



REVISÃO DE SOFTWARES LIVRES E DE CÓDIGO ABERTO DE VISÃO COMPUTACIONAL PARA AGRICULTURA DE PRECISÃO

Rogério P. dos SANTOS 1, Samuel Nelson Melegari de SOUZA1

¹Departamento de pós-graduação em Engenharia de Energia na Agricultura Universidade Estadual do Oeste do Paraná – *Campus Cascavel* – Paraná - BR

E-mails:rogerio.dosantos@ifpr.edu.br, samuel.souza@unioeste.br

Resumo: A utilização de drones na agricultura de precisão tem se mostrado promissora, proporcionando um apoio fundamental para as práticas agrícolas. Entre as tecnologiaschave que impulsionam essa abordagem, destaca-se a visão computacional, que permite a análise e interpretação de imagens e vídeos capturadas por drones. Neste contexto, o objetivo deste estudo foi realizar uma revisão dos softwares de visão computacional de código aberto, aplicados a drones na área da agricultura de precisão. A metodologia empregada abrangeu fontes de pesquisa como Google Acadêmico, GitHub e GitLab, sendo definidos critérios precisos de seleção. Nove estudos pertinentes foram identificados que exploram a aplicação de software de código aberto em drones agrícolas, permitindo uma comparação desses recursos. Os resultados destacam a disponibilidade limitada de soluções de código aberto, apesar das vantagens que esses softwares oferecem, como customização de algoritmos e engajamento da comunidade. Esta análise reforça a urgência de mais iniciativas para criar ferramentas colaborativas para o progresso contínuo da agricultura de precisão.

Palavras-chave: Agricultura de precisão; drones; softwares livres; visão computacional.

INTRODUÇÃO

A agricultura de precisão, impulsionada pelo uso de drones, transformou a maneira como os agricultores monitoram suas fazendas. Essas aeronaves não tripuladas possibilitam a coleta de dados essenciais sobre solo, saúde das plantas e topografia. No entanto, apesar do rápido crescimento dos serviços de drones comerciais, as limitações operacionais ainda precisam ser superadas. A baixa autonomia de decisão e a necessidade de operadores experientes são alguns desafios. Superar essas limitações nos permite detectar problemas precocemente, otimizar as práticas agrícolas e aumentar a eficiência da produção (HONG et al., 2021).

As tecnologias de visão computacional podem ser implementadas em drones para automatizar as tarefas agrícolas, como monitoramento de plantações, detecção de doenças ou pragas, mapeamento de áreas de irrigação e aplicação precisa de insumos. Um conjunto







de dados de imagens agrícolas é essencial para treinar e aprimorar os algoritmos utilizados pelos drones, permitindo assim melhorar o desempenho na detecção e análise de elementos relevantes no campo (LU; YOUNG, 2020).

De acordo com Kaur e Chahal (2022), algoritmos de visão computacional de código aberto são cruciais para o avanço das aplicações na agricultura de precisão. Ao contrário do software proprietário, o software livre e de código aberto oferece vantagens significativas, como a capacidade de personalizar e melhorar continuamente por meio da colaboração da comunidade de desenvolvedores. A utilização de software de código aberto nesse contexto é especialmente benéfica, pois permite a disseminação de tecnologias mais acessíveis e eficientes. Além da redução de custos, promove a inovação e a melhoria constante dos sistemas.

Neste sentido, o objetivo deste estudo consiste em uma revisão de sistemas de visão computacional e de código aberto destinados à utilização em drones, no contexto da agricultura de precisão. O método empregado engloba a exploração de fontes de informação como Google Acadêmico (ESCAMILLA et al., 2022), GitHub e GitLab (EXTERMAN, 2021), com a definição de critérios para a seleção e exclusão. Foram identificadas nove investigações pertinentes, todas elas voltadas para a exploração da implementação de softwares de código aberto em drones utilizados na agricultura. Isso nos permitiu efetuar a comparação de funcionalidades dessas soluções. Os resultados evidenciam a escassez de opções disponíveis no âmbito dos softwares de código aberto, apesar das vantagens intrínsecas, como a possibilidade de ajustar algoritmos de acordo com necessidades específicas e o envolvimento ativo da comunidade.

Portanto, para alcançar o propósito do estudo, será apresentada a fundamentação teórica, seguido pela descrição da metodologia utilizada na revisão. A seção de resultados e discussão apresenta os principais trabalhos encontrados nesta revisão. Por fim, às considerações finais do estudo, bem como as referências bibliográficas utilizadas.

FUNDAMENTAÇÃO TEÓRICA

A Inteligência Artificial abrange o estudo e o desenvolvimento de sistemas capazes de realizar tarefas que normalmente requerem a inteligência humana. O Machine Learning é uma área da Inteligência Artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem que os sistemas aprendam com os dados sem serem explicitamente programados. As Redes Neurais são modelos computacionais inspirados no funcionamento do cérebro humano. Eles são compostos por unidades interconectadas, chamadas neurônios,





que processam e transmitem informações. Por fim, Deep Learning é um subcampo do Machine Learning que usa redes neurais artificiais profundas para extrair representações de dados complexas e hierárquicas (AGGARWAL et al., 2022).

De acordo com Foster (2022), as RNNs (Recurrent Neural Networks)são projetadas para processar dados sequenciais, enfatizando a ordem e a dependência temporal. Usado em processamento de linguagem natural, análise de séries temporais e geração de texto. As GANs (Generative Adversarial Networks), consistem em redes neurais concorrentes geradora e discriminadora que competem em um jogo de soma zero. Amplamente utilizado na geração de amostras realistas, como imagens, voz e síntese de música. Já as MLPs (Multilayer Perceptrons) são redes neurais com várias camadas de neurônios, incluindo a camada de entrada, camadas ocultas e camada de saída. Comumente usado em problemas de classificação e regressão. As ResNets (redes residuais) possuem variantes de CNNs que introduzem conexões residuais para melhorar o fluxo de informações e evitar a degradação do desempenho à medida que a rede se torna mais profunda. Amplamente utilizado em tarefas de visão computacional, como classificação de imagens e detecção de objetos. As CNNs (Convolutional Neural Networks) são projetadas para processar dados de grade, como imagens, capturando recursos locais e hierárquicos. Amplamente utilizado em tarefas de visão computacional, como classificação de imagens, detecção de objetos e segmentação semântica. Por fim, temos as VAEs (Variational Autoencoders), que são modelos generativos que aprendem uma representação latente de dados de entrada, válidos para modelar distribuições complexas. Empregado em tarefas de imagem, reconstrução de dados e compressão de informações. E os modelos de Difusão onde simulam a geração iterativa de dados, onde cada etapa depende das ações anteriores. Usado em tarefas de imagem e preenchimento de lacunas em dados ausentes.

METODOLOGIA

Nesta seção, serão apresentados detalhes de como o estudo foi conduzido, incluindo as fontes de informação utilizadas, a estratégia de pesquisa adotada, os critérios de elegibilidade e o procedimento para selecionar os estudos incluídos. Também serão descritos os itens de dados coletados e como os artigos foram selecionados para análise.

Fontes de informação: para o propósito desta revisão, usamos Google Acadêmico, GitHub e GitLab como fontes eletrônicas de dados, conforme mostrado na Tabela 1, com as respectivas URLs. A pesquisa mais recente foi realizada em Junho de 2023. O Google Acadêmico é um mecanismo de busca gratuito e acessível que organiza uma variedade de formatos de publicações acadêmicas científicas e é atualmente o banco de dados mais







abrangente de artigos científicos (MARTÍN-MARTÍN et al., 2021). Já o GitHub e o Git-Lab são repositórios de software de código aberto, pesquisáveis e publicamente disponíveis. O GitHub atualmente é um repositório de código mais utilizado (EXTERMAN, 2021) e também é amplamente utilizado em publicações acadêmicas (ESCAMILLA et al., 2022), seguido pelo GitLab (EXTERMAN, 2021).

Tabela 1 - Fontes de informação usadas nesta revisão e suas respectivas URLs.

Fonte	URL			
Google Acadêmico	https://scholar.google.com			
GitHub	https://github.com			
GitLab	https://gitlab.com			
Fantas Antania mánuia (2022)				

Fonte: Autoria própria (2023)

Estratégia de pesquisa: São definidas as strings de busca de acordo com as fontes de dados a serem pesquisadas, e os termos-chave correspondentes a cada uma das fontes de informação. Os termos-chave são junções definidas dentro da sequência de busca e separadas por aspas. A Tabela 2 apresenta a fonte de informação e as sequências de busca respectivas, incluídos os atributos de exclusão definidas no critério de elegibilidade e procedimentos de seleção. Neste sentido, na linha da base de dados do Google Acadêmico, foram adicionados atributos de exclusão, e, nas linhas das bases de dados do GitHub e do GitLab, as sequências de busca foram definidas e realizadas por tópicos separadas por aspas duplas.

Tabela 2 - Strings de pesquisa usadas para cada fonte de informação.

	, F 1 F		
Fontes	String de busca por base e atributos de exclusão		
Google Acadêmico	"open source" and drones "Computer Vision" or		
	"uav" and "precision agriculture" -simulators		
	-games -farming -simulator -mobile		
GitHub e GitLab	"precision-agriculture" "uav-images"		
	"autonomous-drone" "Computer Vision"		
	Fonte: Autoria própria (2023)		

Fonte: Autoria propria (2023)

Critérios de Elegibilidade e Procedimento de Seleção: após definir a string de busca para cada fonte de informação, foi necessário estabelecer os critérios de inclusão e exclusão. Se um trabalho específico atender a esses critérios, é automaticamente selecionado. O cuidado foi tomado para mesclar possíveis entradas duplicadas.

Os seguintes critérios de inclusão foram considerados para esta revisão:

• Trabalhos focados em aplicação de visão computacional aplicadas aos drones e agricultura de precisão; Trabalhos compostos de software livre e de código aberto com o código-fonte completo publicamente disponível, preferencialmente em repositório







de software como GitHub, GitLab ou equivalente; Trabalhos publicados entre 2019 e 2023; O código-fonte deve ter sido atualizado pelo menos uma vez nos últimos 3 anos (antecedendo Junho de 2023).

A justificativa para selecionar trabalhos publicados desde 2019 baseia-se na disponibilidade e custo de sistemas aéreos não tripulados e tecnologias 4G, essenciais para a agricultura de precisão, que começaram a se tornar populares por volta daquele ano (PACE, 2022; ALKOBI, 2019). Com relação a trabalhos com código atualizado nos últimos 3 anos, este foi um intervalo de tempo selecionado empiricamente para excluir software obsoleto ou abandonado, uma ocorrência comum em projetos de código aberto (KAUR; CHAHAL, 2022). Por sua vez, três critérios de exclusão foram definidos, a saber:

Aplicações não relacionadas a drones e agricultura de precisão; Aplicações de simulador de fazenda, jogos de simulação de fazenda; Outros jogos e sistema para smartphones.

Itens de Dados: Para cada trabalho selecionado, extraímos quatro itens de dados:

• Fonte e ano de publicação; Nome da aplicação; Interface(s) do usuário e Linguagem(s) de programação; Repositório do código-fonte.

Os repositórios de código-fonte para os trabalhos selecionados são encontrados diretamente nas buscas realizadas no GitHub e GitLab, ou no artigo de pesquisa associado encontrado no Google Acadêmico. O ano de publicação é retirado da data de publicação do artigo de pesquisa, ou, no caso de o trabalho não ter um artigo associado, o ano do primeiro commit no respectivo repositório de código-fonte.

A linguagem de programação em que uma aplicação é desenvolvida, embora seja principalmente de interesse dos desenvolvedores, é especialmente importante em software livre e de código aberto, que pode ser livremente modificado, melhorado e estendido. Desenvolvedores ou usuários com habilidades técnicas devem considerar preferencialmente o software desenvolvido em uma linguagem com a qual estejam familiarizados, especialmente se pretendem modificar o código-fonte e/ou implementar novos recursos. O software pode ser desenvolvido em mais de uma linguagem, e isso também deve ser considerado no processo de seleção, pois pode aumentar a complexidade do desenvolvimento.

Para fins de revisão, as linguagens de programação foram determinadas a partir do recurso "Linguagens" do GitHub, que detalha as linguagens usadas em um projeto por porcentagem, ou, se essas informações eram insuficientes ou inexistentes, por meio da exploração manual do código-fonte. Consideramos apenas as linguagens usadas para desenvolver os







projetos, ignorando as linguagens usadas exclusivamente para documentação (por exemplo, HTML ou Markdown) ou para o processo de construção (por exemplo, scripts Bash ou Makefiles).

A interface do usuário define a maneira como o usuário final interage com o software em questão. Pode ser uma interface de frente, como uma interface gráfica ou de texto, ou uma interface de fundo, trabalhando como uma biblioteca ou como um plugin para outra ferramenta. A interface do usuário pode limitar o tipo de usuários finais de um determinado pacote. Por exemplo, as interfaces de texto, que permitem que uma ferramenta seja controlada por meio da linha de comando, assim como as bibliotecas, são tipicamente mais adequadas para desenvolvedores e/ou usuários com habilidades técnicas. Várias ferramentas oferecem mais de um tipo de interface, ampliando assim o escopo de usuários potenciais.

Seleção dos Trabalhos: A Tabela 3 apresenta a quantidade de trabalhos encontrados em suas respectivas bases eletrônicas. No total, foram identificados 306 artigos no Google Acadêmico e 29 projetos nos repositórios GitHub e GitLab, perfazendo um total de 335 trabalhos reconhecidos usando as strings de busca descritas na Subseção de estratégia de pesquisa. Após aplicar os critérios de filtragem, foram excluídos 297 artigos da base do Google Acadêmico e 29 do GitHub e GitLab, totalizando 326 trabalhos excluídos. Portanto, o cenário é apresentado com 09 trabalhos, correspondendo à proposta definida nos objetivos da pesquisa.

Tabela 3 - Número de trabalhos identificados com as strings de busca para cada fonte de informação, bem como número de trabalhos excluídos e incluídos após a aplicação dos critérios de filtragem. O valor em negrito corresponde ao número de trabalhos selecionados para esta revisão.

Info. fonte	Identificado	Excluídos	Incluídos
Google Acadêmico	306	297	09
GitHub	28	28	0
GitLab	1	1	0
Totals	335	326	09

Fonte: Autoria própria (2023)

RESULTADOS E DISCUSSÃO

Analisamos as obras mais relevantes, resumindo pelas características conforme Tabela 4 com itens, referência bibliográfica, nome abreviado, linguagem de programação e links para repositórios de código-fonte. A tabela fornece uma visão geral do trabalho, permitindo entender a abordagem e as tecnologias utilizadas nos estudos revisados. Além disso, é feita uma breve descrição do trabalho desenvolvido.





Tabela 4 - Tabela de Softwares de Navegação Autônoma para Drones

1				Repositório
4			guagem	
1	(OCA; FLORES,	The AgriQ	Textual/Python	github.com/LAPyR/NDVI
	2021)			with-modified-Mobius-
				cameras
2	(STEWART et	Torchgeo	Biblioteca/Python	(github.com/microsoft/
	al., 2022)			torchgeo
3	(RIIHIAHO;	Desmiler	Textual/Python	(github.com/silmae/
	ESKELINEN;			desmiler
	PÖLÖNEN,			
	2021)			
4	(KOTIS;	Traj4Drones	Textual/Python	(github.com/KotisK/
	SOULARIDIS,			ReconTraj4Drones>
	2023)			
5	(CASAPIA et al.,	Classificação	Textual/R	(github.com/xime377/
	2019)	UAV		UAV-classification>
6	(GURGU;	WildNav	Gráfica / Python	(github.com/TIERS/
	QUERALTA;			$\operatorname{wildnav} angle$
	WESTERLUND,			
	2022)			
7	(KERN et al.,	Open	Biblioteca/ C++	(github.com/
	2020)	REALM		laxnpander/
				OpenREALM
8	(LI et al., 2021)	See Cu-	Texto/ Python	(github.com/joanlyq/
		cumbers		SeeCucumbers
9	(YANG et al.,	MoDaCS	Gráfica / Python	(github.com/
	2020)			TESTgroup-BNL/
			oria prápria (2023)	MoDaCS〉

Fonte: Autoria própria (2023)

Segundo Oca e Flores (2021), Tabela 4, item [1], utiliza-se um sistema de espectro duplo para capturar imagens aéreas de culturas, combinando câmera Mobius modificada e software de código aberto. As etapas incluem retificação da imagem, georreferenciação, reconstrução ortomosaica, calibração da refletância e cálculo de índices de vegetação.

No trabalho Stewart (2022), Tabela 4, item [2], utiliza métodos de aprendizado profundo (deep learning) e a biblioteca TorchGeo para integrar dados geoespaciais no contexto do PyTorch. A abordagem busca facilitar a aplicação de deep learning em tarefas de sensoriamento remoto, com modelos pré-treinados para imagens multiespectrais e recursos para carregamento e transformação de dados.

MINISTÉRIO DA

EDUCAÇÃO

Os autores Riihiaho e Eskelinen (2021), Tabela 4, item [3], desenvolveram um soft-







ware de código aberto para corrigir a erros espectrais, conhecida como "smile" e o desalinhamento chamado de "tilt" em uma imagem hiperespectral. Os resultados mostraram que é possível corrigir essas distorções sem a necessidade de um laboratório óptico.

O trabalho de Kotis (2023), Tabela 4, item [4] apresenta uma estrutura para reconstrução, modelagem e enriquecimento de trajetórias semânticas de drones. Através do uso de dados georreferenciados e ferramentas de análise, é possível obter informações contextuais significativas sobre o movimento dos drones.

No estudo de Casapia (2019), Tabela 4, item [5], foi desenvolvido um fluxo de trabalho de classificação baseado em objetos para imagens de Veículo Aéreo Não-Tripulado de alta resolução. A abordagem combinou processamento de imagem e funcionalidades de SIG para identificar e delinear as copas das palmeiras na floresta tropical.

O trabalho de Guru e Queralta (2022), Tabela 4, item [6], apresenta um algoritmo de localização baseado em visão computacional para drones em ambientes naturais. Utilizamos recursos de deep learning para calcular as coordenadas geográficas do Veículo Aéreo Não-Tripulado, correspondendo características visuais de fotografias RGB capturadas pela câmera do drone a seções de um mapa pré-construído de imagens de satélite georreferenciadas.

O trabalho de Kern (2020), Tabela 4, item [7], apresenta o OpenREALM é um framework para Drones que permite mapeamento em tempo real. Com uma câmera conectada ao Drone, ele cria mosaicos de alta resolução e informações 3D precisas. O mapeamento pode ser feito em terrenos planos ou complexos, usando a posição global do drone para georreferenciar os dados.

Segundo Li (2021), Tabela 4, item [8], esses detectores de objetos têm sido amplamente usados em diversas áreas, e um estudo piloto com o YOLOv3 mostra o potencial dessa abordagem. Com a georreferência das imagens, a detecção automatizada pode agilizar a análise de distribuição e possibilitar pontos valiosos ao combinar dados com outros mapeamentos.

O trabalho de Yang (2020), Tabela 4, item [9], apresenta desenvolvimento do Osprey, um drone econômico que fornece imagens e medições hiperespectrais das copas de vegetação. Isso permite um mapeamento preciso dos tipos de plantas e revela informações detalhadas sobre as propriedades do vegetal.

Os trabalhos encontrados demonstram a relevância e avanços no uso de drones e sensoriamento remoto para estudar a vegetação e ecossistemas terrestres. As diferentes abordagens, como detecção automatizada de objetos, mapeamento preciso de copas de vegetação







e correção de erros espectrais, destacam a flexibilidade e adaptabilidade dessas tecnologias. Essas ferramentas oferecem vantagens significativas para análises ambientais mais eficientes e precisas, tornando-se importantes aliadas em pesquisas científicas e tomadas de decisão em diversos campos, desde a agricultura de precisão até a conservação ambiental.

CONSIDERAÇÕES FINAIS

A revisão realizada sobre softwares livres e de código aberto de visão computacional voltados para drones na agricultura de precisão revelou uma seleção de nove trabalhos de destaque. Os trabalhos identificados oferecem recursos essenciais, como processamento de imagens, detecção de objetos, segmentação de culturas e análise de saúde das plantas. Essas funcionalidades são de extrema importância para otimizar as práticas agrícolas. No entanto, é importante destacar que se observou a falta de trabalhos que disponibilizam os códigos-fonte dos softwares em plataformas de acesso aberto. Essa ausência pode dificultar a replicação e o aprofundamento das pesquisas, prejudicando a transparência e a confiabilidade dos resultados obtidos. Outro ponto relevante é que alguns dos trabalhos revisados apresentaram falta de atualizações adequadas em seus códigos ao longo do tempo. A rápida evolução da tecnologia exige a constante adaptação dos softwares para garantir sua eficácia e relevância contínuas. Além disso, a natureza aberta dessas ferramentas possibilita a colaboração e o aprimoramento contínuo, impulsionando o avanço da agricultura de precisão e tornando essas tecnologias acessíveis a mais agricultores. Portanto, futuras pesquisas e desenvolvimentos nessa área devem incentivar a disponibilização dos códigos-fonte para ampliar a disseminação do conhecimento e fomentar a colaboração entre os pesquisadores.

REFERÊNCIAS

AGGARWAL, K. et al. Has the future started? the current growth of artificial intelligence, machine learning, and deep learning. *Iraqi Journal for Computer Science and Mathematics*, v. 3, n. 1, p. 115–123, 2022.

ALKOBI, J. *The Evolution of Drones: From Military to Hobby & Commercial.* 2019. Percepto: (https://percepto.co/the-evolution-of-drones-from-military-to-hobby-commercial/). Accessed on March 5, 2023.

CASAPIA, X. T. et al. Identifying and quantifying the abundance of economically important palms in tropical moist forest using uav imagery. *Remote Sensing*, MDPI, v. 12, n. 1, p. 9, 2019.

ESCAMILLA, E. et al. The rise of github in scholarly publications. In: SILVELLO, G. et al. (Ed.). *Linking Theory and Practice of Digital Libraries: Proceedings of the 26th International Conference on Theory and Practice of Digital Libraries*. [S.l.]: Springer International Publishing, 2022. (TPDL 2022), p. 187–200.







EXTERMAN, D. *GitLab vs GitHub – A 2022 Comparison*. 2021. Incredibuild: (https://www.incredibuild.com/blog/gitlab-vs-github-comparison). Accessed on March 7, 2023.

GURGU, M.-M.; QUERALTA, J. P.; WESTERLUND, T. Vision-based gnss-free localization for uavs in the wild. In: 2022 7th International Conference on Mechanical Engineering and Robotics Research (ICMERR). [S.l.: s.n.], 2022. p. 7–12.

HONG, Y. et al. Autonomous mission of multi-uav for optimal area coverage. *Sensors*, MDPI, v. 21, n. 7, p. 2482, 2021.

KAUR, R.; CHAHAL, K. K. Exploring factors affecting developer abandonment of open source software projects. *Journal of Software: Evolution and Process*, Wiley Online Library, v. 34, n. 9, p. e2484, 2022.

KERN, A. et al. Openrealm: Real-time mapping for unmanned aerial vehicles. In: 2020 International Conference on Unmanned Aircraft Systems (ICUAS). [S.l.: s.n.], 2020. p. 902–911.

KOTIS, K.; SOULARIDIS, A. Recontraj4drones: A framework for the reconstruction and semantic modeling of uavs' trajectories on movingpandas. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 13, n. 1, p. 670, 2023.

LI, J. Y. Q. et al. Seecucumbers: Using deep learning and drone imagery to detect sea cucumbers on coral reef flats. *Drones*, v. 5, n. 2, 2021. ISSN 2504-446X. Disponível em: (https://www.mdpi.com/2504-446X/5/2/28).

LU, Y.; YOUNG, S. A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, Elsevier, v. 178, p. 105760, 2020.

MARTÍN-MARTÍN, A. et al. Google scholar, microsoft academic, scopus, dimensions, web of science, and opencitations' coci: a multidisciplinary comparison of coverage via citations. *Scientometrics*, Springer, v. 126, p. 871–906, 2021.

OCA, A. M. de; FLORES, G. The agriq: A low-cost unmanned aerial system for precision agriculture. *Expert Systems with Applications*, Elsevier, v. 182, p. 115163, 2021.

PACE, L. *4G: History, Origin, and More.* 2022. History-Computer: (https://history-computer.com/4g-guide/). Accessed on March 5, 2023.

RIIHIAHO, K. A.; ESKELINEN, M. A.; PÖLÖNEN, I. A do-it-yourself hyperspectral imager brought to practice with open-source python. *Sensors*, MDPI, v. 21, n. 4, p. 1072, 2021.

STEWART, A. J. et al. Torchgeo: deep learning with geospatial data. In: *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*. [S.l.: s.n.], 2022. p. 1–12.

YANG, D. et al. A multi-sensor unoccupied aerial system improves characterization of vegetation composition and canopy properties in the arctic tundra. *Remote Sensing*, MDPI, v. 12, n. 16, p. 2638, 2020.

