de 25 cm/pixel, um único painel solar individual possui dimensão média de 9x7 pixels. O capítulo 6 discute como pequenos objetos são mais dificilmente detectados nas tarefas de classificação, detecção e segmentação de imagens.

A Figura 27 exemplifica estes problemas, onde numa mesma imagem tirada a 30 metros dos objetos-alvo, as mesmas classes de objeto (SAS) possuem cores, contraste, posicionamento e formatos visualmente distintos. Nesta mesma figura também é possível perceber a similaridade das cores e geometria dos painéis de ambas as classes SAS e UFV. Outro fator que torna a tarefa de classificação destes alvos um desafio é sua disposição nos telhados, como são colocados muito próximos uns aos outros e são muito semelhantes, um classificador pode ter problemas ao separá-los.

Figura 27 - Exemplo de diferenças intraclasses SAS (destacadas em azul anil) e semelhança de ambas classes SAS e UFV



Fonte: Junqueira (2021)

Em ortofotos, estes dois sistemas possuem similaridades e são difíceis de se distinguir um do outro, mesmo que tenham uma melhor resolução espacial se comparadas às imagens de satélite. A Figura 28 demonstra um exemplo de uma ortofoto contendo diversos exemplos dos dois sistemas, UFV e SAS em uma mesma cena, comparada à uma fotografia tirada por drone do mesmo local. Nesta mesma figura também é possível perceber a diferença na riqueza de detalhes da imagem de drone feita a 30 metros de altura do solo (Figura 28, acima) se comparada à ortofoto do mesmo local tomada de altura desconhecida (Figura 28, abaixo).

Figura 28 - Fotografia aérea produzida por drone (acima) comparada a ortofoto do Google (abaixo) no mesmo local



Fonte: Junqueira (2021) (acima) e Google Earth (2018) (abaixo)

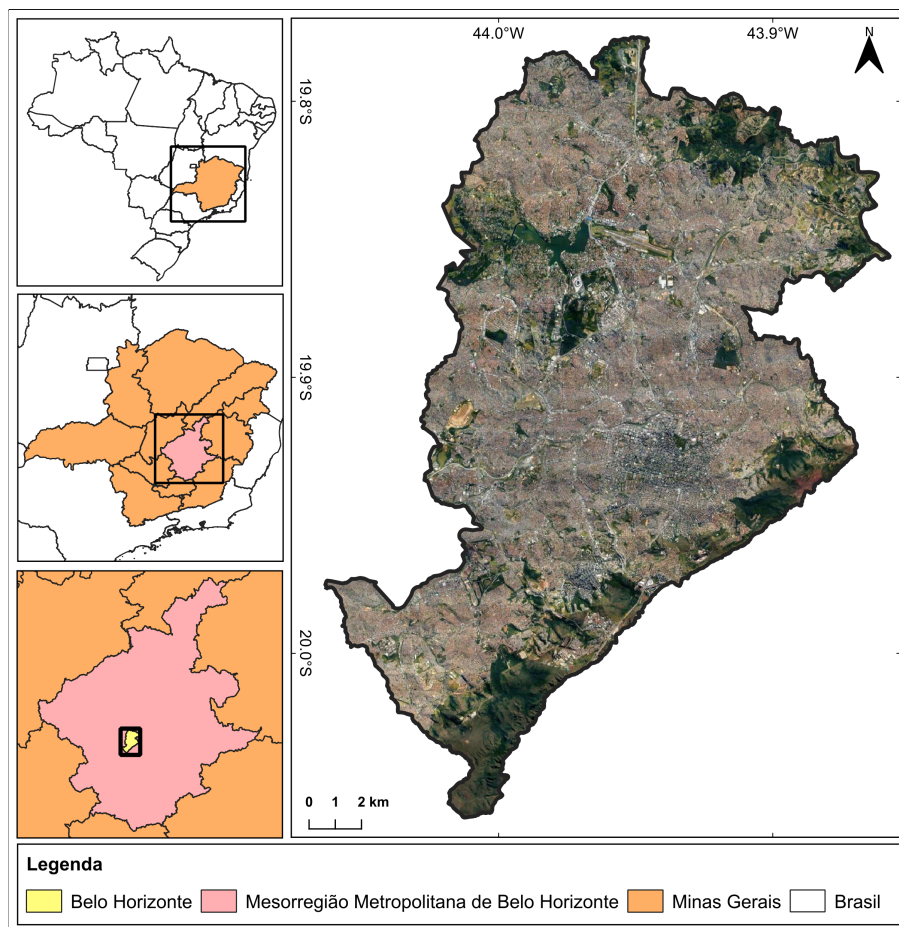
Puttemans et al. (2015) sugeriram, na conclusão de seu artigo, que uma das possibilidades de melhora na performance da detecção de UFV, seria o uso de detectores baseados em CNN como o trabalho de Girshick et al. (2014). A arquitetura sugerida pelos autores é a R-CNN, predecessora da Mask R-CNN, utilizada no presente trabalho.

4. MATERIAIS E MÉTODOS

4.1 Área de estudo

Este trabalho utilizou como recorte de estudo, o município de Belo Horizonte (Figura 29), capital do Estado de Minas Gerais, Brasil, caracterizado por uma área de 331,89 km² e população estimada de 2.512.070 habitantes para o ano de 2019 (IBGE, 2019), o que representa uma densidade populacional estimada de 7.568 hab./km². Belo Horizonte é o 6º município mais populoso e a 3ª região metropolitana mais populosa do Brasil, atrás de São Paulo e Rio de Janeiro (IBGE, 2019). Belo Horizonte possui 487 bairros e 9 regionais (PRODABEL, 2015). Belo Horizonte possui aproximadamente 740.000 edificações e sua área construída é de aproximadamente 297 km², 89% de sua área total (PRODABEL, 2015).

Figura 29 - Localização da área de estudo - Datum: WGS84 - Sistema de Coordenadas Geográficas



Fonte: Adaptado de Google Earth (2018)

4.2 Hardware utilizado

A obtenção e o processamento dos dados deste trabalho foram realizados em ambiente local de desenvolvimento, sendo utilizado computador para o treinamento e validação do modelo de segmentação com as seguintes especificações técnicas listadas no Quadro 1.

Quadro 1 - Especificações de *hardware* do ambiente de desenvolvimento

Descrição	Especificação
Memória (RAM)	32 GB
Processador (CPU)	Intel® Core™ i7-10700K CPU @ 3.80GHz × 16
Armazenamento	3 TB (SSD 480GB + SSD 480GB + HDD 2TB)
Sistema Operacional	Ubuntu 20.04.2 LTS
Arquitetura	64-bits
Placa Gráfica (GPU)	GeForce RTX™ 3060 Ti Ghost OC
Memória RAM (GPU)	8 GB GDDR6 (256 bits)
Velocidade do clock da memória (GPU)	7000Mhz (14Gbps)
Largura da Banda (GPU)	448 GB/s

Fonte: dados do próprio autor

Até dezembro de 2020, a metodologia desenvolvida utilizou o *Google Colaboratory*, que é um aplicativo em nuvem que oferece uma interface de execução de códigos Python, por meio de um ambiente de “*Notebooks Jupyter*” e permite o uso opcional de placas gráficas (GPU) gratuitamente, para treinamento de redes profundas. O *hardware* especificado no Quadro 1 foi adquirido com recursos próprios pelo autor, no final de dezembro de 2020, com o propósito de otimizar o fluxo de trabalho e facilitar o desenvolvimento e a manipulação dos dados, modelos e algoritmos aqui utilizados.

4.3 Softwares utilizados

Para as operações relacionadas às bases cartográficas vetoriais, confecção dos mapas e visualização dos dados obtidos optou-se pela utilização do *software* QGIS.

O QGIS é um ambiente SIG de código aberto e gratuito, que oferece funções de visualização, edição e gerenciamento de dados espaciais (QGIS, 2009). A versão do QGIS utilizada neste trabalho foi “3.18-Zurich”. Os rótulos de treinamento do modelo de classificação foram criados a partir de vetores desenhados no QGIS, em formato *geopackage* (.gpkg) e a partir de cada polígono criado, um *script* foi criado para operações com arquivos vetoriais.

Para a etapa de busca pelas imagens de melhor qualidade ofertadas pelo Google (2021), foi utilizado o aplicativo *desktop* Google Earth Pro, considerando que o software oferece os metadados (data e sensor de origem) das imagens.

A programação dos algoritmos utilizados nesta pesquisa foi feita em linguagem Python (versão 3.8.5).

A biblioteca *Geospatial Data Abstraction Library* (GDAL) é uma biblioteca de abstração de dados geoespaciais disponibilizada em linguagens de programação como C++ e Python (GDAL/OGR, 2019). GDAL contém funções que permitem a fácil manipulação de dados de formato raster nos mais diversos formatos como *Tagged Image File Format* (TIFF), *Georeferenced Tagged Image File Format* (GeoTIFF), *Joint Photographic Experts Group* (JPEG) e *Portable Network Graphics* (PNG). Também nesta biblioteca, GDAL disponibiliza um pacote de funções de manipulação de dados vetoriais chamada *OpenGIS Simple Features Reference Implementation* (OGR) e também foi utilizado em etapas da metodologia proposta. Dessa maneira, foram integrados programas de Interface de Linha de Comando (CLI) do GDAL/OGR e também da biblioteca nativa Python do GDAL/OGR à interface dos algoritmos usados.

A arquitetura de segmentação de instâncias usada foi a Mask R-CNN (HE et al., 2017), implementada dentro do software Detectron2 com pesos pré-treinados da base de dados COCO.

4.4 Dados utilizados

Para que se conecte à rede *on-grid*¹⁸ de geração fotovoltaica é necessária a homologação da UFV junto à Agência Nacional de Energia Elétrica (ANEEL). Ao homologar e regularizar uma UFV junto a esta agência reguladora, os dados da geração de cada UFV são disponibilizados pela ANEEL em seu website.

Para coletar as informações das UFV homologadas na ANEEL, foi elaborado um *script* Python para extraí-las, filtrando os dados apenas do município “Belo Horizonte” e também do tipo de geração “UFV”. Estes dados foram coletados em 14 de março de 2020.

Este *script* de extração foi construído de modo a obter pontos de quatro municípios, a saber: Belo Horizonte/MG, Brasília/DF, Rio de Janeiro/RJ e Campinas/SP. As razões da escolha por estes municípios específicos estão detalhadas na seção a seguir.

Esta extração resultou em um arquivo em formato vetorial GeoJSON, já com todos os pontos geolocalizados, usando o datum WGS84 no Sistema de Coordenadas Geográficas.

Google *Maps Platform* (GMP) é um dos produtos Google contendo serviços e ferramentas de dados geoespaciais, mapas, imagens orbitais oferecidas gratuitamente para propósitos não-comerciais e outros. Estes serviços são disponibilizados em diferentes interfaces como o Google Earth (aplicativo desktop e web de imagens da Terra em formato 2D e 3D), Maps Javascript API (interface em linguagem de programação Javascript, para incorporar mapas em páginas da web), Google Maps (banco de dados geográficos com uma ampla variedade de dados espaciais de todo planeta, incluindo informações de trânsito, edificações urbanas, mapas segmentados de áreas naturais e etc.), e outros. Dentro do Google Maps, é possível fazer requisições a um de seus canais chamado *Maps URLs* (GOOGLE, 2019) que a partir de parâmetros definidos, é possível obter como resposta, partes de um mosaico, composto por imagens orbitais e ortofotos em algumas localizações específicas. As imagens fornecidas por esta interface nem sempre são ortofotos mas

¹⁸ *On-grid* se refere aos UFV que estão ligados à rede de energia das concessionárias como a Companhia Energética de Minas Gerais (CEMIG) que atende o estado de Minas Gerais, por exemplo.

há locais com cobertura destes produtos. Além do *Maps URLs*, o GMP também oferece um serviço chamado *Maps Static API* (MSA) que tem como objetivo a obtenção de imagens de satélite e mapas temáticos para serem consumidos dentro de outros produtos Google como o *Maps Javascript API*. O MSA retorna uma imagem (.PNG ou .JPEG) em resposta a uma requisição de protocolo HTTP. Para cada requisição é possível solicitar diversos parâmetros de retorno como resolução da imagem, escala, coordenadas do ponto central em que se deseja a imagem, nível de *zoom*, cores das edificações e outras modalidades de mapas temáticos (GOOGLE, 2019).

A origem dos dados de saída tanto do *Maps URLs*, quanto do MSA é a mesma, um mosaico global de imagens cobrindo 98% da superfície terrestre composto por diversos produtos mesclados, de diferentes sensores orbitais. Estes produtos são adquiridos comercialmente pelo Google de empresas como a *Maxar Technologies Inc*, *Planet Labs* e outras. O que muda entre os dois métodos é a forma de acesso/extração dos dados e suas dimensões de saída, mas se tratam de recortes do mesmo mosaico de imagens. Ambos métodos fornecem imagens em formato RGB.

A Figura 30 apresenta a diferença no retorno das requisições feitas à interface *Maps URLs* (Figura 30 à esquerda) e MSA (Figura 30 à direita). A primeira¹⁹ possui 256x256 pixels, a segunda²⁰ possui 600x600 pixels (que pode ser alterada mudando os parâmetros de solicitação à API).

¹⁹ Requisição feita utilizando parâmetros como parâmetros (lyrs=s, x=396250, y=583338 e z=20). A requisição completa é composta pelo *link* <http://mt0.google.com/vt/lyrs=s&x=396250&y=583338&z=20> (Coordenadas dadas em metros para o datum WGS84 / Pseudo-Mercator - EPSG:3857)

²⁰ Requisição feita utilizando parâmetros como parâmetros (center=-19.8631054,-43.9581906, zoom=20, size=600x600, maptype=satellite e key=Chave_da_API oculta por razões de proteção aos dados) A requisição completa é composta pelo *link* https://maps.googleapis.com/maps/api/staticmap?center=-19.8631054,-43.9581906&zoom=20&size=600x600&maptype=satellite&key=API_KEY (Coordenadas dadas no sistema de coordenadas geográficas para o datum WGS84 - EPSG:4326)

Figura 30 - Exemplo de retorno de uma requisição de imagem de *Maps URLs* (à esquerda) e MSA (à direita)



Fonte: Google (2021) e Google (2019)

Cada requisição feita à MSA é cobrada como um serviço (aproximadamente US\$0,002/imagem), e exatamente por isso possui uma maior flexibilidade no uso de diferentes parâmetros (resolução de saída de no mínimo 2x2 pixels até no máximo 640x640 pixels por imagem), entretanto, sua utilização é mais comum em aplicações comerciais. Todas imagens obtidas pelo MSA vêm com uma marca d'água que pode ser observada na Figura 30 (à direita).

Já as requisições feitas ao *Maps URLs* são gratuitas, apesar de possuir parâmetros mais restritos (apenas parâmetros de latitude, longitude e *zoom*, resolução fixa de 256x256 pixels²¹) e uso limitado (máximo de 50 requisições por segundo ou 5000 requisições diárias). A utilização do Maps URLs geralmente é embutida em aplicações web gratuitas.

Para determinar quais municípios seriam usados como dados de treinamento do modelo, foi feita uma inspeção visual, usando o Google Earth Pro, aferindo a qualidade das imagens de cada um dos dez municípios mais populosos do Brasil, segundo a classificação do IBGE (2020). As características investigadas nestas imagens foram a resolução espacial (tamanho do pixel) e presença de ortorretificação explícita (buscando a ausência de distorções em edificações de grande porte).

²¹ Cada imagem de 256x256 pixels do *Maps URLs* é chamada de *tile* (telha)

Com isto, pretendeu-se selecionar os três municípios com imagens de melhor qualidade disponíveis, exceto de Belo Horizonte, por se tratar da área de estudo e inferência do modelo. As imagens de Belo Horizonte foram usadas como referência de qualidade, pois uma vez que o modelo é aplicado nesta área, os dados de treinamento não podem possuir características muito distintas em termos de qualidade se comparados com o conjunto de dados onde a inferência é aplicada.

A Tabela 1 apresenta a lista dos municípios inspecionados.

Tabela 1 - Lista dos dez municípios mais populosos do Brasil

Posição	Município	Unidade Federativa	População (hab.)
1	São Paulo	São Paulo	12.325.232
2	Rio de Janeiro	Rio de Janeiro	6.747.815
3	Brasília	Distrito Federal	3.055.149
4	Salvador	Bahia	2.886.698
5	Fortaleza	Ceará	2.686.612
6	Belo Horizonte	Minas Gerais	2.521.564
7	Manaus	Amazonas	2.219.580
8	Curitiba	Paraná	1.948.626
9	Recife	Pernambuco	1.653.461
10	Goiânia	Goiás	1.536.097

Fonte: IBGE (2020)

A Figura 31 mostra exemplos das qualidades encontradas nas imagens das nove cidades avaliadas.

Figura 31 - Exemplos de qualidade das imagens disponibilizadas para os dez municípios mais populosos do Brasil, exceto Belo Horizonte/MG com escala de 1:500



Fonte: Google (2021)

Nesta etapa de inspeção, percebeu-se que as imagens com resolução espacial de 7 cm/pixel em média, eram as imagens fornecidas por Maxar Technologies Inc. Das nove cidades inspecionadas, apenas Brasília/DF e Rio de Janeiro/RJ obtiveram qualidade semelhante à qualidade das imagens de Belo Horizonte/MG. Ao avaliar os metadados de cada um dos locais, constatou-se que o elemento comum entre as imagens dos três municípios era o sensor de origem. Estas cidades eram cobertas por sensores da Maxar Technologies Inc (o Google Earth Pro não informa, além da data da imagem e empresa fornecedora, nenhum metadado adicional destas imagens).

Então, considerando que só haviam dois municípios com boa qualidade de imagens entre os dez listados, buscou-se o próximo município mais populoso e que também recebe a cobertura dos sensores da Maxar Technologies Inc. O município que se encaixou nestas especificações foi Campinas/SP, o 14º município mais populoso do Brasil com 1.213.792 habitantes.

Ao avaliar a qualidade das imagens que têm a Maxar Technologies Inc como fonte, percebeu-se que elas possuem características correlatas à ortofotos provenientes de fotografias aéreas ortorretificadas, e apesar de não ser explicitamente dito por Google em seus metadados, considerou-se as imagens fornecidas por eles como sendo ortofotos. Portanto, com esta etapa realizada, ficaram definidos Rio de Janeiro/RJ, Brasília/DF e Campinas/SP como os locais de onde foram extraídas as imagens de UFV para treinar o modelo proposto.

Duas formas distintas de extração de imagens do Google foram utilizadas em etapas diferentes, equilibrando seus respectivos *tradeoffs*.

Para extrair as imagens de treinamento, utilizou-se o MSA com a localização geográfica (latitude e longitude) de pontos aleatórios no entorno de feições representativas das classes anotadas, gerando para cada um desses pontos, uma requisição ao MSA com parâmetros de *zoom* valor 21 (máximo de resolução oferecida pelo MSA), latitude e longitude destes pontos e tamanho de imagem de 600x600 pixels.

Já para a etapa de inferência do modelo, foram extraídas imagens de toda a extensão da área de estudo, optou-se por gerar uma composição de rasters de dimensão $0,01^\circ \times 0,01^\circ$ ⁽²²⁾ extraídos do *Maps URLs*. Esta dimensão foi definida de modo que cada raster do mosaico ocupasse, no máximo, 750 *megabytes* (MB) de uso de disco. Para realizar estas requisições, foi utilizada a biblioteca Python “*requests*” para comunicação com a API do MSA e *Maps URLs*.

A resolução espacial das imagens Google que compuseram a ortofoto da área de estudo é de 7 cm, datadas de 16/05/2018, e são providas por Maxar Technologies e servidas pelo Google na interface do MSA. Já a resolução espacial das imagens Google nos três municípios de treinamento é de 7 cm, datadas de 01/07/2018 (Rio de Janeiro/RJ), 19/01/2018 (Brasília/DF) e 09/06/2018

²² Aproximadamente 1050x1050 metros.

(Campinas/SP), Sendo providas por Maxar Technologies e servidas pelo Google por meio do *Maps URLs*.

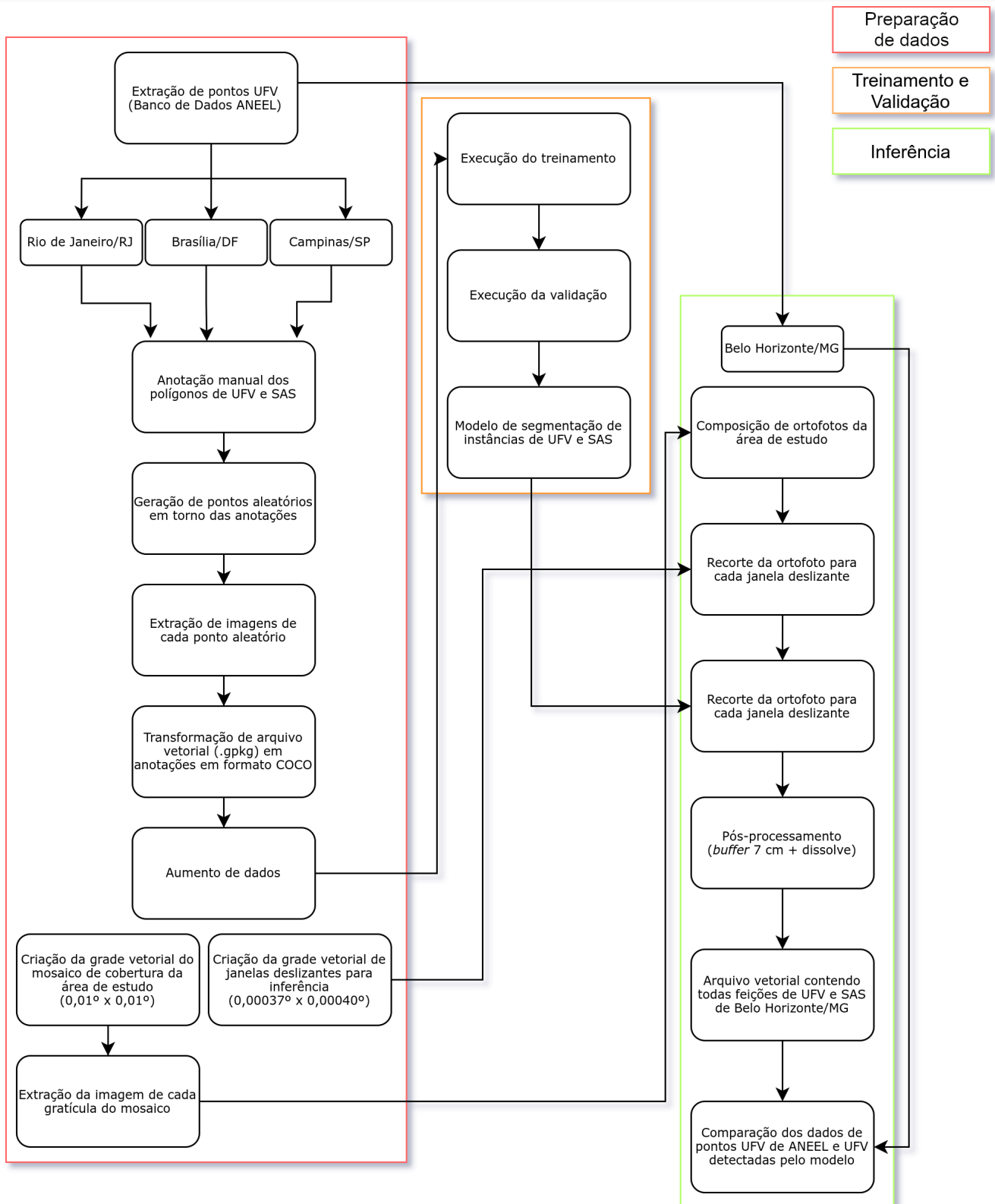
Foram utilizados também os arquivos vetoriais disponibilizados pela Empresa de Informática e Informação do Município de Belo Horizonte (PRODABEL, 2015) contendo os limites municipais, bairros e localização das edificações de Belo Horizonte. A localização das edificações serviram para auxiliar na criação do *grid* de rasters compondo a cobertura de ortofotos da cidade, ou seja, locais sem edificações não foram segmentados pelo modelo para poupar recursos computacionais e diminuir o tempo de inferência.

Todos os procedimentos realizados nesta metodologia foram utilizando o datum “WGS84” (EPSG:4326), utilizando o Sistema de Coordenadas Geográficas, compatível com o datum oficial brasileiro SIRGAS 2000. Esta opção se deu por conta dos dados da ANEEL e do Google serem fornecidos neste datum.

4.5 Metodologia

O plano de trabalho desta pesquisa está representado no fluxograma da Figura 32, cada uma dessas etapas estão descritas com detalhes, nas seções seguintes.

Figura 32 - Fluxograma da metodologia do trabalho



Fonte: Elaboração do autor

A primeira etapa da metodologia foi realizada com um *script* Python de extração dos dados geográficos de geração distribuída do banco de dados da ANEEL. Considerando as datas das imagens que foram usadas em cada um dos municípios, após a extração dos pontos UFV da ANEEL, foram filtrados e removidos os pontos com datas posteriores à imagem de seus respectivos municípios. Como as imagens usadas na pesquisa não poderiam possuir UFV homologados em datas posteriores à própria data das imagens, esta remoção foi feita de modo a diminuir a quantidade de pontos analisados na busca ativa pelo objeto-alvo. A Tabela 2 resume os dados encontrados para cada local, e, demonstra o número de UFV homologados na ANEEL até o dia 14/03/2021, informação que confirma os dados expostos na seção 1.1 onde afirma-se que o crescimento desses sistemas ocorre de maneira muito rápida.

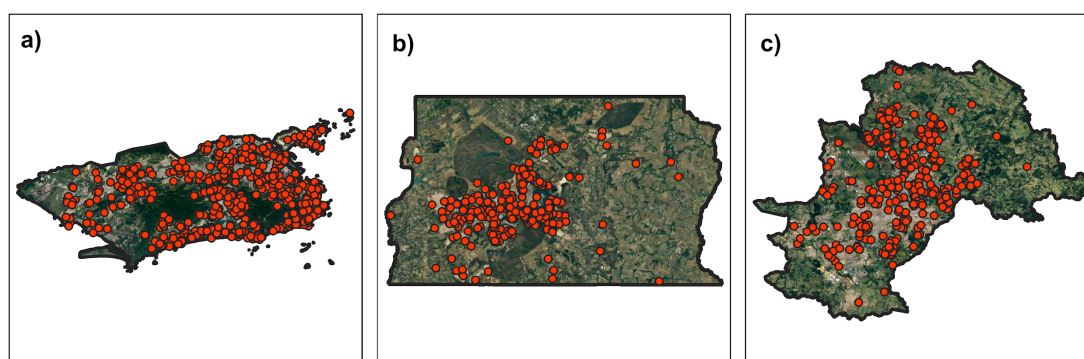
Tabela 2 - Resumo dos dados extraídos de ANEEL contendo pontos UFV

Município	Pontos UFV até a 01/07/2018	Pontos UFV em 14/03/2021
Rio de Janeiro/RJ	957	5081
Brasília/DF	333	2974
Campinas/SP	645	2122
TOTAL	1935	10177

Fonte: ANEEL (2021)

A Figura 33 apresenta a distribuição espacial dos pontos UFV em cada município.

Figura 33 - Distribuição espacial dos pontos UFV (em vermelho) extraídos de ANEEL em (a) Rio de Janeiro/RJ, (b) Brasília/DF e (c) Campinas/SP



Fonte: Google (2018) e ANEEL (2020)

Os pontos da ANEEL foram utilizados como base para a busca visual nos arredores de cada ponto. Dentro do QGIS, empregou-se a função “XYZ Tiles” do *software*, a qual cria pirâmides de *tiles* do *Maps URLs* e permite a visualização dessas imagens com os pontos sobrepostos à elas.

O próximo passo foi criar o arquivo vetorial das anotações (valores de referência), onde foi criado em formato *Geopackage* (.gpkg), o qual corresponde ao formato nativo de dados vetoriais do QGIS e sendo criada uma coluna denominada “classe”. Assim, foram inspecionados os 1935 pontos extraídos da ANEEL nos três municípios. Esta inspeção foi realizada da seguinte maneira: cada um dos pontos foi aproximado no ambiente do sistema aproximando-se para a escala de visualização 1:500, sendo toda as feições visíveis de UFV e SAS foram desenhadas manualmente, criando um polígono delimitando estas feições e atribuindo o valor de classe “UFV” ou “SAS” na tabela de atributos do vetor criado. É importante ressaltar que não foi realizada a busca ativa por SAS, sendo apenas realizadas as anotações dos polígonos de SAS quando estes estavam presentes nas mesmas cenas que os UFV. A Figura 34 demonstra dois exemplos deste procedimento.

Para o processo de diferenciação das anotações manuais das classes alvo, ou seja, interpretar se um painel presente nos telhados pertence à classe UFV ou SAS, foram consideradas as características principalmente de cor e tamanho dos painéis fotovoltaicos. Os UFV em geral são majoritariamente azuis e também estão dispostos em composições maiores do que os painéis de SAS.

Figura 34 - Exemplo do procedimento de anotação dos dados de referência



Fonte: Adaptado de Google Earth (2018)

Depois da anotação de todas as feições de UFV e SAS encontrados no procedimento descrito acima, o arquivo vetorial com as anotações passou por um processamento por meio do algoritmo de geração de pontos aleatórios no entorno de cada uma das feições anotadas. Estes pontos aleatórios foram criados a partir de um *script* em Python, usando a biblioteca OGR (GDAL/OGR, 2019) na qual se aplica um *buffer zone* (zona tampão) de 0.0001° (aprox. 10 metros) em todas as feições de classe UFV (Figura 35).

Figura 35 - Exemplo da geração de pontos aleatórios em torno das anotações



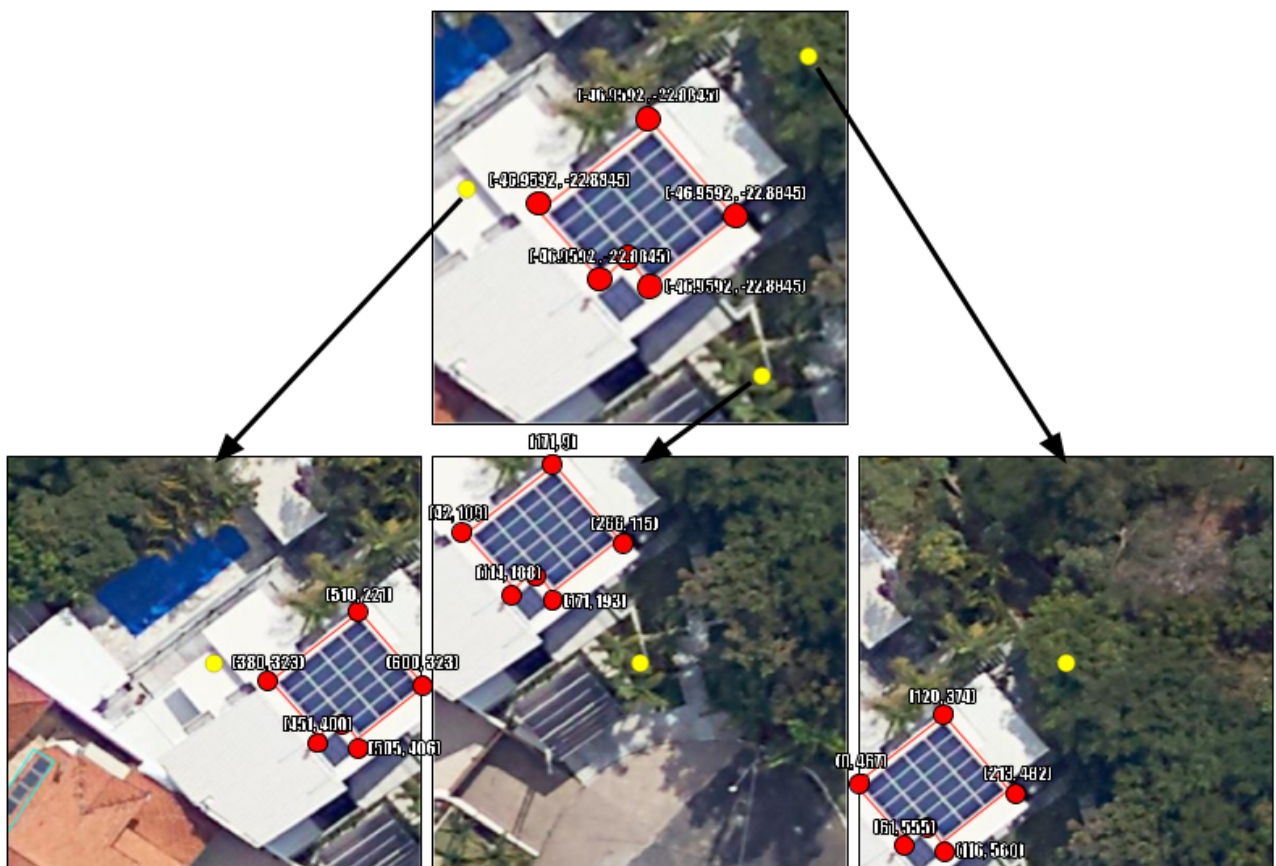
Fonte: Adaptado de Google Earth (2018)

Este procedimento se justifica por ser uma maneira de aumentar os dados disponíveis de classificação com uma mesma anotação poligonal, cada um dos pontos gera imagens anotadas com dimensões e características diferentes. A escolha pelo valor de três pontos foi arbitrária, mas levando em conta que cada anotação gera três pontos aleatórios, sendo que valores maiores do que três configuram custo maior na etapa de extração das imagens, já que cada requisição à MSA tem um valor fixo (US\$0,002/imagem). Além disso, o aumento destes pontos também aumenta o custo monetário da geração dos dados.

A próxima etapa consistiu em realizar a extração das imagens para cada um dos pontos aleatórios gerados segundo o procedimento descrito acima. Para isso, um *script* Python foi criado para percorrer todos os pontos aleatórios, usando suas

coordenadas como parâmetros para uma requisição ao MSA, solicitando uma imagem de dimensão 600x600 pixels de cada ponto gerado. Dessa maneira, para cada imagem solicitada, o *script* traduziu as coordenadas geográficas do arquivo de anotações para suas correspondentes coordenadas relativas à imagem requisitada. A Figura 36 mostra um exemplo deste procedimento de conversão, onde os polígonos dos valores de referências (Figura 36, detalhe acima) são convertidos para as anotações em formato COCO (Figura 36, três figuras no detalhe abaixo), gerando, portanto, para cada ponto aleatório, uma amostra de treinamento diferente.

Figura 36 - Exemplo de conversão de anotações em formato vetorial para anotações formato COCO



Fonte: Adaptado de Google Earth (2018)

Com as imagens extraídas de MSA e suas respectivas anotações COCO convertidas, a próxima etapa metodológica foi a aplicação do aumento de dados a todas as imagens de amostra. As técnicas de aumento de dados selecionadas e aplicadas às imagens de treinamento estão expostas no Quadro 2.

Quadro 2 - Técnicas de aumento de dados aplicadas à todas as imagens

Nº	Procedimento	Procedimento Aleatório
1	Rotação 90°	Borrão gaussiano; ou Borrão médio; ou Borrão mediano; ou Adição de ruído gaussiano; ou Variação de claridade; ou Aplicação de contraste linear.
2	Rotação 180°	
3	Rotação 270°	
4	Espelhamento Horizontal	
5	Rotação 90° + Espelhamento Horizontal	
6	Rotação 180° + Espelhamento Horizontal	
7	Rotação 270° + Espelhamento Horizontal	

Fonte: Próprio autor (2021)

Todas as imagens amostradas receberam cada um dos procedimentos de aumento de dados listados no Quadro 2, sendo que a cada um dos sete procedimentos, se aplicou apenas a um dos procedimentos aleatórios destacados na coluna “Procedimento Aleatório”. Isso significa que para cada imagem de treinamento, gerada a partir dos pontos aleatórios, foram geradas sete novas imagens transformadas.

Para este procedimento foi utilizada a biblioteca Python de aumento de dados chamada “*imgaug*” (JUNG et al., 2019) que além das transformações nas imagens, também executa a alteração das anotações para que as anotações permaneçam correspondendo aos valores de referência.

A Figura 37 apresenta um exemplo de uma imagem das amostras e as sete transformações aplicadas à ela.

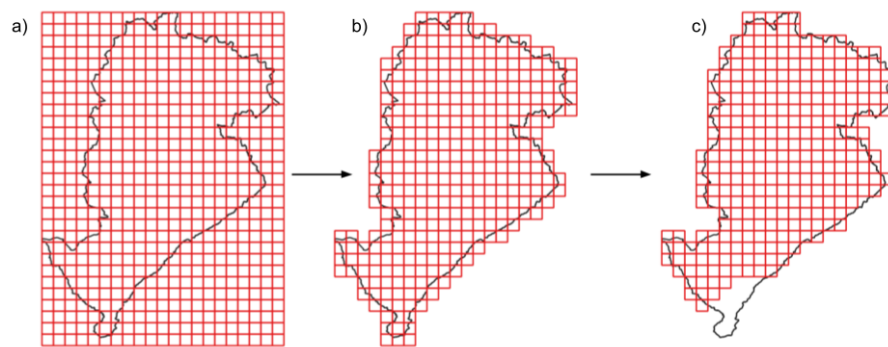
Figura 37 - Aplicação dos sete procedimentos de aumento de dados em uma imagem das amostras



Fonte: Adaptado de Google Earth (2018)

A última parte da preparação dos dados consistiu na construção do mosaico *raster* com a ortofoto da área de estudo. Primeiro, foram criadas camadas vetoriais para uso na extração do mosaico raster da área de estudo. Para isso, utilizou-se a base vetorial disponibilizada por Prodabel (2015) contendo o limite municipal de Belo Horizonte/MG. Com este limite, foi gerado um *grid* utilizando-se a ferramenta “Criar grade” do QGIS, inserindo $0,01^\circ$ (1050 metros) como parâmetro de espaçamento horizontal e espaçamento vertical. Foram gerados 609 *grids* (Figura 38a), mas como esta ferramenta gera feições em todo o retângulo envolvente do município, foram filtrados somente os *grids* que tinham interseção com o limite municipal de Belo Horizonte. Este filtro reduziu as grades para 352 (Figura 38b). Mas, considerando que há partes da cidade que não possuem edificações e portanto não poderiam conter UFV ou SAS, foram eliminadas as grades que não continham interseção com nenhuma das edificações contidas na base vetorial de edificações da Prodabel. Finalmente, foram obtidos 306 *grids* (Figura 38c) para a extração do mosaico *raster* das ortofotos.

Figura 38 - Processo de criação do *grid* vetorial de $0,01^\circ \times 0,01^\circ$ para composição do mosaico de ortofotos

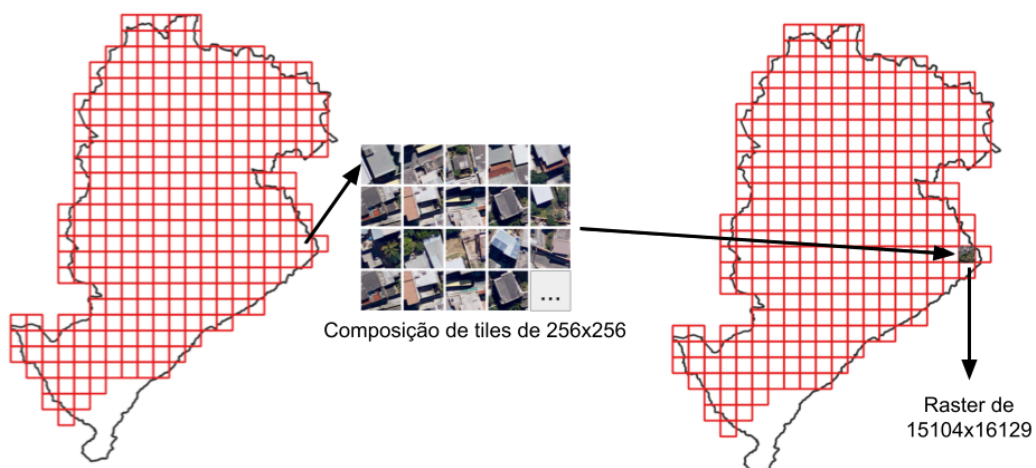


Fonte: Próprio autor (2021)

Com os 306 *grids* preparados, elaborou-se um *script* Python para gerar arquivos formato TIFF a partir dos *tiles* disponibilizados no *Maps URLs*. Para cada *grid* de $0,01^\circ \times 0,01^\circ$ (Figura 38) foram solicitadas à API do *Maps URLs* os *tiles* internos ao *grid*. Como já descrito anteriormente, cada *tile* de retorno desta API, possui dimensão fixa de 256x256.

A Figura 39 apresenta visualmente o esquema do procedimento supramencionado onde foram extraídas os *tiles* de cada *grid* (detalhe em vermelho), resultando em arquivos *raster* (formato TIFF), que foram baixados para o ambiente local de desenvolvimento. Estes arquivos resultantes são as ortofotos que foram segmentadas em busca dos alvos UFV e SAS.

Figura 39 - Exemplo da geração de um dos componentes do mosaico *raster* a partir de dados extraídos do *Maps URLs*



Fonte: Adaptado de Google (2021)

Os *tiles* da API do *Maps URL* são entregues em formato PNG, ou seja, não são georreferenciados. Portanto, o *script* foi preparado para georreferenciar cada um dos *tiles*, calculando a partir do pixel central (que tem-se a informação da coordenada) as coordenadas dos quatro cantos do *tile* e transformando o arquivo PNG em um TIFF devidamente georreferenciado. Daí, todos os *tiles* baixados foram mesclados utilizando-se o algoritmo disponibilizado no GDAL denominado “*gdal_merge.py*”.

Adicionalmente, ainda foi preciso contornar o problema do limite diário de requisições ao *Maps URLs*, já que uma máquina (identificada por seu IP) só pode requisitar 5000 *tiles* por dia²³ e a área de estudo é composta por 1.137.402 *tiles* (3717 *tiles/grid* x 306 *grids*) de 256x256 pixels. Para isto, foi configurada uma rede temporária de servidores em nuvem, no provedor *DigitalOcean*²⁴. Um servidor²⁵ neste provedor tem o custo de US\$0,00744/hora e foi escolhido exatamente pelo fato da cobrança pelo uso ser feita pelas horas utilizadas e cada um dos servidores possuir um IP diferente. Portanto, para cada graticula do mosaico, abriu-se um servidor temporário que extraiu *tile a tile* do *Maps URLs*. Foram abertos 306 servidores temporários e o procedimento de extração levou 30 minutos por servidor, em média, o que significou um gasto de US\$1,15 para a execução desta tarefa.

Além do *grid* vetorial de extração do mosaico, foram criadas janelas deslizantes. As janelas deslizantes são os vetores usados para cortar o mosaico de ortofotos em imagens de 600x600 pixels.

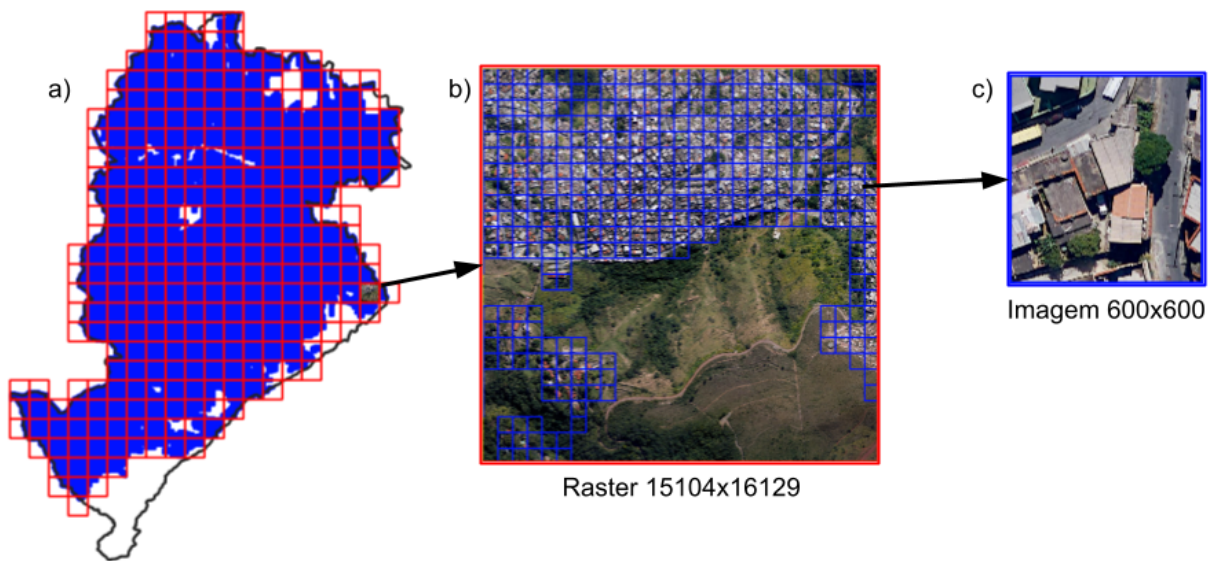
Então, para a criação destas janelas utilizou-se novamente a ferramenta “Criar grade” do QGIS, inserindo 0,00037° (38 metros) como parâmetro de espaçamento horizontal e 0,00040° (44 metros) para espaçamento vertical. As janelas deslizantes foram filtradas, removendo todas as janelas que cobriam áreas sem nenhuma edificação. A Figura 40 apresenta toda a área de cobertura do vetor de janelas deslizantes.

²³ Restrição imposta pelos servidores Google.

²⁴ Disponível em: <<https://www.digitalocean.com/>>

²⁵ O serviço de hospedagem em nuvem oferecido pela DigitalOcean é chamado de “*droplet*”.

Figura 40 - Área de cobertura das janelas deslizantes (azul) na área de estudo e extração de imagem de 600x600 a partir de uma janela deslizante



Fonte: Adaptado de Google Earth (2021)

A criação do vetor de janelas deslizantes é o último passo da preparação de dados e com a preparação dos dados concluída, a etapa de treinamento e validação foi feita inserindo as imagens e anotações de treinamento e validação no devido diretório do Detectron2. Depois foi necessário ajustar as condições de treinamento do modelo, dentro do *framework*.

Fazendo uso da técnica de transferência de aprendizado (conforme descrito na seção 3.4), utilizou-se o modelo de pesos pré-treinados “*mask_rcnn_R_50_FPN_3x*²⁶”, que foi treinado no banco de imagens *ImageNet*. Foram adotados os parâmetros padrão oferecidos no Detectron2, e o treinamento foi realizado em 1500 iterações (15 épocas).

A validação foi realizada em seguida, pelo próprio Detectron2, resultando em métricas no formato COCO. Assim como exposto em He et al. (2017), para a validação, as métricas são implementadas por padrão no Detectron2 e os resultados de saída são dados em formato COCO. Portanto, foram utilizadas as métricas de AP, AP50, AP75, AP_{peq}, AP_{méd} e AR.

²⁶ Pesos pré-treinados com uma programação de aprendizagem de 3x (~37 épocas COCO), usando a rede ResNet-50 que é uma CNN de 50 camadas de profundidade, combinada com FPN (ver seção 3.3) e que possui o melhor *tradeoff* entre velocidade de inferência e acurácia, das redes disponíveis em Detectron2.

O treinamento do modelo resultou em um arquivo de pesos calibrados (formato “.pth”, associado ao *framework* PyTorch) que respondeu à tarefa de detectar UFV e SAS em uma imagem. O termo “modelo treinado”, utilizado durante o texto, se refere exatamente a estes pesos calibrados.

A última etapa da metodologia proposta consistiu em executar uma varredura em todas as imagens geradas pelo corte das ortofotos por cada uma das janelas deslizantes, com o emprego do modelo treinado. À este procedimento, dá-se o nome de inferência, terminologia usada pela literatura e na documentação do Detectron2 para aplicação do modelo treinado à imagens de entrada.

Dessa maneira, cada imagem gerada pelas janelas deslizantes possuem uma dimensão de 600x600 (Figura 40). Para extrair cada um desses recortes de 600x600 pixels a partir das ortofotos, foi elaborado um *script* Python que executou o seguinte procedimento:

1. Carregamento do arquivo vetorial de janelas deslizantes;
2. Carregamento do diretório contendo as 306 ortofotos;
3. Para cada janela deslizante:
 - a) Identificação de qual das 306 ortofotos possui área de interseção com a janela deslizante;
 - b) Para as ortofotos com área de interseção com a janela, extrair toda a extensão da janela em um arquivo de formato PNG, usando a ferramenta “gdal_translate” (GDAL/OGR, 2019);
 - c) Fornecimento ao Detectron2, como imagem de entrada a ser segmentada, a imagem obtida no passo 3b;
 - d) Caso a imagem possua algum objeto identificado (UFV ou SAS) com um *score* (valor de confiança do modelo) maior que 0,5 (*score threshold*), transformar a máscara de saída de cada instância, em um arquivo TIFF, usando a extensão da janela deslizante como pontos de controle para georreferenciar a imagem;
 - e) Vetorizar o arquivo TIFF (máscara binária) do passo 3d usando a ferramenta “gdal_polygonize” (GDAL/OGR, 2019) e aplicar a simplificação dos vértices em 0,000001° (2 cm) para suavizar o serrilhamento da segmentação;

- f) Vincular ao polígono gerado no passo 3e, o nome da classe (UFV ou SAS) que foi identificada na máscara e seu respectivo *score*.
- g) Adicionar o polígono com a classe e *score* ao arquivo final de segmentação de instâncias.

O resultado deste procedimento foi um único arquivo contendo todas as instâncias de UFV e SAS identificadas pelo modelo, após a varredura em cada janela deslizante nas ortofotos. Este arquivo contém em cada uma de suas feições, sua classe (UFV ou SAS) e *score*²⁷ (0,5 à 1).

Por fim, também foi feita uma etapa de pós-processamento do arquivo de segmentações, para corrigir algumas inconsistências topológicas e mitigar o efeito de borda das janelas deslizantes na segmentação de instâncias. Para tanto, utilizando o QGIS, foi aplicado à estes vetores o procedimento de zona *buffer* de 7 cm (tamanho do pixel dos dados de entrada) e o procedimento de dissolver as feições por classe, para que feições sobrepostas fossem mescladas em um único vetor, de forma que não houvesse um mesmo objeto classificado duas vezes com o mesmo rótulo.

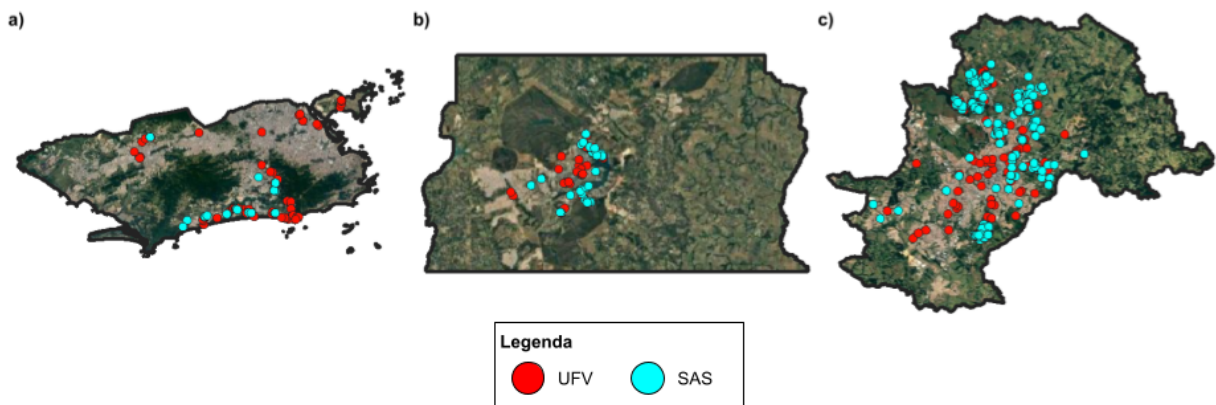
²⁷ Probabilidade de 0 a 1 do objeto pertencer à classe-alvo.

5. RESULTADOS

5.1 Preparação de dados

A etapa de extração dos dados de pontos UFV da ANEEL resultou em 957 pontos para o Rio de Janeiro/RJ, 333 pontos para Brasília/DF e 645 pontos para Campinas/SP, totalizando 1935 pontos UFV que serviram como referência para as anotações dos dados de treinamento. Executando o procedimento de aproximar cada um dos 1935 pontos com escala de 1:500, foram anotados todos os objetos UFV e SAS e este passo foi realizado em 26 horas no total, divididos em três dias de trabalho. A Figura 41 mostra a localização das anotações em cada município.

Figura 41 - Localização das anotações UFV e SAS no Rio de Janeiro/RJ (a), Brasília/DF (b) e Campinas/SP (c)



Fonte: Adaptado de Google Earth (2018)

Foram anotadas 1013 feições UFV no total, sendo 164, 171 e 678 para os municípios do Rio de Janeiro/RJ, Brasília/DF e Campinas/SP, respectivamente. Já para a classe SAS, foram anotadas 706 feições no total, sendo 22, 65 e 619 para os municípios do Rio de Janeiro/RJ, Brasília/DF e Campinas/SP, respectivamente (Tabela 3).

Em praticamente todos os pontos da ANEEL que foram percorridos, e que haviam UFV na cena, percebeu-se um deslocamento entre o telhado com o módulo fotovoltaico e o ponto geográfico ofertado pela ANEEL. Supõe-se que isso acontece

pelo fato da ANEEL alimentar seu banco de dados com as coordenadas geográficas do endereço do proprietário, que resulta neste deslocamento e imprecisão do ponto.

Com os vetores das anotações de treinamento prontos, o *script* de percorrer cada uma das feições, distribuindo pontos aleatórios no entorno dos polígonos foi executado. Foram gerados 5157 pontos aleatórios, distribuídos ao redor das anotações.

Cada coordenada geográfica dos pontos aleatórios, foi usada para extrair as imagens do MSA e receber sua respectiva anotação no formato COCO, conforme descrito na seção 4.5. Esta etapa levou 2:12hs para ser executada. Obteve-se, portanto, 3666 imagens (3090 de treinamento e 576 de validação).

Com as imagens extraídas do MSA, o próximo passo foi usá-las no procedimento de aumento de dados. Cada imagem recebeu sete tratamentos de aumento de dados (Quadro 2), aplicadas a 5157 imagens, resultando em 41256 imagens anotadas para o treinamento do modelo.

Por fim, as amostras anotadas foram separadas em 80% para treinamento (33005 imagens) e 20% para validação (8251 imagens). Estas informações estão resumidas na Tabela 3.

A separação foi feita usando um *script* Python que separou aleatoriamente as amostras em seus devidos diretórios.

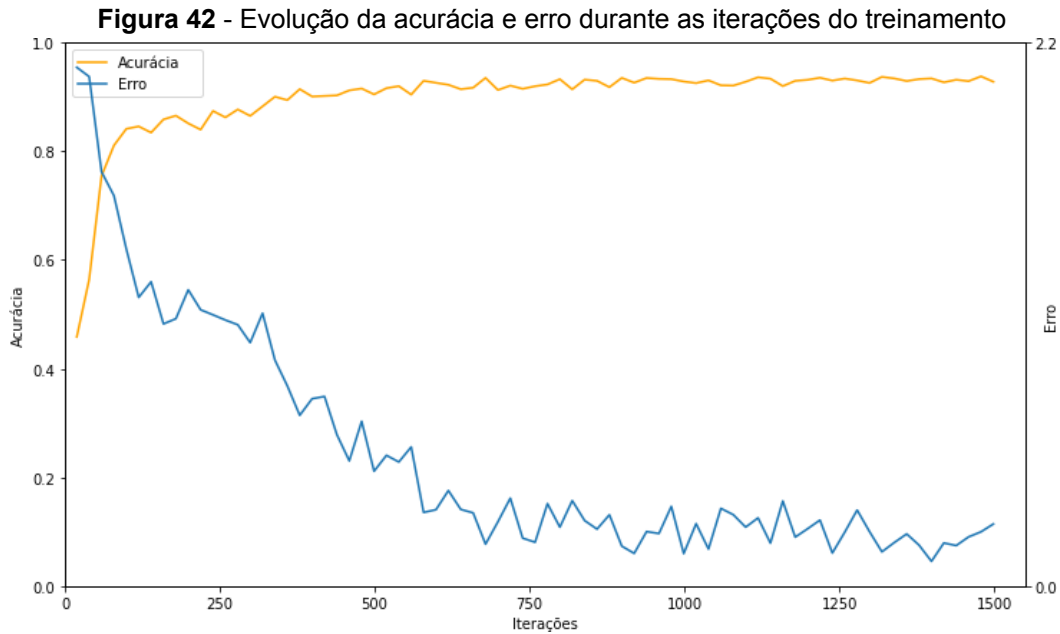
Tabela 3 - Número de feições anotadas para treinamento

Município	UFV	SAS	UFV (pontos aleatórios)	SAS (pontos aleatórios)	UFV (dados aumentados)	SAS (dados aumentados)
Brasília/DF	171	65	513	195	4104	1560
Campinas/SP	678	619	2034	1857	16272	14856
Rio de Janeiro/RJ	164	22	492	66	3936	528
TOTAL	1013	706	3039	2118	24312	16944

Fonte: Próprio autor (2021)

5.2 Treinamento e validação

Foi feita a execução do treinamento que durou um tempo total de 6 minutos e 30 segundos para as 1500 iterações (Figura 42).



A validação resultou nas métricas finais e a Tabela 4 demonstra os resultados de AP e AR, sendo discriminados também, seus valores para tamanhos diferentes de objetos, bem como a AP de cada classe. As discussões sobre estes resultados estão no capítulo 6.

Tabela 4 - Métricas de validação do modelo treinado para diferentes tamanhos de objetos

Nº Métrica	IoU	Área	Objeto	Valor
1 Precisão Média (AP)	0,5 - 0,95	-	-	0,187
2 Precisão Média (AP)	0,5	-	-	0,263
3 Precisão Média (AP)	0,75	-	-	0,219
4 Precisão Média (AP _{peq})	0,5 - 0,95	< 32 ² pixels	Pequeno	0,061
5 Precisão Média (AP _{méd})	0,5 - 0,95	> 32 ² pixels e < 96 ²	Médio	0,226
6 Precisão Média (AP _{gnd})	0,5 - 0,95	> 96 ² pixels	Grande	0,040
7 Recall Médio (AR)	0,5 - 0,95	-	1 objeto	0,130
8 Recall Médio (AR)	0,5 - 0,95	-	10 objetos	0,359
9 Recall Médio (AR)	0,5 - 0,95	-	100 objetos	0,367
10 Recall Médio (AR _{peq})	0,5 - 0,95	< 32 ² pixels	Pequeno	0,096
11 Recall Médio (AR _{méd})	0,5 - 0,95	> 32 ² pixels e < 96 ²	Médio	0,465
12 Recall Médio (AR _{gnd})	0,5 - 0,95	> 96 ² pixels	Grande	0,71
13 Precisão Média (AP)	0,5 - 0,95	-	UFV	0,309
14 Precisão Média (AP)	0,5 - 0,95	-	SAS	0,160

Fonte: Próprio autor (2021)

5.3 Inferência

As ortofotos do Google para a área de estudo, são datadas de 16/05/2018. Foram gerados um total de 306 arquivos no formato TIFF com uma resolução espacial de 7 cm/pixel, contendo 15104x16129 pixels. Cada um destes rasters ocupa 731 MB de espaço em disco, portanto, todos eles somaram 223.686 MB ou aproximados 223,6 GB para toda a área dos grids dos rasters. Vale ressaltar que alguns *grids* dentro da área do município foram excluídos da área de estudo por não possuírem nenhuma edificação em seu interior.

A inferência do modelo em todas as imagens do município, passando por todas as 173.546 janelas deslizantes levou 11:43hs para ser executada, e resultou em um processamento médio de 4,11 janelas/segundo, atestando a rapidez e performance da Mask R-CNN. Faz-se essencial apontar que a maior parte do tempo gasto no processamento é utilizado pelas etapas de vetorização das segmentações (*scripts* do GDAL e outras bibliotecas Python) e não das predições em si (Detectron2 e Mask R-CNN). Foi utilizado um valor de *threshold* de 0,5 na segmentação, o que significa que predições com *scores* menores que 0,5, foram desconsideradas.

Após o processamento, uma vez que as segmentações já foram transformadas em vetor no arquivo de formato *.gpkg*, este arquivo foi aberto no software QGIS para pós-processamento dos vetores encontrados. O arquivo vetorial pré-processado, possuía 33203 feições no total sendo 5487 feições da classe UFV e 27716 da classe SAS. Este arquivo vetorial resultou em 13,9 MB de espaço em disco.

A etapa de pós-processamento da segmentação das imagens foi executada e aplicou-se uma zona *buffer* de 7 cm e *dissolve* aos polígonos segmentados. A Figura 43 demonstra como este pós-processamento vetorial resolveu as inconsistências topológicas de algumas feições detectadas pelo modelo.

Figura 43 - Exemplo de problemas topológicos (a) e (c) corrigidos por pós-processamento (b) e (d)



Fonte: Adaptado de Google Earth (2018)

Com o pós-processamento dos vetores, foram obtidas 3655 feições de UFV e 26369 feições de SAS como produto final da metodologia, totalizando 30024 feições. A Tabela 5 lista os quantitativos obtidos para as classes UFV e SAS nas etapas pré e pós processamento.

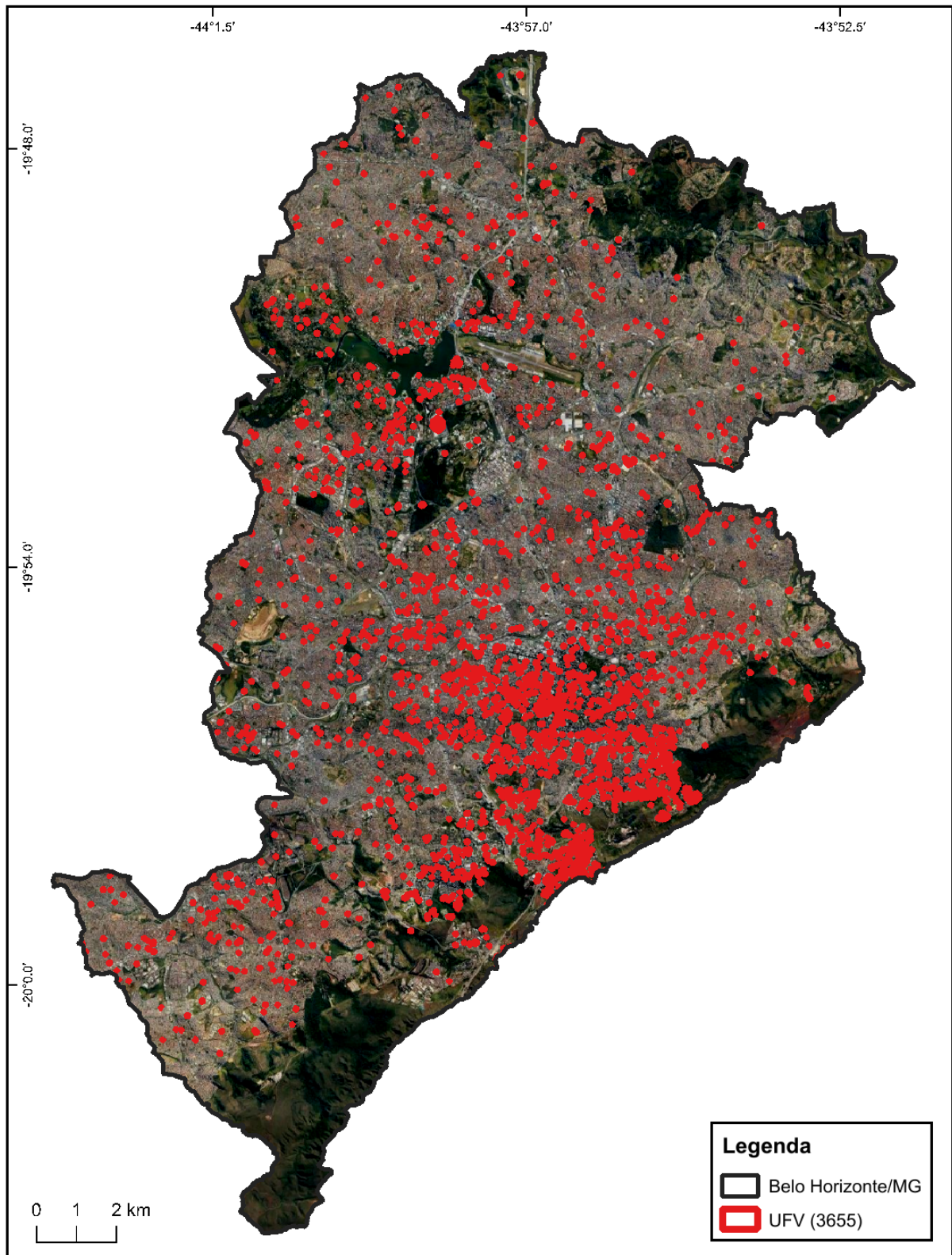
Tabela 5 - Número de feições encontradas na área de estudo por classe e etapa de processamento

Classe	Pré-processamento	Pós-processamento
UFV	5487	3655
SAS	27716	26369
TOTAL	33203	30024

Fonte: Próprio autor (2021)

As Figuras 44 e 45 apresentam os mapas contendo a distribuição espacial completa dos objetos-alvo desta pesquisa. Estas figuras representam o resultado final que pretendeu-se nesta pesquisa, com a detecção e segmentação pixel a pixel de todas as feições UFV e SAS da área de estudo.

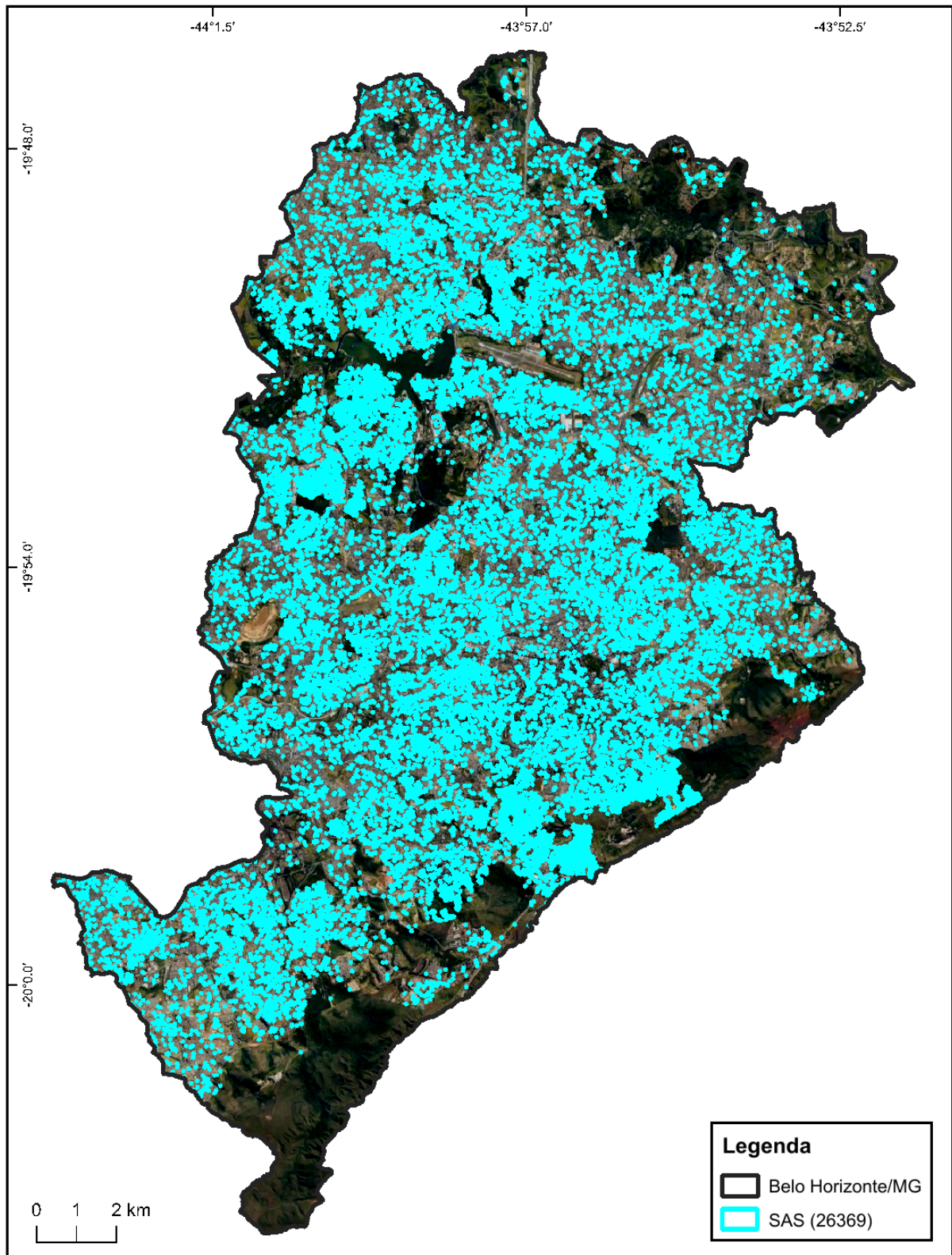
Figura 44 - Distribuição espacial das UFV detectadas na área de estudo²⁸



Fonte: Adaptado de Google Earth (2018)

²⁸ É possível acessar o mapa interativo *online* com os resultados finais desta pesquisa, contendo todas as feições UFV detectadas pelo modelo, disponível em <<https://abreufilho.com.br/mestrado/mapa/ufv>>

Figura 45 - Distribuição espacial dos SAS detectadas na área de estudo²⁹



Fonte: Adaptado de Google Earth (2018)

²⁹ É possível acessar o mapa interativo *online* com os resultados finais desta pesquisa, contendo todas as feições SAS detectadas pelo modelo, disponível em <<https://abreufilho.com.br/mestrado/mapa/sas>>

6. DISCUSSÕES

As métricas resultantes da etapa de validação (Tabela 4) ajudam a elucidar a performance do modelo. O modelo possui diferentes escalas de precisão, para objetos de área média (entre 32x32 pixels e 96x96 pixels) ele responde bem com uma $AP_{\text{méd}}=0,226$. Já para objetos com áreas menores que 32x32 pixels ou maiores que 96x96 pixels, a precisão é muito baixa, sendo $AP_{\text{peq}}=0,61$ e $AP_{\text{gnd}}=0,40$. Isto pode explicar-se pelo fato de quase todos os dados anotados terem uma dimensão média, o que faz o modelo mais sensível aos dados da mesma natureza.

A mesma lógica se aplica à métrica de AR, onde os objetos de tamanho médio possuem $AR_{\text{méd}}=0,465$, comparados aos valores de $AR_{\text{peq}}=0,096$ e $AR_{\text{gnd}}=0,71$. O que reforça ainda mais a sensibilidade ao tamanho dos objetos que o modelo tem. Considerando que o valor de $AR_{\text{méd}}$ é mais que o dobro de $AP_{\text{méd}}$ é possível afirmar que o modelo é muito bom em encontrar objetos verdadeiros, mas também encontra muitos objetos falsos positivos. Fato este, muito explicado pela heterogeneidade dos dados em uma imagem urbana.

Um exemplo que ilustra bem esta afirmação está na Figura 46, onde até mesmo um anotador humano teria dificuldade em classificar o que seriam placas fotovoltaicas nas imagens. Pergunta-se: as feições apontadas pelas setas vermelhas na Figura 46 são UFV dispostas de alguma maneira não-convencional? Ou são um tipo de telhado translúcido? Tais respostas só seriam possíveis com uma avaliação em campo ou há elementos objetivos que permitam fazer tal diferenciação apenas ao olhar a imagem?

Figura 46 - Exemplos de ambiguidade dos dados

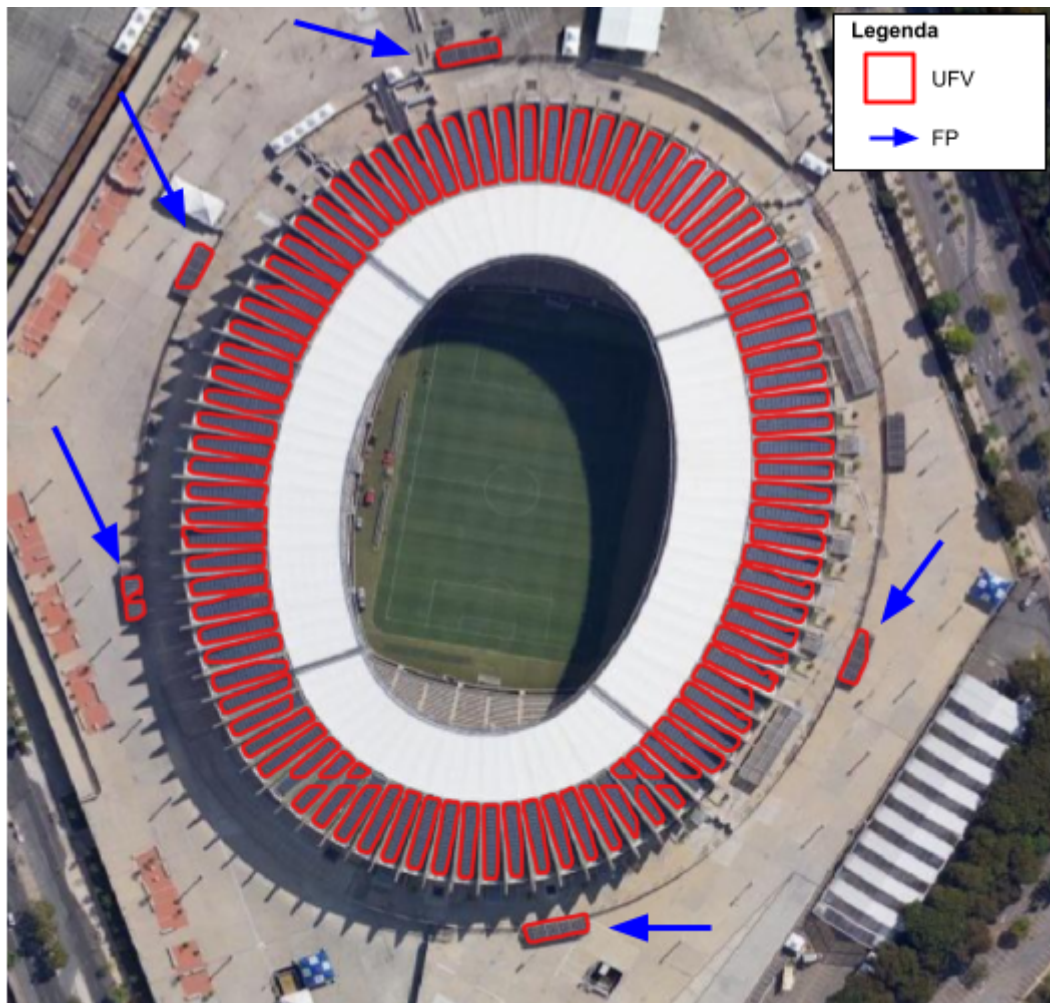


Fonte: Adaptado de Google Earth (2018)

As métricas de validação de AP para as duas classes de objetos, UFV e SAS (AP=0,309 e AP=0,160, respectivamente), permitem ligar o fato de que grandes composições de módulos fotovoltaicos aumentam a sensibilidade do modelo à esta classe especificamente, ainda que possua menos amostras de treinamento. Ora, se em uma avaliação visual dos resultados, pode-se confirmar que é muito comum que SAS ocupem pequenos espaços nos telhados das edificações, ao passo que grandes composições de módulos fotovoltaicos foram vistas com frequência durante a pesquisa, então cabe afirmar que o tamanho médio que cada uma dessas classes ocupa, fala muito sobre a superioridade da precisão do modelo ao classificar UFV se comparada a precisão de SAS.

Um bom exemplo encontrado que demonstra a boa aderência da segmentação na área do entorno do estádio Mineirão (Figura 47), onde o modelo foi capaz de identificar todas as linhas de UFV. Nesta mesma imagem é possível também perceber a presença de 7 feições que são falsos positivos (no terraço, área externa do estádio). Ou seja, apesar de um grande poder de detecção de feições corretas, o modelo ainda erra muito ao detectar falsos positivos em demasia.

Figura 47 - UFV detectados no Estádio Mineirão com falsos positivos



Fonte: Adaptado de Google Earth (2018)

Além das UFV do Mineirão, houve outros locais com grandes composições de módulos fotovoltaicos. O modelo respondeu bem a grandes áreas com UFV, embora seja claramente afetado pelas bordas das janelas deslizantes, como demonstra a figura.

Figura 48 - Locais com grandes composições de módulos fotovoltaicos corretamente segmentados



Fonte: Adaptado de Google Earth (2018)

Uma observação a ser feita sobre a falta de capacidade do modelo em classificar imagens pequenas, é que na etapa de aumento de dados, nenhuma técnica de transformação incluiu o redimensionamento ou diferentes escalas dos objetos anotados, isto pode ter afetado o poder do modelo em lidar com pequenos objetos. A Figura 49 deixa esta constatação de maneira visual, onde as grandes composições de UFV são corretamente classificadas enquanto um grande número de UFV menores são ignoradas na segmentação ou classificadas como SAS. Uma das explicações para este fenômeno pode ser o fato dos dados de treinamento não possuírem muitas representações UFV pequenas.

Figura 49 - Feições grandes de módulos fotovoltaicos corretamente segmentadas com diversos falsos positivos



Fonte: Adaptado de Google Earth (2018)

Ainda sobre o procedimento de aumento de dados, já que a área de estudo (e grande parte do território brasileiro) está localizada no hemisfério sul, e, os objetos-alvo deste trabalho, UFV e SAS são instalados voltados ao norte buscando uma maior eficiência energética, é preciso avaliar se a rotação das imagens durante o processo de treino pode influenciar na taxa de acerto da segmentação. Portanto, sugere-se que publicações futuras possam atestar o impacto da rotação de imagens na qualidade da detecção destes elementos em imagens aéreas.

Durante a realização desta pesquisa, confirmou-se o que diz LabelBox (2021) que em modelos de redes neurais, em geral, a preparação dos dados é a etapa mais trabalhosa. A maior parte do tempo investido nesta pesquisa, além da busca por literatura especializada, foi deixar os dados consistentes de modo que servissem de entrada para os algoritmos criados. Isso foi válido tanto para etapa de anotação dos valores de referência, quanto para correta extração das ortofotos do Google.

Outro problema enfrentado, foram as distorções das ortofotos usadas, onde um mesmo ponto (principalmente edificações com vários andares) apresenta diferentes resultados na tomada das imagens e pode oferecer falsos negativos e positivos relativos à presença de UFV. Dado que estas observações ocorrem raramente, para os propósitos deste trabalho, este aspecto foi ignorado. Ainda assim, uma possível solução para este problema seria o uso de imagens do tipo *True-Ortofoto*, que corrige o problema gerado pela variação radial das fotografias que compõem a ortofoto e que ainda é um tipo de produto pouco difundido. Com o advento dos drones e o barateamento para o acesso a estes produtos, ficará cada vez mais fácil o uso deste produto. A Figura 50 apresenta um exemplo deste problema de classificação problemática por conta da variação radial, encontrada na área de estudo.

Figura 50 - Distorções nas ortofotos onde edificações foram incorretamente classificadas como UFV

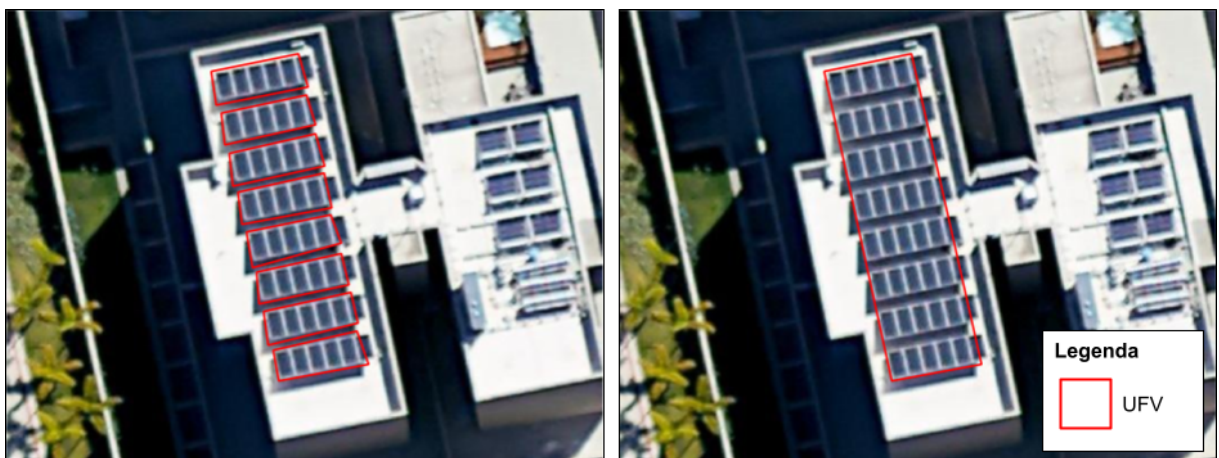


Fonte: Adaptado de Google Earth (2018)

Há uma importante discussão também a ser feita no conceito do que deve ser considerado o valor de referência, quando da etapa de anotação. Seria toda a composição feita pelas placas fotovoltaicas/aquecimento solar, ou devem ser

anotadas separadamente, cada linha de placas como sendo um valor de referência? Ou ainda, deveriam ser anotadas cada uma das placas individualmente (levando em conta a baixa performance do modelo para detectar pequenos objetos)? O que deve ser considerado um valor de referência pode ser subjetivo em alguns casos e essa subjetividade nas anotações pode atrapalhar o modelo a generalizar este fenômeno para novos dados, fora do treinamento. Um exemplo ilustrado deste problema está na Figura 51.

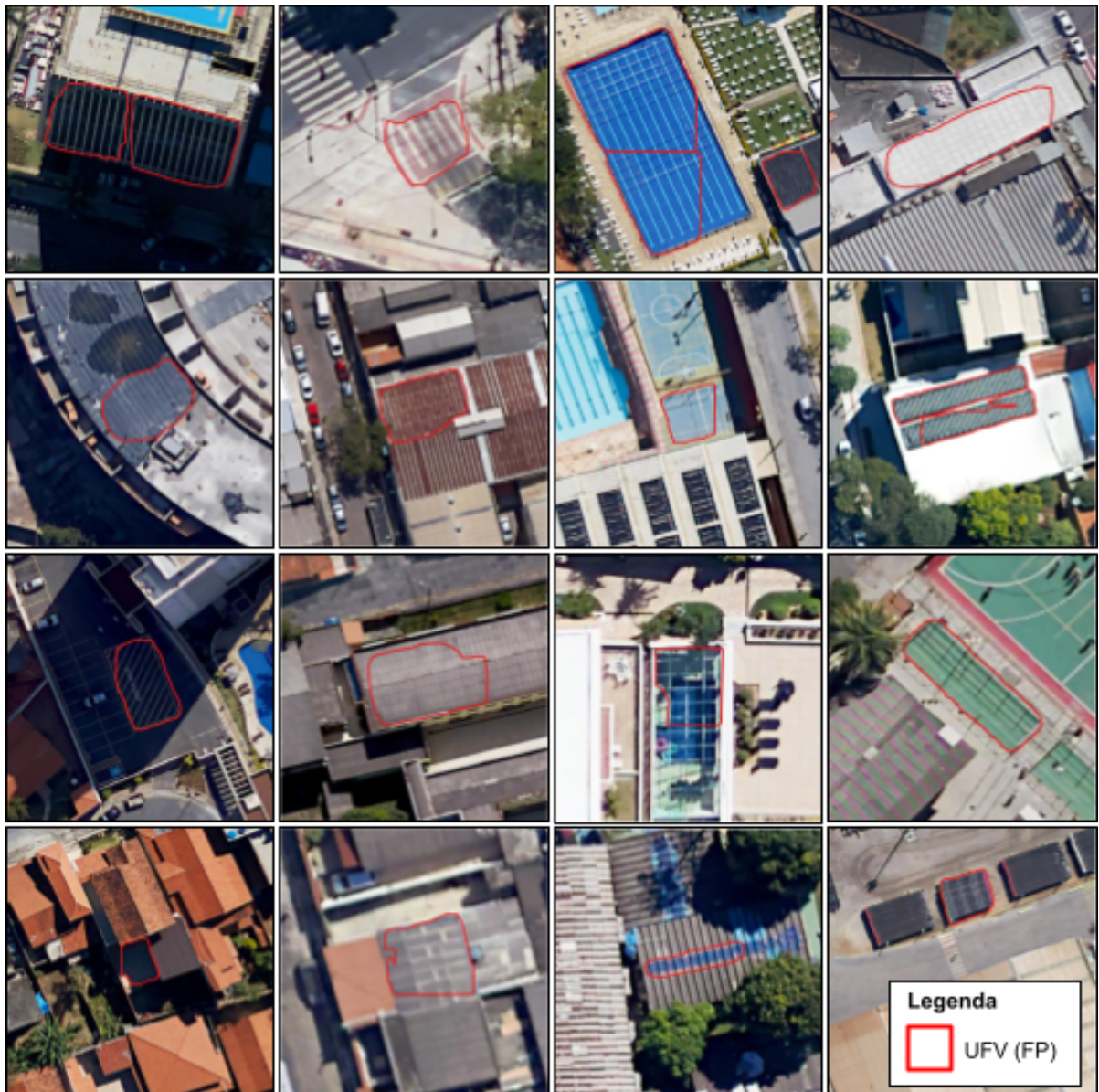
Figura 51 - Subjetividade dos valores de referência



Fonte: Adaptado de Google Earth (2018)

A Figura 52 apresenta algumas das feições incorretamente classificadas (falsos positivos) como UFV. Estas imagens apresentam alguns padrões semelhantes às características encontradas nas placas fotovoltaicas.

Figura 52 - Exemplos de falsos positivos preditos pelo modelo na área de estudo



Fonte: Adaptado de Google Earth (2018)

Durante a avaliação dos resultados, percebeu-se que a decisão de não adotar janelas deslizantes sobrepostas, de fato, representou um problema para a segmentação correta dos objetos-alvo desta metodologia. A Figura 53 demonstra um exemplo de como um mesmo objeto de classe SAS foi classificado como UFV em uma janela deslizante e classificado corretamente como SAS em outra janela.

Figura 53 - Exemplo de classificação ambígua na borda de uma janela deslizante



Fonte: Adaptado de Google Earth (2018)

As ortofotos das três cidades usadas para treinamento tinham características muito semelhantes à ortofoto da área de estudo. É preciso avaliar como o modelo que foi treinado com estas imagens se comportaria em imagens de outros sensores.

Carvalho et al. (2020) também criou um algoritmo de transformação de polígonos para anotações COCO, uma das primeiras etapas desenvolvidas no presente trabalho. Os mesmos autores também propuseram uma solução de *non-max supression* para resolver o problema das classificações de janelas deslizantes, que foram fator relevante nos resultados desta pesquisa. Eles discutem neste artigo por exemplo, como diferentes tamanhos de janelas deslizantes e tamanhos fixos de janelas resultam em performances diferentes nas classificações e que o equilíbrio entre acurácia e complexidade computacional de aumento de janelas deslizantes é um desafio.

No início da elaboração desta metodologia, a proposta inicial era de se usar um método de classificação de imagens tradicional, dando a cada imagem apenas um rótulo. À medida em que foi se desenvolvendo o trabalho, o autor presenciou uma série de novas técnicas e *frameworks* surgindo neste período, de maneira que o trabalho foi ganhando refinamento e complexidade ao absorver estas técnicas. A arquitetura de rede utilizada no trabalho, Mask R-CNN foi lançada em 2018, o framework foi lançado em 2019 e foi dando robustez ao produto desta pesquisa. Durante o íterim da pesquisa, também foram encontradas pesquisas executando metodologias parecidas com a aqui apresentada, como exemplo cita-se o trabalho de Carvalho et al. (2020) e Pešek (2018).

7. CONSIDERAÇÕES FINAIS

Em cada etapa da metodologia aqui proposta, considerou-se que há melhorias importantes a serem feitas e que ao longo da pesquisa foram anotadas e sugeridas no presente capítulo.

Quando utiliza-se ortofotos com grande detalhamento local e em grandes extensões, para que não se consuma tanto esforço computacional e tempo em áreas que estão fora de contexto, como áreas verdes ou grandes lagos, sugere-se para trabalhos futuros, a implementação de métodos que possam trabalhar de modo a previamente identificar as áreas de interesse para posterior processamento. As soluções de propostas de regiões utilizadas internamente para treinamento e inferência das CNN, talvez possam ser aplicadas também à inferência de ortofotos.

As ortofotos usadas para a classificação são de 2018. Como foi apontado na Tabela 2, de 2018 até a presente data, houve um grande aumento das instalações de UFV em todos os locais envolvidos na pesquisa, tanto na área de estudo como nas áreas utilizadas para gerar os dados de treinamento. Sugere-se sempre aplicar a modelagem e a metodologia aqui apresentadas sempre com as ortofotos mais recentes na medida do possível e com a mesma data dos bancos de dados das instalações georreferenciadas.

Na etapa de inicialização do treinamento do modelo, neste trabalho foram utilizados pesos pré-treinados em bancos de imagens que possuíam imagens naturais, de animais, objetos e outras classes que não se aplicam quando trata-se de ortofotos e/ou imagens aéreas em geral. Uma possível solução para melhorar a performance do modelo de classificação, seria inicializar a rede com pesos pré-treinados em imagens aéreas.

Outra consideração para a melhoria da performance do modelo proposto, seria a elaboração de um método em que os resultados falsos positivos que fossem detectados por um operador humano, pudessem alimentar o modelo sem que seja necessário recalibrar hiperparâmetros e executar o treinamento novamente.

Também seria um fator de relevante melhora do modelo, se os dados de entrada possuíssem bandas além do RGB (em especial, dados de LiDAR ou outros espectros como ultravioleta e infravermelho). Uma camada adicional de LiDAR por

exemplo, poderia ajudar o modelo e seria baseada também na altura em que estão as UFV, já que normalmente estes objetos estão localizados nos telhados das edificações).

Como posto no capítulo 6, para avaliar como o modelo de segmentação se comportaria em imagens geradas por sensores diferentes aos quais recebeu dados de treinamento sugere-se para publicações futuras, investigações de técnicas que possam aumentar os dados usando por exemplo, uma solução criativa proposta por Xu e Zhao (2018), que propõem o uso de Redes Adversariais Generativas (GANs) para gerar imagens orbitais artificiais (sintéticas) que poderiam ser utilizadas para transferir estilos capturados de um sensor para outros. Estes diferentes estilos das imagens incluem as características espaciais e espectrais distintas de dois ou mais sensores, luminosidade das cenas, ângulos de tomada e outros.

Não fez parte do escopo deste trabalho a calibração fina dos pesos da rede, ajustes de hiperparâmetros e testes objetivando unicamente o aumento da performance da rede treinada. Com isso posto, um valor de $AP_{UFV}=0,309$ para o objeto-alvo, foi considerado suficiente e animador, dado o propósito voltado à construção da metodologia em si.

Como exposto no capítulo 6, o modelo não responde bem às feições pequenas de UFV e que isso poderia ser mitigado com um melhor procedimento de aumento de dados e também a inserção de dados sintéticos de UFV pequenos. Portanto, para futuras publicações, serão avaliadas novas técnicas de redimensionamento na etapa de aumento de dados e técnicas de criação de dados sintéticos de treinamento.

A segmentação acurada de objetos pequenos, em uma área de 331 km² é um grande desafio, considerando-se, principalmente quando o objeto-alvo está distribuído por imagens tão heterogêneas, tal como as imagens urbanas. Este trabalho não pretendeu criar um modelo extremamente acurado ou preciso, mas, demonstrar que é possível uma melhora de performance computacional para tarefas de classificação em grandes extensões de área e com boas taxas de acertos, ainda que haja necessidade de ajustes no método proposto. A rotulagem dos dados de treinamento deste trabalho levou 26 horas para a anotação de 1719 objetos, mas este trabalho resultou em uma classificação que encontrou mais de dez vezes a

quantidade de objetos em outra cena, sem a necessidade de mais trabalho manual a ser feito. Nesse sentido, acredita-se que os objetivos propostos neste trabalho foram cumpridos.

Em termos de perspectivas futuras pode-se afirmar que os resultados aqui obtidos pressupõem que existe grande potencial no campo de processamento e identificação e segmentação automática de objetos em imagens orbitais e imagens aéreas. Deve-se ressaltar que os mesmos procedimentos poderiam ser realizados sobre ortofotos verdadeiras (*true ortofotos*, com correções de visadas e distorções laterais) para melhores resultados. Com a disseminação e popularização cada vez mais crescente da fotogrametria, a qual hoje não é restrita às grandes empresas, a realização dos voos obtidos para a geração das ortofotos aqui mencionadas se torna economicamente viável.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDULLA, Waleed. "**Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow**". Artigo de blog. Matterport. 19 de mar. de 2018. Disponível em:
<<https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>> Acesso em 20 de mar. de 2021.
- Agência Nacional de Energia Elétrica (ANEEL). "**Geração Distribuída**" Servidor de mapas do ArcGIS do órgão. Disponível em:
<https://sigel.aneel.gov.br/arcgis/rest/services/PORTAL/Geracao_Distribuida/MapServer> Acesso em 14 de mar. de 2021.
- Agência Nacional de Energia Elétrica (ANEEL). "**Resolução Normativa nº 876 de 10 de mar. de 2020**". 10 de mar. de 2020. Disponível em:
<<http://www2.aneel.gov.br/cedoc/ren2020876.pdf>>
- Agência Nacional de Energia Elétrica (ANEEL). "**Como é composta a tarifa final da energia elétrica**", fev. 2017, Disponível em:
<http://www.aneel.gov.br/conteudo-educativo/-/asset_publisher/vE6ahPFxsW/Ht/content/composicao-da-tarifa/654800?inheritRedirect=false>
- AMERIKANOS, Paris-Panagiotis. "**Image Analysis In Digital Pathology Based On Machine Learning Techniques & Deep Neural Networks**". Tese - Departamento de Sistemas Digitais da Universidade do Pireu, Grécia. 2020.
- ABDURAHMANOV, Majnun. "**Automated Detection Of Objects Of Interest From Aerial Imagery Using Transfer Learning**". Tese de mestrado, não publicada Tallinn University Of Technology, 2020. Disponível em:
<<http://rgdoi.net/10.13140/RG.2.2.19724.33926>>
- ACSM, American Congress on Surveying and Mapping. "**American Society for Photogrammetry and Remote Sensing**". Glossary of the Mapping Sciences, American Society of Civil Engineers, p. 370, ISBN 9780784475706. 1994.
- BASTOS, Bruno da Cunha; ERCOLIN FILHO, Leonardo. "**Utilização da Tecnologia LIDAR em Projetos de Engenharia de Infraestrutura Viária**", 2012. Disponível em: <<http://www.anea.org.br/artigos/LidarInfraEstruturaViaria.pdf>>
Acesso em: 22 out. 2019

- BALLARD, Dana H., BROWN, Christopher M. **“Computer Vision”**. Englewood Cliffs: Prentice-Hall, 1982.
- BILGEN, Selçuk et al. **“Global warming and renewable energy sources for sustainable development: A case study in Turkey”**. Renewable and Sustainable Energy Reviews, v. 12, n. 2, p. 372–396, fev. 2008. Disponível em: <<http://dx.doi.org/10.1016/j.rser.2006.07.016>>.
- BISONG, Ekaba. **“Building Machine Learning and Deep Learning Models on Google Cloud Platform”**. [S.l.]: Apress, 2019. Disponível em: <<http://dx.doi.org/10.1007/978-1-4842-4470-8>>.
- British Petroleum (BP). **“BP Statistical Review of World Energy”**, 68th ed. 2019.
- BROWNLEE, Jason. **“How to Use Mask R-CNN in Keras for Object Detection in Photographs”**. Deep Learning for Computer Vision. 24 de mai. de 2019. Disponível em: <<https://machinelearningmastery.com/how-to-perform-object-detection-in-photographs-with-mask-r-cnn-in-keras/>> Acessado em 18 de mar. de 2021.
- BROWNLEE, Jason. **“Classification versus Regression in Machine Learning”**. Start Machine Learning. 11 de dez. de 2017. Disponível em: <<https://machinelearningmastery.com/classification-versus-regression-in-machine-learning>> Acessado em 15 de mar. de 2021.
- BROWNLEE, Jason. **“Overfitting and Underfitting With Machine Learning Algorithms”**. Machine Learning Algorithms. 21 de mar. de 2016. Disponível em: <<https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>> Acessado em 17 de mar. de 2021.
- BHUIYAN, Md Abul Ehsan et al. **“Understanding the Effects of Optimal Combination of Spectral Bands on Deep Learning Model Predictions: A Case Study Based on Permafrost Tundra Landform Mapping Using High Resolution Multispectral Satellite Imagery”**. Journal of Imaging, v. 6, n. 9, p. 97, 17 set. 2020. Disponível em: <<http://dx.doi.org/10.3390/jimaging6090097>>.

- CARVALHO, Osmar Luiz Ferreira De et al. “**Instance Segmentation for Large, Multi-Channel Remote Sensing Imagery Using Mask-RCNN and a Mosaicking Approach**”. *Remote Sensing*, v. 13, n. 1, p. 39, 24 dez. 2020. Disponível em: <<http://dx.doi.org/10.3390/rs13010039>>.
- CHINA, Comitê Central do Partido Comunista da China, “**The 13th Five-Year Plan for Economic and Social Development of the People’s Republic of China**”; Xinhua, China. out. 2016. Disponível em: <<http://en.ndrc.gov.cn/newsrelease/201612/P020161207645765233498.pdf>>
- CHINA, República Popular da. “**Renewable Energy Law of the People’s Republic of China**”, 2006. Disponível em: <<https://web.archive.org/web/20190124160351/http://english.mofcom.gov.cn/article/policyrelease/Businessregulations/201312/20131200432160.shtml>>
- CORDTS, M; OMRAN, M; RAMOS, S; REHFELD, T; ENZWEILER, M; BENENSON, R; FRANKE, U; ROTH, S; e SCHIELE, B. “**The Cityscapes Dataset for Semantic Urban Scene Understanding**” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- DENG, J. et al. “**Imagenet: A large-scale hierarchical image database.**” In 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. 2009.
- DENG, Li. “**Deep Learning: Methods and Applications**”. *Foundations and Trends® in Signal Processing*, v. 7, n. 3–4, p. 197–387, 2014. Disponível em: <<http://dx.doi.org/10.1561/20000000039>>.
- DOUCETTE, Pete; AGOURIS, Peggy; STEFANIDIS, Anthony. “**Automated Road Extraction from High Resolution Multispectral Imagery**”. *Photogrammetric Engineering & Remote Sensing*, v. 70, n. 12, p. 1405–1416, dez. 2004. Disponível em: <<http://dx.doi.org/10.14358/PERS.70.12.1405>>.
- DOUGLAS, David H; PEUCKER, THOMAS K. “**Algorithms For The Reduction Of The Number Of Points Required To Represent A Digitized Line Or Its Caricature**”. *Cartographica: The International Journal for Geographic Information and Geovisualization*, v. 10, n. 2, p. 112–122, dez. 1973. Disponível em: <<http://dx.doi.org/10.3138/FM57-6770-U75U-7727>>

- ECMA, European Computer Manufacturers Association. “**ECMA-404 - The JSON Data Interchange Syntax**”. 2ª edição. Dez. 2017. Disponível em: <<http://www.ecmainternational.org/publications/files/ECMA-ST/ECMA-404.pdf>>. Acesso em: 15 de abr. de 2018.
- ELISHA, Odemakinde. “**Overcoming overfitting in image classification using data augmentation**” Artigo de blog. 26 de jun. de 2020. Disponível em: <<https://heartbeat.fritz.ai/overcoming-overfitting-in-image-classification-using-data-augmentation-9858c5cee986>> Acesso em: 19 de mar. de 2021
- Energy Information Administration (EIA). “**Annual Energy Outlook**”, Washington, DC, EUA, 2019 - Disponível em: <<https://www.eia.gov/outlooks/aeo/pdf/aeo2019.pdf>> Acesso em: 18 de ago. de 2019.
- FAIR, Facebook AI Research. “**Detectron2: A PyTorch-based modular object detection library**”. ML Applications/Open Source. 10 de out. de 2019. Disponível em: <<https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library/>> Acesso em 19 de mar. de 2021.
- FARIA, Thaísa Santos. “**Classificação em área urbana apoiada em imagens aéreas e dados LiDAR**”. Dissertação de mestrado. Universidade Federal de Minas Gerais. 2017.
- FENG, Di et al. “**Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges**”. IEEE Transactions on Intelligent Transportation Systems, p. 1–20, 2020.
- GDAL/OGR contributors (2019). “**GDAL/OGR Geospatial Data Abstraction software Library**.” Open Source Geospatial Foundation. Disponível em: <<https://gdal.org>>.
- GIRSHICK, Ross. “**Fast R-CNN**” Conference on Computer Vision and Pattern Recognition (CVPR). 30 de abr. de 2015. Disponível em: <<https://arxiv.org/abs/1504.08083>> Acessado em 18 de mar. de 2021.

- GIRSHICK, Ross; DONAHUE, Jeff; DARRELL, Trevor; MALIK, Jitendra. “**Rich feature hierarchies for accurate object detection and semantic segmentation**” Conference on Computer Vision and Pattern Recognition (CVPR). 2014. Disponível em: <<https://arxiv.org/abs/1311.2524>> Acessado em 18 de mar. de 2021.
- GOODFELLOW, Ian, BENGIO, Yoshua, COURVILLE, Aaron. “**Deep Learning**”. MIT Press, 1º edição. 2016.
- GOERTZEL, “**Ben. From Narrow AI to AGI via Narrow AGI?**”. Artigo de blog. Disponível em: <<https://medium.com/@bengoertzel/from-narrow-ai-to-agi-via-narrow-agi-8d828d8c2aa2>> Acesso em 17 de mar. de 2021.
- GOODING, James et al. “**Solar City Indicator: A methodology to predict city level PV installed capacity by combining physical capacity and socio-economic factors**”. Solar Energy, v. 95, p. 325–335, set. 2013. Disponível em: <<http://dx.doi.org/10.1016/j.solener.2013.06.027>>.
- GOOGLE. “**Maps URLs**” Google Maps Platform. Documentação em repositório digital. 2021. Disponível em: <<https://developers.google.com/maps/documentation/urls/get-started>> Acesso em 19 de mar. de 2021.
- GOOGLE. “**Maps Static API - Overview**” (seção de site da web)” ago. 2019. Disponível em: <<https://developers.google.com/maps/documentation/maps-static/intro>>.
- GOOGLE EARTH. Maxar Technologies 2018. 16 de maio de 2018. Disponível em: <<https://earth.google.com/web/>>.
- GONZALEZ, Sebastian; ARELLANO, Claudia; TAPIA, Juan E. “**Deepblueberry: Quantification of Blueberries in the Wild Using Instance Segmentation**”. IEEE Access, v. 7, p. 105776–105788, 2019. Disponível em: <<http://dx.doi.org/10.1109/ACCESS.2019.2933062>>.
- HAFIZ, Abdul Mueed; BHAT, Ghulam Mohiuddin. “**A survey on instance segmentation: state of the art**”. International Journal of Multimedia Information Retrieval, v. 9, n. 3, p. 171–189, 3 jul. 2020. Disponível em: <<http://dx.doi.org/10.1007/s13735-020-00195-x>>.

- HAUKKALA, Teresa. “**Does the sun shine in the High North? Vested interests as a barrier to solar energy deployment in Finland**”. Energy Research & Social Science, v. 6, p. 50–58, mar. 2015. Disponível em: <<http://dx.doi.org/10.1016/j.erss.2014.11.005>>.
- HARDESTY, Larry. “**Explained: Neural networks**”. MIT News Office - MIT News on Campus and Around the World. Disponível em: <<https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>> Acessado em 15 de mar. de 2021.
- HE, Kaiming; GKIOXARI, Georgia; DOLLAR, Piotr; GIRSHICK, Ross. “**Mask R-CNN**”. 2017 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION (ICCV), out. 2017. Disponível em: <<http://dx.doi.org/10.1109/ICCV.2017.322>>.
- HERRMANN, Gustavo Casarim. “**Análise E Aplicação De Redes R-cnn Para Identificação De Cobras Peçonhentas A Partir De Imagens**”. Trabalho de conclusão de curso. Universidade Estadual de Londrina. 2019
- HUANG, Zhaojian; MENDIS, Thushini; XU, Shen. “**Urban solar utilization potential mapping via deep learning technology: A case study of Wuhan, China**”. Applied Energy, v. 250, p. 283–291, set. 2019. Disponível em: <<http://dx.doi.org/10.1016/j.apenergy.2019.04.113>>.
- HUBEL, D. H. & Wiesel, T. N. (1959). “**Receptive fields of single neurones in the cat’ s striate cortex.**” Journal of Physiology, 1959. 148, 574-591.
- IBGE, Instituto Brasileiro de Geografia e Estatística “**Estimativas da população residente nos municípios brasileiros com data referência em 1º de julho de 2020**”. 28 de ago. de 2020. Acesso em: 19 de mar. de 2021.
- IBGE e Órgãos Estaduais de Estatística, Secretarias Estaduais de Governo e Superintendência da Zona Franca de Manaus - SUFRAMA, 2016. Disponível em: <<https://cidades.ibge.gov.br/brasil/mg/belo-horizonte/panorama>> Acesso em: 21 de nov. 2019.
- IBGE, Cadastro Central de Empresas (CEMPRE). Rio de Janeiro: IBGE, 2019 Disponível em: <<https://cidades.ibge.gov.br/brasil/mg/belo-horizonte/panorama>> Acesso em: 21 de nov. 2019.

- IRENA. International Renewable Energy Agency. **“Renewable Power Generation Costs in 2019”**. Abu Dhabi, 2020. Disponível em: <https://www.irena.org/publications/2020/Jun/Renewable-Power-Costs-in-2019>. Acesso em: 24 abril 2021.
- International Energy Agency (IEA). **“Market Report Series: Renewables 2018”**, Paris, France, 2018.
- IPCC, 2014: **“Climate Change 2014: Synthesis Report, Summary for Policymakers”**. Geneva, Switzerland, 31 pp.
- JUNG, Alexander B. et al. **“imgaug”** Repositório Online. 2019. Disponível em: <https://github.com/aleju/imgaug> Acesso em 19 de mar. de 2021
- JUNQUEIRA, Bárbara. Fotografias realizadas em campo no bairro Mangabeiras, Belo Horizonte - Minas Gerais. 12 de mar. de 2021.
- KAVLAKOGLU, Eda. **“AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What’s the Difference?”**. IBM Cloud Blog. 27 de mai. de 2020. Disponível em: <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks> Acesso em 17 de mar. de 2021.
- KERN, Florian; ROGGE, Karoline S. **“The pace of governed energy transitions: Agency, international dynamics and the global Paris agreement accelerating decarbonisation processes?”** Energy Research & Social Science, v. 22, p. 13–17, dez. 2016. Disponível em: <http://dx.doi.org/10.1016/j.erss.2016.08.016>.
- KOUHESTANI, Fariborz Mansouri et al. **“Evaluating solar energy technical and economic potential on rooftops in an urban setting: the city of Lethbridge, Canada”**. International Journal of Energy and Environmental Engineering, v. 10, n. 1, p. 13–32, 9 nov. 2018. Disponível em: <http://dx.doi.org/10.1007/s40095-018-0289-1>.
- KUZNETSOVA, Alina et al. **“The Open Images Dataset V4”**. International Journal of Computer Vision, v. 128, n. 7, p. 1956–1981, 13 mar. 2020. Disponível em: <http://dx.doi.org/10.1007/s11263-020-01316-z>.

- LABELBOX. **“Image Segmentation 101”**. Introduction to Image Segmentation for Machine Learning and AI. Website. 2021. Disponível em: <<https://labelbox.com/image-segmentation-overview>> Acesso em 19 de mar. de 2021.
- LEE, Stephen et al. **“DeepRoof”**. “THE 25TH ACM SIGKDD INTERNATIONAL CONFERENCE”, 2019, [S.I.]: ACM Press, 2019. Disponível em: <<http://dx.doi.org/10.1145/3292500.3330741>>.
- LI, Miao et al. **“A Review of Remote Sensing Image Classification Techniques: the Role of Spatio-contextual Information”**. European Journal of Remote Sensing, v. 47, n. 1, p. 389–411, jan. 2014. Disponível em: <<http://dx.doi.org/10.5721/EuJRS20144723>>.
- LIN, Tsung-Yi et al. **“Feature Pyramid Networks for Object Detection”**. In: 2017 IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), jul. 2017, [S.I.]: IEEE, jul. 2017. Disponível em: <<http://dx.doi.org/10.1109/CVPR.2017.106>>.
- LIN, Tsung-Yi et al. **“Microsoft COCO: Common Objects in Context”**. Computer Vision – ECCV 2014. [S.I.]: Springer International Publishing, 2014. p. 740–755. Disponível em: <http://dx.doi.org/10.1007/978-3-319-10602-1_48>.
- McCULLOCH, Warren S.; FITCH, Frederic B.; PITTS, Walter. **“A logical calculus of the ideas immanent in nervous activity.”** Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. Journal of Symbolic Logic, v. 9, n. 2, p. 49–50, jun. 1944. Disponível em: <<http://dx.doi.org/10.2307/2268029>>.
- MOHANRAJAN, Sam Navin; LOGANATHAN, Agilandeeswari; MANOHARAN, Prabukumar. **“Survey on Land Use/Land Cover (LU/LC) change analysis in remote sensing and GIS environment: Techniques and Challenges”**. Environmental Science and Pollution Research, v. 27, n. 24, p. 29900–29926, 5 jun. 2020. Disponível em: <<http://dx.doi.org/10.1007/s11356-020-09091-7>>
- MAINZER, Kai et al. **“Assessment of rooftop photovoltaic potentials at the urban level using publicly available geodata and image recognition techniques”**. Solar Energy, v. 155, p. 561–573, out. 2017. Disponível em: <<http://dx.doi.org/10.1016/j.solener.2017.06.065>>.

- MARGOLIS, R.; GAGNON, P.; MELIUS, J.; CALEB, P. e ELMORE, R. "**Rooftop Solar Photovoltaic Technical Potential in the United States: A Detailed Assessment**", Technical Report, National Renewable Energy Laboratory, jan. 2016 - Disponível em: <<https://www.nrel.gov/docs/fy16osti/65298.pdf>>
- MARGOLIS, R.; ZUBOY, J. "**Non technical Barriers to Solar Energy Use: Review of Recent Literature**". Office of Scientific and Technical Information (OSTI), 1 set. 2006. Disponível em: <<http://dx.doi.org/10.2172/893639>>.
- MARGOLIS, R.; GAGNON, P.; MELIUS, J.; CALEB, P. e ELMORE, R. "**Rooftop Solar Photovoltaic Technical Potential in the United States: A Detailed Assessment**", Technical Report, National Renewable Energy Laboratory, jan. 2016 - Disponível em: <<https://www.nrel.gov/docs/fy16osti/65298.pdf>>
- MARQUES, Eduarda Almeida Leão. "**Estudo sobre redes neurais de aprendizado profundo com aplicações em classificação de imagens**". 2016. ix, 51 f., il. Monografia (Bacharelado em Estatística)—Universidade de Brasília, Brasília, 2016. Acesso em: 23 nov. 2020 - Disponível em: <<https://bdm.unb.br/handle/10483/15147>>
- MELIUS, J.; MARGOLIS, R.; e ONG, S. "**Estimating Rooftop Suitability for PV: A Review of Methods, Patents, and Validation Techniques**." Golden, CO: National Renewable Energy Laboratory. dez. 2013 NREL/TP-6A20-60593 - Disponível em: <<https://www.nrel.gov/docs/fy14osti/60593.pdf>>.
- MELLO, C. C. .S.; SALIM, D. H. C.; TEIXEIRA, L. P. T.; "USO DE VANT COMO FERRAMENTA AUXILIAR NA GESTÃO INTEGRADA DE ATERROS SANITÁRIOS", p. 2292-2302 . In: São Paulo: Blucher, 2017.ISSN 2357-7592, DOI 10.5151/xveneeamb-230. Disponível em: <https://www.proceedings.blucher.com.br/download-pdf/293/27459>. Acesso em: 11/08/2021.
- MITIDIERI, Mauricio Franco. "**Análise do Potencial de Geração Distribuída de Energia Solar Fotovoltaica nos Setores Bancário, de Educação Básica e Postos De Gasolina**" – Rio de Janeiro: UFRJ/COPPE, 2017.

- MOORE, Gordon E. "**Cramming more components onto integrated circuits**", Electronics, v. 38, n. 8, abr. 1965. Acesso em: 19/08/2019 - Disponível em: <https://web.archive.org/web/20090126170054/http://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf>
- NAAM, Ramez. "**Smaller, cheaper, faster: Does Moore's law apply to solar cells?**", Scientific American, mar. 2011 - Acesso em: 20/08/2019 - Disponível em: <<https://blogs.scientificamerican.com/guest-blog/smaller-cheaper-faster-does-moores-law-apply-to-solar-cells/>>
- NERO, M. A.; SALIM, D. H. C. ; MELLO, C. C. S. ; AMARANTE, R. T. ; FONSECA, B. M. ; TEMBA, P. C. "**Case study of a model of local solar radiation potential and discussion on the associated sustainable applications and potentials**". SUSTENTABILIDADE EM DEBATE, v. 11, p. 173-207, 2020.
- Organização das Nações Unidas (ONU). "**Paris Agreement**" C.N.63.2016.TREATIES-XXVII.7.d, 2016.
- Organização das Nações Unidas (ONU). "**World Population Prospects 2019: Highlights**" ST/ESA/SER.A/423, Department of Economic and Social Affairs, Population Division 2019.
- OSHEA, Keiron; NASH, Ryan. "**An Introduction to Convolutional Neural Networks**". arXiv. 2015. Disponível em: <<https://arxiv.org/abs/1511.08458>> Acesso em 17 de mar. de 2021.
- PEŠEK, Ondřej. "**Mask R-CNN in GRASS GIS**". Tese de Mestrado. Université Libre de Bruxelles. Jun. de 2018. Disponível em: <<https://github.com/ctu-geoforall-lab-projects/dp-pesek-2018/blob/master/text/ondrej-pesek-dp-2018.pdf>> Acessado em: 15/03/2021
- PARTAIN, Larry et al. "**Swanson's Law plan to mitigate global climate change**". IEEE 43rd Photovoltaic Specialists Conference, jun. 2016. Disponível em: <<http://dx.doi.org/10.1109/PVSC.2016.7750284>>.
- PRODABEL, Empresa de Informática e Informação do Município de Belo Horizonte. 2015.

- Programa das Nações Unidas para o Desenvolvimento (PNUD), Instituto de Pesquisa Econômica Aplicada (Ipea) e Fundação João Pinheiro (FJP), 2010. <http://www.atlasbrasil.org.br/2013/pt/perfil_m/belo-horizonte_mg> Acesso em 21 de nov. 2019
- PUTTEMANS, Steven; VAN RANST, Wiebe; GOEDEMÉ, Toon. “**Detection of Photovoltaic Installations in RGB Aerial Imaging: A Comparative Study**”. 2016. Disponível em: <<https://core.ac.uk/download/pdf/45291796.pdf>>
- PYTORCH. “**About: Learn PyTorch’s features and capabilities**” Website. Disponível em: <<https://pytorch.org/features/>> Acesso em 19 de mar. de 2021.
- QGIS Development Team, 2009. “**QGIS Geographic Information System**”. Open Source Geospatial Foundation <<https://qgis.org>> Acesso em nov. 2019.
- RAJ, Hritik. “**MATLAB | RGB image representation**”. 26 de jun. de 2018. Disponível em: <<https://www.geeksforgeeks.org/matlab-rgb-image-representation/>> Acessado em 15/03/2021.
- ROHRER, Brandon. “**How convolutional neural networks work, in depth**” Vídeo do YouTube, End-to-End Machine Learning School Course 193. 17 de out. de 2018. Disponível em: <https://www.youtube.com/watch?v=JB8T_zN7ZC0> Acesso em 17 de mar. de 2021
- RYLATT, M; GADSDEN, S; LOMAS, K. “**GIS-based decision support for solar energy planning in urban environments**”. Computers, Environment and Urban Systems, v. 25, n. 6, p. 579–603, nov. 2001. Disponível em: <[http://dx.doi.org/10.1016/S0198-9715\(00\)00032-6](http://dx.doi.org/10.1016/S0198-9715(00)00032-6)>.
- SAHA, Sumit. “**A Comprehensive Guide to Convolutional Neural Networks, the ELI5 way**”. Towards Data Science. 15 de dez. de 2018. Acesso em 20/11/2020 - Disponível em: <<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>>
- SATSUMMIT, Satellites in Global Development. “**Spatial and Temporal Resolution**”. 2017. Disponível em: <<https://landscape.satsummit.io/capture/spatial-and-temporal-resolution.html>> Acessado em 15 de mar. de 2021.

- SCHMIDHUBER, Jürgen. “**Deep learning in neural networks: An overview. Neural Networks**”, v. 61, p. 85–117, jan. 2015. Disponível em: <<http://dx.doi.org/10.1016/j.neunet.2014.09.003>>.
- SEED, Startups And Entrepreneurship Ecosystem Development. “**Edital de chamamento para a edição especial do Seed 2021**”. Subsecretaria de Ciência, Tecnologia e Inovação da Secretaria De Estado De Desenvolvimento Econômico do Governo Do Estado De Minas Gerais. Disponível em: <<http://seed.mg.gov.br/wp-content/uploads/2020/12/SEED-EDI%C3%87%C3%83O-ESPECIAL-EDITAL.pdf>> Acesso em 15 de mar. de 2021.
- SILVA, Rodrigo Pinto. “**Análise da qualidade posicional de produtos gerados por LST e ARP em diferentes alturas de voo**”. 2019. 102 f. Dissertação (Mestrado em Engenharia Civil) - Universidade Federal de Viçosa, Viçosa. 2019. Disponível em: <https://www.locus.ufv.br/handle/123456789/25747>. Acesso em: 11/08/2021.
- REN, Shaoqing; HE, Kaiming; GIRSHICK, Ross; SUN, Jian. “**Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks**”. Conference on Computer Vision and Pattern Recognition (CVPR). 2016. Disponível em: <<https://arxiv.org/pdf/1506.01497.pdf>> Acesso em 18 de mar. de 2021.
- SMIL, Vaclav. “**Distributed Generation and Megacities: Are Renewables the Answer?**” IEEE Power and Energy Magazine, v. 17, n. 2, p. 37–41, mar. 2019. Disponível em: <<http://dx.doi.org/10.1109/MPE.2018.2884112>>.
- SOLOMON, Chris, e Toby Breckon. “**Fundamentals of Digital Image Processing a Practical Approach with Examples in Matlab**”. Wiley-Blackwell, 2011.
- SONG, Shiran et al. “**Intelligent Object Recognition of Urban Water Bodies Based on Deep Learning for Multi-Source and Multi-Temporal High Spatial Resolution Remote Sensing Imagery**”. Sensors, v. 20, n. 2, p. 397, 10 jan. 2020.
- SOYTAS, Ugur; SARI, Ramazan. “**Energy consumption and GDP: causality relationship in G-7 countries and emerging markets**”. Energy Economics, v. 25, n. 1, p. 33–37, jan. 2003. Disponível em: <[http://dx.doi.org/10.1016/S0140-9883\(02\)00009-9](http://dx.doi.org/10.1016/S0140-9883(02)00009-9)>.

- STRUPEIT, Lars; PALM, Alvar. **“Overcoming barriers to renewable energy diffusion: business models for customer-sited solar photovoltaics in Japan, Germany and the United States”**. Journal of Cleaner Production, v. 123, p. 124–136, jun. 2016. Disponível em: <http://dx.doi.org/10.1016/j.jclepro.2015.06.120>.
- SUZUKI, Satoshi; BE, Keiichi A. **“Topological structural analysis of digitized binary images by border following”**. Computer Vision, Graphics, and Image Processing, v. 30, n. 1, p. 32–46, abr. 1985. Disponível em: [http://dx.doi.org/10.1016/0734-189X\(85\)90016-7](http://dx.doi.org/10.1016/0734-189X(85)90016-7).
- SWANSON, Richard M. **“A vision for crystalline silicon photovoltaics”**. Progress in Photovoltaics: Research and Applications, v. 14, n. 5, p. 443–453, 2006. Disponível em: <http://dx.doi.org/10.1016/j.pip.2006.07.009>.
- SZOSTAK, Rodrigo. Células Solares Híbridas Baseadas Em Perovskitas. 2016. 67f. Dissertação (Mestrado em Química) - Instituto de Química, Universidade Estadual de Campinas, 2016.
- TONG, Kang; WU, Yiquan; ZHOU, Fei. **“Recent advances in small object detection based on deep learning: A review”**. Image and Vision Computing, v. 97, p. 103910, maio 2020. Disponível em: <http://dx.doi.org/10.1016/j.imavis.2020.103910>.
- TU, Qiang et al. **“The profitability of onshore wind and solar PV power projects in China - A comparative study”**. Energy Policy, v. 132, p. 404–417, set. 2019. Disponível em: <http://dx.doi.org/10.1016/j.enpol.2019.05.041>.
- U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE). 2018. Acesso em 22/09/2019 - Disponível em: <https://www.energy.gov/eere/solar/solar-energy-glossary>
- USMAN; B. **“Satellite imagery land cover classification using Kmeans clustering algorithm: computer vision for environmental information extraction”**. Elixir Journal of Computer Science and Eng: 18671–18675. 2013.

- VARGAS, Michael Collin. **“Energia solar disponível e energia gerada em sistemas fotovoltaicos em diferentes inclinações e orientações de telhados de edificações”**. 2021. 125 f. Dissertação (Mestrado em Engenharia de Energia na Agricultura) - Universidade Estadual do Oeste do Paraná, Cascavel - PR. Disponível em: http://tede.unioeste.br/bitstream/tede/5384/5/Michael_Vargas2021.pdf. Acesso em: 11/08/2021.
- XIN-GANG, Zhao; YI-MIN, Xie. **“The economic performance of industrial and commercial rooftop photovoltaic in China”**. Energy, v. 187, p. 115961, nov. 2019. Disponível em: <<http://dx.doi.org/10.1016/j.energy.2019.115961>>.
- XU, Chunxue; ZHAO, Bo. **“Satellite Image Spoofing: Creating Remote Sensing Dataset with Generative Adversarial Networks (Short Paper)”**. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany, 2018. Disponível em: <<http://drops.dagstuhl.de/opus/volltexte/2018/9395/>>.
- YAN, Jinyue et al. **“City-level analysis of subsidy-free solar photovoltaic electricity price, profits and grid parity in China”**. Nature Energy, v. 4, n. 8, p. 709–717, ago. 2019. Disponível em: <<http://dx.doi.org/10.1038/s41560-019-0441-z>> Acesso em: 20 nov. 2019
- YU, Jiafan et al. **“DeepSolar: A Machine Learning Framework to Efficiently Construct a Solar Deployment Database in the United States”**. Joule, v. 2, n. 12, p. 2605–2617, dez. 2018. Disponível em: <<http://dx.doi.org/10.1016/j.joule.2018.11.021>>.
- LIVINGSTONE, Devon; CHAU, Justin. **“Otosopic diagnosis using computer vision: An automated machine learning approach”**. The Laryngoscope, 18 set. 2019. Disponível em: <<http://dx.doi.org/10.1002/lary.28292>>.
- WU, Yuxin; KIRILLOV, Alexander; MASSA, Francisco; LO, Wan-Yen; GIRSHICK, Ross. **Detectron2**. 2019. Disponível em: <<https://github.com/facebookresearch/detectron2>>