



**MOISÉS DE SOUZA FEITOSA**

**INOVAÇÕES TECNOLÓGICAS BASEADAS EM  
FERRAMENTAS DE BAIXO CUSTO PARA  
IMAGEAMENTO DE ORGANISMOS DO SOLO  
ASSISTIDO POR VISÃO COMPUTACIONAL**

**LAVRAS-MG**

**2021**

**MOISÉS DE SOUZA FEITOSA**

**INOVAÇÕES TECNOLÓGICAS BASEADAS EM FERRAMENTAS DE BAIXO  
CUSTO PARA IMAGEAMENTO DE ORGANISMOS DO SOLO ASSISTIDO  
POR VISÃO COMPUTACIONAL**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência do Solo na área de concentração Biologia, Microbiologia e Processos Biológicos do Solo, para obtenção do título de Mestre.

Prof. Dr. Teotonio Soares de Carvalho

Orientador

**LAVRAS-MG**

**2021**

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Feitosa, Moisés de Souza.

Inovações tecnológicas baseadas em ferramentas de baixo custo para imageamento de organismos do solo assistido por visão computacional / Moisés de Souza Feitosa. - 2021.

101 p.

Orientador(a): Teotonio Soares de Carvalho.

Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2021.

Bibliografia.

1. Microbiologia do Solo. 2. Aprendizado de Máquina. 3. Ciência Aberta. I. Carvalho, Teotonio Soares de. II. Título.

**MOISÉS DE SOUZA FEITOSA**

**INOVAÇÕES TECNOLÓGICAS BASEADAS EM FERRAMENTAS DE BAIXO  
CUSTO PARA IMAGEAMENTO DE ORGANISMOS DO SOLO ASSISTIDO  
POR VISÃO COMPUTACIONAL**

**TECHNOLOGICAL INNOVATIONS BASED ON LOW-COST TOOLS FOR  
SOIL ORGANISMS IMAGING AUXILIATED BY COMPUTER VISION**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência do Solo na área de concentração Biologia, Microbiologia e Processos Biológicos do Solo, para obtenção do título de Mestre.

APROVADA em 15 de de dezembro de 2021.

Profa. Dra. Fátima Maria de Souza Moreira – UFLA

Profa. Dra. Elaine Martins da Costa – UFPI

Dr. Willian César Terra – UFLA

Prof. Dr. Teotonio Soares de Carvalho

LAVRAS- MG

2021

A todos aqueles que se dedicam ao desenvolvimento da ciência brasileira,

Dedico

## AGRADECIMENTOS

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

À Deus, pela força dada até o presente momento.

Ao professor orientador Teotonio Soares de Carvalho, por todo o ensinamento sempre calmamente transmitido, apoio, confiança, e, sobretudo, tratamento humano para comigo ao longo de todo este período.

À minha família e à minha companheira Lígia pelo apoio as decisões tomadas que me levaram até aqui.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela concessão da bolsa e verbas sem a quais não seria possível o desenvolvimento deste projeto, bem como as demais agências de fomento (Fapemig e Capes).

À sociedade brasileira por ter proporcionado a condução deste trabalho e minha formação em Ciência do Solo.

Ao companheiro de laboratório, Matheus Nicoletti, pelo apoio, especialmente, nas várias dúvidas que surgiram na condução de testes que envolviam fungos micorrízicos arbusculares, e pelas inúmeras caronas que evitaram as cansativas escaladas de morro para chegar à UFLA.

Aos docentes (Alex, Carlos Alberto, Fátima, Júnior César, Marco Aurélio, Maria Lígia, Michele, Sérgio Henrique, Teotonio e Yuri) com os quais tive o prazer de aprender ao longo das disciplinas ensinamentos que serão fundamentais para minha atuação profissional.

À Universidade Federal de Lavras, em especial ao Departamento de Ciência do Solo, pela infra-estrutura fornecida para as disciplinas e condução dos experimentos.

À Willian César Terra pelas lâminas contendo raízes parasitadas por nematoides utilizadas nesse experimento.

Ao Laboratório de Micorrizas da FURB pela disponibilização de lâminas de raízes micorrizadas para testes que serviram como base para parte desta dissertação.

A todos aqueles que gentilmente partilham seu conhecimento na internet através de vídeos, tutoriais em texto e participação em fóruns. Por diversas vezes estes desconhecidos ajudaram a esclarecer dúvidas que surgiram ao longo da condução das tarefas que envolviam programação conduzidas nesta dissertação.

“E que a atitude de recomeçar é todo dia,  
toda hora, é se respeitar na sua força e fé,  
e se olhar bem fundo até o dedão do pé”

## RESUMO

Os equipamentos e análises utilizados no estudo de organismos do solo estão em constante evolução. Rotineiramente são utilizados microscópios para a condução de algumas destas análises. Embora bastante difundidos, estes equipamentos podem possuir custo bastante elevado. Um dos componentes que pode contribuir significativamente para esta realidade é o estágio automatizado, que, apesar de sua grande utilidade, pode custar até dezenas de milhares de reais. Uma das estratégias que podem ser utilizadas para superar esta barreira é a utilização de estágios automatizados desenvolvidos no escopo da ciência aberta. Com a adoção desta abordagem o pesquisador pode montar o próprio estágio através de softwares e hardwares livres, o que torna a aquisição de um microscópio com estágio automatizado (que aumentará a capacidade de análise em relação a um microscópio tradicional) mais favorável em termos financeiros. Além disso, há a possibilidade de personalização do equipamento às condições locais (de disponibilidade de peças que venham a compor o microscópio), bem como as condições da amostra. Esta adaptabilidade se torna fundamental no estudo de nematoides parasitas de plantas já que, tradicionalmente, a visualização de raízes parasitadas se dá em lâminas não convencionais, de maiores dimensões, não compatíveis com as dimensões dos estágios usualmente utilizados em microscópios comuns. Há de se ter em mente que um dos usos principais que se dá a um estágio automatizado é o escaneamento digital de amostras. Nesta estratégia, múltiplas imagens (que representam a totalidade da amostra ou de uma região de interesse) são geradas sendo necessário uma estratégia de junção destas imagens, de forma que, em última instância, seja ampliado o campo de visão em relação à amostra. Além de usar equipamentos em constante mudança, a análise de organismos do solo também está sujeita a mudanças nas metodologias utilizadas. Recentemente vem crescendo o uso de ferramentas de aprendizado de máquina nestas análises como por exemplo no estudo de bactérias, nematóides e fungos presentes no solo, já que uma série de análises outrora realizadas manualmente, repetidamente, que necessitam de mão-de-obra altamente especializadas e dispendiosas em termo de tempo podem ser substituídas pelo uso de algoritmos treinados para cada uma destas tarefas. Visando contribuir para a popularização do uso de estágios automatizados e de estratégias que envolvem visão computacional (uma sub-área do aprendizado de máquina) apresentamos nesta dissertação dois estágios automatizados de baixo custo (o primeiro reaproveitando um microscópio de campo claro presente no Laboratório de Microbiologia do Solo da Universidade Federal de Lavras, e o segundo um microscópio de baixo custo com estágio automatizado que proporciona o escaneamento total de lâminas contendo raízes parasitadas por nematoides), algoritmo de medição de nematoides contidos em imagens, bem como de seleção de regiões na lâmina escaneada que continham estes organismos para posterior junção e ampliação do campo de visão destas regiões e um algoritmo de segmentação de hifas de fungo micorrízico arbuscular presentes em imagens de raízes. Para a confecção dos softwares de controle dos estágios e para os algoritmos de seleção e medição dos nematoides, foi utilizada a linguagem de programação Python. Para o algoritmo de segmentação das hifas também foi utilizada a mesma linguagem. Foram obtidos resultados satisfatórios na reconstrução de regiões contendo nematoides, bem como para as medições destes organismos e a segmentação das hifas de fungos micorrízicos. Acreditamos que nossa abordagem e os resultados aqui apresentados são importantes para a popularização do uso de estágios automatizados e de ferramentas de aprendizado de máquina para o estudo de organismos do solo.

**Palavras chave:** Aprendizado de máquina, fungos micorrízicos arbusculares, nematoide, ciência aberta, microscopia.



## ABSTRACT

The machinery and analysis used in soil organisms studies have been in constant evolution. Usually, microscopes are used to carry out some of those analyses. Though commonly employed in labs, this equipment is usually expensive. One of the components which could be important to the elevated cost of a microscope is the automatized stage. That cost can be up to dozens of thousands of dollars. One strategy that can be used to overcome that barrier is acquiring the stage within the scope of Open Science. With that approach, researchers can assemble their own equipments using softwares and hardwares of open source. That approach will reduce costs drastically to acquire an automatized stage (which will increase the analysis capacity when compared to traditional microscopes). Moreover, the automatized stage can be adapted to local conditions (for example parts of an old microscope can be reutilized, then reducing costs), as well as to sample's particularities. This versatility is fundamental in plant parasites nematodes. Traditionally those organisms have been observed in non-conventional microscope slides (wider and longer than conventional ones). That reality makes the observation of nematodes under a microscope more difficult, because the stages of microscopes usually are smaller than the slides dimensions. We have to bear in mind that one of the main uses we can employ to an automatized stage is scanning samples automatically (that means, capturing images from the totality of the sample). In this approach, multiple images (which represent the totality of the sample or only a region) are acquired. It is also necessary a strategy to stitch those images, so that, in the end, the visual field can be amplified by combining multiple images into a single one. Besides that, methods used in soil organisms analyses are in constant change. Recently, machine learning tools have been utilized in the study of a variety of organisms such as bacteria, fungi, and nematodes. This approach is advantageous, particularly in analyses that are labour-intensive and/or need a specialized worker. In that scenario, trained (and tested) algorithms can be deployed to perform those tasks. Aiming to contribute to the popularization of automatized stages and of computer vision (a sub-area of machine learning) approaches to soil organisms analyses, we present in this dissertation two low-cost automatized stages (the first one reuses an old bright-field microscope, while the second one is based on a low-cost digital microscope that has enough resolution to observe nematodes in parasitized soybean roots). Adding to that, we present here algorithms that make some measures on nematodes images, select images with nematodes to posterior stitching to create images of regions containing those organisms in order to amplify the visual field over the sample, and an algorithm to perform segmentation of arbuscular mycorrhizal fungi's hyphae. To write both softwares of control stages, we utilized the programming language Python. For the hyphae's segmentation algorithm we also used a Python code. It was obtained satisfactory results in imaging mostly regions containing nematodes with both high resolution and wide vision field, as well as for the measurements of nematodes in images, and hyphae segmentation. Altogether, we believe, it is presented here a cost-effective approach that can popularize the use of automatized stages and machine learning tools for the study of soil organisms.

**Keywords:** Computer Vision, Arbuscular Mycorrhizal Fungi, Nematode, Open Science, Microscopy

## SUMÁRIO

1. Introdução Geral.....	13
2. Referencial Teórico .....	14
2.1 Ciência Aberta (CA).....	14
2.2 Visão Computacional (VC).....	16
2.3 Aprendizagem Profunda (AP) e Inteligência Artificial (AI) .....	16
2.3.1 Histórico .....	16
2.3.2 Aprendizagem Profunda (AP).....	18
2.4. Redes neurais convolucionais (RNC): .....	19
2.5. UNet.....	21
2.6. Aplicações de aprendizado de máquina .....	22
2.6.1 Aplicações na genômica e proteômica .....	23
2.6.2 Aplicações na biologia e agricultura .....	23
3. Referências.....	27
ARTIGO I.....	33
1. Introdução.....	34
2. Metodologia .....	35
2.1 - Estágio automatizado acoplado a microscópio de campo claro (MicroStage) .....	35
2.2 Microscópio digital de baixo custo acoplado a um estágio automatizado para escaneamento de amostras de maiores dimensões (NemaStage).....	36
2.3 Preparo das amostras biológicas.....	38
2.3.1 Amostras de fungo micorrízico arbuscular e escaneamento .....	38
2.3.2 Amostras de raízes infectadas por nematóides.....	38
3. Uso de StageControl.....	39
4. Aplicação do método .....	40
4.1 Uso de MicroStage.....	40
4.1.1 Limitações da abordagem.....	42
4.2 Uso de NemaStage .....	42
4.2.1 Limitações da abordagem.....	45
5. Custos de NemaStage .....	45
ARTIGO II .....	48
1. Introdução.....	49
2. Metodologia .....	50
2.1 Obtenção de juvenis de segundo estágio (J2) de <i>Meloidogyne javanica</i> .....	50
2.2 Penetração de juvenis de segundo estágio (J2) de <i>M. javanica</i> em raízes de soja.....	51

<b>2.3 Microscópio, estágio automatizado e software de controle</b>	51
<b>2.4 Segmentação e obtenção de medidas de nematoides em imagens</b>	52
2.4.1 Segmentação de nematoides	52
2.4.2 Medição dos nematoides	53
<b>2.5 Detecção de imagens contíguas de nematoides e junção</b>	54
<b>3. Resultados e Discussão</b>	56
<b>4. Conclusões</b>	65
<b>ARTIGO III</b>	69
<b>1. Introdução</b>	70
<b>2. Metodologia</b>	72
2.1 Obtenção de raízes micorrizadas	72
2.2 Obtenção das imagens de hifas de FMA	73
2.3 Rede Neural Convolutacional para extração de estruturas de FMA	74
<b>3. Resultados e Discussão</b>	76
<b>4. Conclusões</b>	81
<b>Considerações Finais</b>	85
<b>Apêndices</b>	87
Apêndice 1	87
Apêndice 2	87
Apêndice 3	89
Apêndice 4	89
Apêndice 5	91
Apêndice 6	94
Apêndice 7	95
Apêndice 8	97
Apêndice 9	99
Apêndice 10	99
Apêndice 11	99

## 1. Introdução Geral

No estudo de organismos do solo frequentemente se faz necessário o uso de equipamentos que permitam observá-los tendo em mente que estes, muitas vezes, não são visíveis à olho nu. Dentre os equipamentos mais utilizados nestas análises está o microscópio. Desde os primeiros microscópios desenvolvidos no século XVII por Hooke e Van Leeuwenhoek até os dias atuais, este equipamento tem sofrido uma série de mudanças (WOLLMAN et al., 2007). Um dos componentes do microscópio que mais tem mudado ao longo do tempo é o estágio.

Atualmente este componente pode ser automatizado, permitindo que as lâminas de microscopia sejam imageadas por total (ou parcialmente) sem a necessidade de operação manual, o que aumenta a precisão do imageamento obtido. Para alcançar este objetivo, imagens são captadas em posições diferentes (representando cada ponto da lâmina) e posteriormente juntas, reconstruindo regiões da amostra ou a amostra por completo. Um dos maiores limitantes ao uso deste equipamento é o custo financeiro. Em pesquisa de mercado realizada em julho de 2021 foram encontrados estágios automatizados que custavam mais de R\$40000,00. Uma das abordagens que podem ser utilizadas para superar esta barreira é obter este equipamento no escopo da ciência aberta. Desta forma são utilizados softwares e hardwares de código livre para confecção e uso de estágios automatizados (WOELFLE et al., 2011), o que facilita a adoção desta tecnologia por um número maior de centros de pesquisa através do custo reduzido.

Além de novos equipamentos, uma série de novas ferramentas vêm sendo utilizadas na biologia e ciências agrárias. Ao longo da última década algoritmos de aprendizado de máquina, sobretudo os que utilizam aprendizado profundo, são utilizados nas mais variadas análises de amostras biológicas como plantas, mitocôndrias e gado (CARRANZA-ROJAS et al., 2017; FISCHER et al., 2020; WEBER et al., 2020). Estas ferramentas também são utilizadas no estudo de organismos importantíssimos do solo como fungos micorrízicos arbusculares e nematóides (EVANGELISTI et al., 2021; THEVENNOUX et al., 2021). Esse grupo de fungos notadamente estabelece simbiose com uma ampla variedade de plantas (WANG e QIU, 2006) proporcionando uma série de efeitos benéficos como a melhoria da capacidade da planta lidar com estresses e maior produção de biomassa (FONSECA et al., 2019; LI et al., 2019). Os nematóides, a depender da espécie, podem ser parasitas de plantas, causando prejuízos na ordem de bilhões de dólares na agricultura mundial anualmente (MESA-VALLE et al., 2020). Através de metodologias que envolvam programação e algoritmos de aprendizado de máquina, a análise de amostras biológicas pode ser revolucionada. Por exemplo, a precisão de uma análise pode ser aumentada quando comparada à realizada por observadores humanos (HAKIM et al., 2018) além de ocorrerem mais rapidamente que as análises tradicionais.

Nesta dissertação propomos abordagens de baixo custo que envolvem aprendizado de máquina e uso de softwares e hardwares de código aberto para o estudo de fungos micorrízicos arbusculares e nematóides parasitas de plantas. Nomeadamente, apresentamos: i) Dois estágios automatizados de baixo custo. Um para uso em imageamento de fragmentos de raízes colonizadas por fungo micorrízico arbuscular e o outro para imageamento de nematóides contidos em raízes; ii) Software de acesso livre para o controle destes estágios; iii) Código para medidas de nematóides; iv) Código de junção de imagens para compor regiões que continham nematóides aumentando assim o campo de visão sob a amostra e diminuindo o número de imagens a serem avaliadas; v)

Algoritmo de segmentação de hifas presentes em raízes de milheto colonizadas por fungo micorrízico arbuscular.

## 2. Referencial Teórico

### 2.1 Ciência Aberta (CA)

Ciência aberta é um termo utilizado para designar um movimento que objetiva que o conhecimento científico, seja na sua produção ou divulgação, seja o mais acessível possível para todos os interessados, quer sejam eles o público fora da academia ou os pesquisadores (WOELFLE et al., 2011). Outros pontos importantes nesse movimento são a transparência que deve ocorrer na geração do conhecimento, a promoção da comunicação entre pesquisadores de forma que o conhecimento adquirido seja compartilhado e o livre acesso da comunidade interessada aos dados que subsidiaram o resultado obtido (ALLEN e MEHLER, 2019). Por transparência é entendido que todos os passos para a obtenção de um resultado devem estar abertos para o público de forma que seja possível sua reprodução. Nesse contexto são usados e criados softwares de uso livre (FORTUNATO e GALASSI, 2021). O uso dessas ferramentas de uso livre possibilita que um número maior de usuários possa utilizar a mesma metodologia utilizada por um determinado grupo de pesquisa, o que aumenta a reprodutibilidade da pesquisa.

Para que a transição entre a maneira tradicional de se fazer ciência para uma abordagem no escopo da ciência aberta ocorra, alguns incentivos por parte da comunidade científica devem ser utilizados. Estudos já demonstram que resultados “nulos” (sem efeito positivo dos tratamentos avaliados), quando utilizam os princípios da CA, têm maior chance de publicação em periódicos científicos do que aqueles que utilizam abordagens tradicionais (ALLEN e MEHLER, 2019). Outra forma de incentivo pode vir através dos financiadores e das revistas científicas. Dar prioridade para quem faz ciência de maneira transparente, seja com selos ou mesmo como prioridade de financiamento, são passos que podem levar a uma maior aceitação da comunidade científica no uso desse escopo (MUNAFÒ et al., 2017).

Dentro da estratégia de proporcionar que o maior número possível de pesquisadores possa utilizar metodologia próxima a realizada no experimento original, é importante utilizar as ferramentas de menor custo possível que mantenham um bom resultado de análise e/ou ferramentas que sejam de livre acesso. Um exemplo disso é o uso de impressoras 3D de baixo custo. Com o uso desse equipamento é possível que o pesquisador faça uso de uma série de novas metodologias, já que é possível imprimir as peças necessárias para a condução do experimento, além de poder adaptá-las para a disponibilidade de material e requerimentos da amostra que está sendo avaliada. Também é importante que os códigos e desenhos para a confecção dessas peças fiquem disponíveis para livre acesso. Assim, pesquisadores de todo o mundo têm a capacidade de reproduzir as peças utilizadas no experimento anterior e aplicá-las para o estudo de interesse (ZHANG et al., 2016). Essa tentativa de disponibilizar hardwares de livre acesso é um tópico por si só dentro da ciência aberta, a *open hardware*.

A Open Source Hardware Association (OSHA) define *open hardware* como o hardware cujo todo o processo de criação é aberto ao público. Isto significa que o usuário, utilizando como base este hardware, pode modificar, estudar, distribuir, fazer e vender o hardware baseado nesse primeiro hardware “molde”. Com o uso dessas ferramentas uma

série de laboratórios podem conduzir experimentos que não poderiam antes graças aos custos diminuídos e possibilidade de produzir o próprio equipamento.

Além de hardwares de uso livre, é importante a disseminação de softwares abertos. Esses softwares podem ser acessados e utilizados sem custo adicional por toda a comunidade interessada. Uma série de projetos nas últimas duas décadas são direcionados a esse fim. No que diz respeito a esses softwares, há um movimento em estágio muito mais avançado em termos de utilização pela comunidade científica em geral quando comparado aos hardwares de código livre.

Um dos exemplos mais conhecidos de sucesso no uso e disseminação de software livre é o uso da IDE (*integrated development environment*) RStudio (RStudio Team, 2020). Esse software utiliza a linguagem R, criada por Ross Ihaka e Robert Gentleman, principalmente para análises estatísticas, análise de dados e confecção de gráficos. Hoje esse software é utilizado e aceito devido a sua precisão nas análises e possibilidade de proporcionar uma clareza para o público e revisores de artigos de como os dados foram tratados e as análises feitas.

Outro exemplo bastante utilizado pela comunidade científica é o FIJI (ImageJ) (SCHNEIDER et al., 2012). Esse software faz parte de um projeto da Universidade de Wisconsin nos Estados Unidos. Esse programa tem como objetivo principal a análise de imagens digitais, sejam elas na ciência de materiais, ciências biológicas, mineralogia, medicina e uma miríade de outras possíveis aplicações. Ainda no campo da análise de imagens podemos destacar o projeto VIA da universidade de Oxford (DUTTA e ZISSERMAN, 2019) e o projeto Apeer da empresa alemã Carl Zeiss.

Outro importante ponto da ciência aberta é o livre acesso a códigos de programação. Esses códigos podem ou não fazer parte de softwares e hardwares. Uma série de ferramentas estão à disposição onde códigos são depositados. A mais famosa dessas ferramentas é o repositório Github. Nesse repositório há possibilidade de contato entre o público e o autor do código. Dessa forma é possível a correção de problemas, ajuste do código para a realidade do projeto e a elucidação de dúvidas de maneira geral.

Uma das ferramentas de livre acesso mais importante no que toca à aprendizagem profunda é o Google Colab. Esse ambiente digital serve como uma IDE baseada em Jupyter Notebook. A característica que faz essa IDE ser revolucionária é a possibilidade de o usuário utilizar uma GPU (acelerador de hardware) disponibilizado pelo Google gratuitamente. Dessa forma é possível o treinamento de modelos de *machine learning* em um tempo muito menor.

Em última instância todos esses recursos podem ser utilizados pelo maior número possível de interessados, o que torna a reprodutibilidade do experimento muito maior, enquanto é aumentada a transparência e confiabilidade do resultado obtido com custos cada vez menores o que aumenta a probabilidade de uso desses recursos pelo público interessado, que passam a produzir mais conhecimento através de ferramentas abertas ao público, agindo como um ciclo que se retroalimenta.

## 2.2 Visão Computacional (VC)

Um dos ramos da computação com crescente interesse por parte da iniciativa privada, comunidade científica e população no geral é a visão computacional (VC). Entende-se por visão computacional as técnicas envolvidas na aquisição, processamento, análise e entendimento realizados sob imagens digitais (WILEY e LUCAS, 2018).

O objetivo final da visão computacional é mimetizar o que ocorre quando um ser humano analisa uma imagem, ou mesmo superar a visão humana em algumas tarefas. Ou seja, a visão computacional objetiva que uma máquina que avalia uma determinada imagem seja capaz de diferenciar os diferentes objetos contidos nela, classificá-los e identificar onde estão os objetos de interesse (CERNEK et al., 2020). O uso dessa tecnologia se torna imprescindível sobretudo para análise de estruturas de difícil diferenciação pelo ser humano, ou que necessite elevada especialização por parte do observador (VÁZQUEZ et al., 2017). Erros na análise, sobretudo na medicina, podem levar a resultados extremamente indesejáveis. Além disso, o uso de máquinas leva a uma maior capacidade de processamento e análise de dados quando comparada ao uso de observadores humanos para a mesma atividade.

Uma série de abordagens podem ser utilizadas em visão computacional. Ao longo da última década as estratégias de resolução de problemas que envolvem visão computacional baseadas em aprendizagem profundo vem sendo o padrão devida à alta acurácia alcançada em uma série de contextos diferentes (HE et al., 2015; LIU et al., 2018).

## 2.3 Aprendizagem Profunda (AP) e Inteligência Artificial (AI)

### 2.3.1 Histórico

Grande parte dos avanços no campo da visão computacional ao longo da década passada se devem ao uso de ferramentas e estruturas de aprendizagem profunda (AP) (VOULODIMOS et al., 2018). Conforme detalhado adiante, na **seção 2.3.2**, AP é uma classe de algoritmos de aprendizado de máquina que utiliza uma série de camadas para entender informações contidas em dados (BENGIO, 2009).

A primeira tentativa de criar uma estrutura de AP é relatada por Pitts e McCulloch em 1943 (PITTS e McCULLOCH, 1990), quando esses autores criaram um modelo computacional baseado nas redes neurais do cérebro humano. Esses dois pesquisadores, com o auxílio de algoritmos e operações matemáticas, utilizaram uma “*threshold logic*” para simular o processo de pensamento humano.

Os avanços no estudo de algoritmos de aprendizagem profunda continuaram nos anos subsequentes. Na publicação de Samuel (1959) foi apresentado, utilizando aprendizado de máquina (AM), um computador capaz de jogar xadrez. Dessa maneira foi demonstrado que o aprendizado de máquinas era uma realidade possível. Assim foi especulado que essa tecnologia poderia ser transposta para outras aplicações.

Fukushima (1980) publicou o que é considerado a primeira RNC (rede neural convolucional). Este autor desenvolveu uma rede chamada Neocognitron. Essa arquitetura permitiu que o computador reconhecesse padrões visuais contidos em

imagens. Muitos dos conceitos pioneiros dessa publicação continuam a ser utilizados até hoje.

Embora os resultados fossem promissores, havia uma série de críticas quanto ao potencial e aplicação prática da inteligência artificial e aprendizagem profunda. Na década de 1970, a aprendizagem de máquina perdeu parte de seu *momentum*. Bar-Hillel (1973) demonstrou que para resultados de tradução realizado por máquinas era necessário o uso de muitos dados de treinamento, a uma taxa que faria qualquer aplicação prática não ser possível. Ainda na década de 1960 o relatório da ALPAC (*AUTOMATIC LANGUAGE PROCESSING ADVISORY COMMITTEE*) concluiu que não havia como prever quando uma máquina pudesse realizar uma tradução considerada “útil” (HUTCHINS, 2003).

Um exemplo do criticismo sofrido está contido na publicação *Alchemy and Artificial Intelligence* de Hubert Dreyfus, à época professor de filosofia no MIT e consultor da RAND Corporation. Dreyfus comparou os meios utilizados para alcançar os resultados usando inteligência artificial com alquimia, ou seja, ele considerou que uma máquina capaz de tomar decisões de maneira que mimetize o cérebro humano é apenas mais uma fantasia. O autor chega à conclusão que “Um padrão geral está tomando forma: Um sucesso inicial e dramático na performance em uma tarefa fácil ou um resultado de baixa qualidade em atividades complexas seguido por resultados que vão diminuindo, desencantando e, em alguns casos, leva ao pessimismo” (DREYFUS, 1965).

Em 1973 Sir *James Lighthill*, matemático na Universidade de Cambridge, publica um relatório que visa atestar o estado da arte da inteligência artificial. A conclusão desse relatório é que o impacto das pesquisas com IA não chegaram sequer perto do que foi prometido/esperado pelos pesquisadores (LIGHTHILL, 1972). Nesse cenário uma onda de pessimismo levou ao menor financiamento de pesquisas envolvendo inteligência artificial. Esse cenário pessimista levou a um período conhecido como *first AI winter* (primeiro inverno da inteligência artificial) (HENDLER, 2006).

Na década de 1980 ocorreram avanços nos estudos que envolviam inteligência artificial. Os pesquisadores começaram a apresentar produtos prontos para uso por parte do público. Por exemplo, na medicina foram criados sistemas que cruzavam informações dos sintomas que o paciente tinha com as informações contidas num banco de dados para assim produzir um diagnóstico da provável doença que acometia o paciente (KAUL et al., 2020).

Da segunda metade dos anos 1990 para os anos 2000 uma série de produtos que utilizam os princípios da inteligência artificial são apresentados para o uso do público fora das estações de pesquisa. São exemplos o robô Kismet projetado pela professora Cynthia Breazal do MIT que era capaz de simular expressões humanas, o robô humanoíde ASIMO apresentado pela Honda (Honda Motor Company) em 2000 e o I-Robot que era um aspirador de pó capaz de evitar obstáculos apresentado em 2002 pela empresa norte-americana Roomba. A partir da década de 2010 produtos que envolvem inteligência artificial deixam de ser exceções para se tornarem regras.

Ainda na década de 1990 ocorreram avanços no que diz respeito à aprendizagem profunda. Em 1999 a Nvidia apresenta para o público geral a primeira GPU (*graphics processing unit*) amplamente disponível. Esse novo equipamento possibilitou uma maior velocidade de processamento de dados, o que subsidiou avanços na aprendizagem



profunda. Ao longo da década de 2000 o tempo de processamento foi diminuindo possibilitando a análise de um maior conjunto de dados.

Com a melhora de performance dos hardwares, a aprendizagem profunda vivenciou incríveis avanços na década de 2000. Porém, havia uma escassez de dados rotulados que poderiam ser utilizados para algoritmos de aprendizado de máquina. Considerando isto, foi publicado em 2009 pela professora Fei-Fei Li, da Universidade de Stanford, o *ImageNet Dataset*. Esse banco de dados contém cerca de 14 milhões de imagens já classificadas, apresentando assim um enorme conjunto de dados de livre acesso para treinamento de algoritmos de aprendizado de máquina.

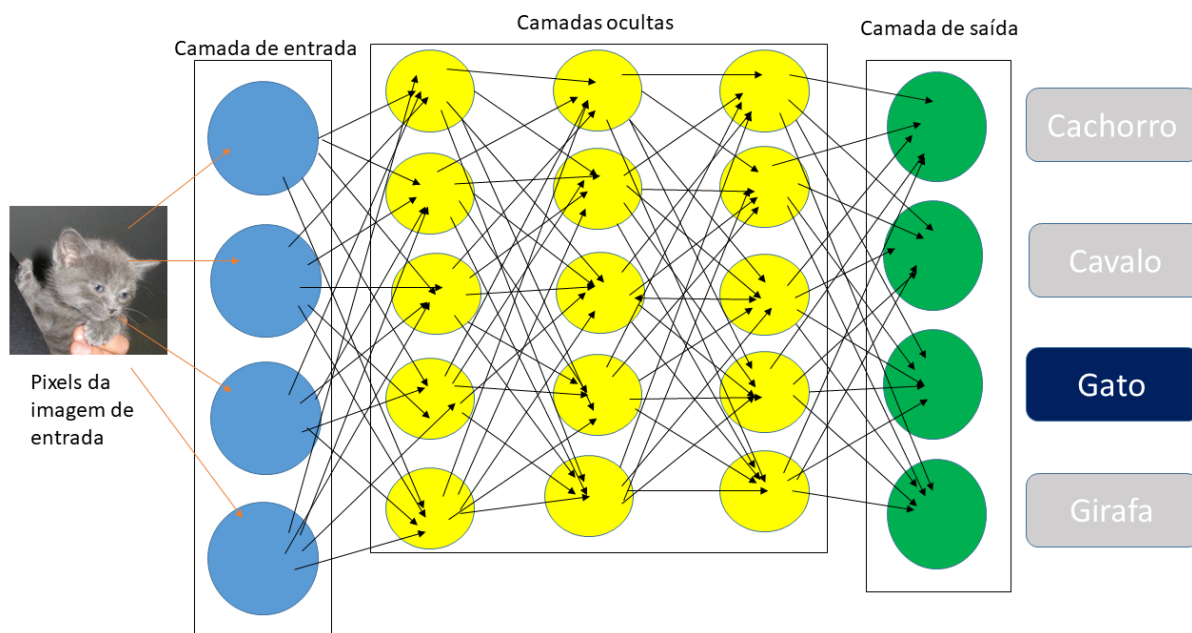
Com a maior disponibilidade de dados e a crescente melhoria no desempenho dos hardwares, o uso de aprendizagem profunda se tornou o padrão utilizado nos estudos que envolvem inteligência artificial a partir da década de 2010. Os algoritmos que utilizavam arquiteturas baseadas em aprendizagem profunda passaram a demonstrar-se muito eficientes na resolução de problemas. São exemplos dessas arquiteturas a VGG19 e VGG16 (SIMONYAN e ZISSERMAN, 2015), ResNet (HE et al., 2015), ZFnet (ZEILER e FERGUS, 2013) e UNet (RONNENBERGER et al., 2015).

### 2.3.2 Aprendizagem Profunda (AP)

Aprendizagem profunda é uma classe de algoritmos de aprendizado de máquina. Esses algoritmos tentam entender múltiplos níveis de representações e abstrações contidas na imagem (no caso de aplicação para visão computacional) para entenderem seu contexto. Isto é alcançado através da utilização de múltiplas camadas para extração de características relevantes que definem os objetos de interesse presentes nos conjuntos de dados (DENG e YUN, 2014; HAILESILASSIE, 2016).

Em uma arquitetura de aprendizagem profunda a primeira camada é referida como entrada (*input*) e a última como saída (*output*). As camadas entre elas são chamadas camadas ocultas (*hidden layers*) (CA). Em um modelo básico de arquitetura de AP, nas camadas ocultas, ocorre o processo de reconhecimento e extração de características presentes na imagem que visam definir quais são os objetos ali contidos. A entrada de uma camada oculta, exceto na primeira camada, é a saída da camada anterior. Ou seja, a informação extraída na camada anterior será alimentada para análise na nova camada (RONNEBERGER et al., 2015). Esse processo é demonstrado na **figura 1** que contém o funcionamento básico de um algoritmo de aprendizagem profunda. Nesse exemplo um conjunto de pixels (imagem) é alimentado para a rede. A rede utiliza esses pixels como entrada que posteriormente sofrem operações responsáveis pelo reconhecimento de características que definem qual é o objeto de interesse contido na imagem nas camadas

ocultas, até que resulte na camada de saída, que entrega um “rótulo” para o usuário de qual é a classe de objeto presente na imagem (neste exemplo gato).



**Figura 1.** Exemplo de como um algoritmo de aprendizagem profunda funciona. Inicialmente os pixels são avaliados na cama de entrada. O Aprendizado desta camada (o resultado das operações de convolução e agrupamento) é compartilhado hierarquicamente entre as camadas. A camada de saída consiste do “resultado” da análise, neste exemplo de que a imagem de entrada se tratava da imagem de um gato.

No processo de aprendizado o usuário pode escolher construir seu algoritmo baseado em dois princípios: i) aprendizado de máquinas supervisionado; ii) aprendizado de máquinas não supervisionado (CHOLLET, 2018).

No aprendizado supervisionado a rede é alimentada com dados “rotulados”. Ou seja, no caso de imagens, a rede recebe as imagens e as máscaras (marcações das regiões de interesse contidas numa imagem), ou imagens e rótulos (geralmente texto que define se o objeto de interesse está contido na imagem para tarefas de classificação). A partir dessa comparação entre máscara/rótulo e imagem a rede aprenderá os padrões contidos na imagem original que definem as regiões de interesse e reproduz o aprendizado para definir as regiões de interesse em um novo conjunto de dados. Já na abordagem de aprendizado de máquina não supervisionado o algoritmo não é alimentado com nenhuma predição realizada pelo usuário. Nessa abordagem, a partir de um conjunto de dados, o algoritmo passa a reconhecer padrões comuns contidos no conjunto e passa a classificá-los como similares. Essa abordagem é muito comum quando se deseja separar dados em grupos similares (ALLOGHANI et al., 2020).

#### 2.4. Redes neurais convolucionais (RNC):

Uma rede neural convolucional é um algoritmo de aprendizagem profunda alimentado com uma entrada (*input*) ao qual são atribuídos pesos (*weights*) e vieses

(*biases*), que extraem as características de interesse na imagem e são compartilhados entre as camadas, gerando uma camada de saída (*output*). As entradas de uma RNC são imagens sucessivamente apresentadas à rede neural (essas imagens são matrizes com uma ou mais camadas a depender do sistema de cores adotado). Por exemplo, em um *tensor* (formato utilizado no tensorflow keras) de formato (32, 224, 224, 3) estão contidas 32 imagens de dimensões 224 x 224 pixels com três canais para cada imagem. Essa camada de saída é o resultado final das operações realizadas em um algoritmo de aprendizado de máquina (CHOLLET, 2018). Por exemplo, em um algoritmo de classificação de animais, a entrada passa por uma série de operações que visam aprender as características que determinam qual é o animal contido naquela imagem. Nesse caso a saída seria a probabilidade do animal contido na imagem ser um determinado animal.

Essa arquitetura recebe esse nome por que é inspirada, vagamente, no funcionamento do cérebro humano (AHMAD et al., 2018). Dessa forma os neurônios individuais presentes na rede avaliam apenas uma pequena parte do que está presente, por exemplo, em uma imagem, mimetizando o que ocorre no cérebro, a cada momento é avaliado apenas uma pequena área do nosso campo de visão. O mesmo ocorre com a rede neural convolucional que de início avalia pequenas regiões separadamente com posterior junção para análise. Posteriormente todos esses “pequenos campos” são sobrepostos cobrindo toda a área de visão/análise da imagem (CHOLLET, 2018).

Basicamente uma rede deste tipo, que analisa imagens consiste de três partes: A camada de entrada, as camadas ocultas (CO) e a camada de saída. A camada oculta consiste de camadas de convolução (CC) que possuem camadas de agrupamento acopladas (CA), estas camadas são responsáveis pela extração de características de uma imagem, e uma camada totalmente conectada (CTC) que mapeia as características obtidas anteriormente, juntando-as e produzindo um resultado final como por exemplo uma imagem binária. Uma das características que permitem a alta eficiência de uma rede neural convolucional é o fato da saída de uma camada (resultado do processo de CC ou CA) ser a entrada da nova camada. Tipicamente redes neurais convolucionais possuem múltiplas camadas ocultas. Em teoria quanto maior o número dessas camadas, maior a capacidade da rede de aprender interações complexas e detectar objetos mais diferentes entre si (CHOLLET, 2018), porém há uma maior necessidade de processamento.

O resultado das primeiras camadas de convolução é a extração de características mais simples como bordas, cor e orientação do gradiente de cor. Com o acréscimo de novas camadas características mais complexas passam a ser aprendidas pelo algoritmo (LE e KAYAL, 2020).

Após essa camada de convolução, a entrada passa por uma camada de *pooling* (camada de agrupamento). Essa camada reduz as dimensões da característica (matriz) obtida na convolução. Essa redução é importante ao reduzir o peso dos dados gerados proporcionando melhor performance computacional do algoritmo e reduz o risco de *overfitting* (fenômeno que ocorre quando o algoritmo se especializa para entender o contexto das imagens em que foi treinado, apresentando pouca precisão na análise de novos dados) (CHOLLET, 2018).

Quanto maior o número de camadas de convolução e camadas de agrupamento aplicadas na rede neural mais características serão extraídas e aprendidas pelo algoritmo. Vale ressaltar que a rede terá maior precisão no seu resultado, mas isso se dará ao custo do

uso de mais recurso computacional de processamento, logo o usuário deve se atentar para o custo benefício do aumento do número de camadas de “filtro” a serem utilizadas. Além disso, com esse acréscimo de camadas de convolução e agrupamento, o modelo obtido pode se tornar muito específico para o conjunto de dados em que foi treinado, ou seja, pode se tornar pouco eficiente para realizar predições em um novo conjunto de dados.

## 2.5. UNet

Uma das arquiteturas de rede neural convolucional utilizada atualmente é a UNet. Essa arquitetura foi apresentada pela primeira vez por Olaf Ronneberger, Philipp Fischer e Thomas Brox da Universidade de Freiburg (Alemanha) em 2015. Utilizando essa arquitetura, esses autores segmentaram estruturas de células nervosas em imagens utilizando um número pequeno de imagens de treinamento para obtenção do modelo. Os autores relatam que os resultados obtidos foram melhores que o melhor método utilizado nessa tarefa de segmentação até então (RONNEBERGER et al., 2015).

O resultado bastante promissor de Ronneberger et al. (2015) tem propellido a comunidade científica a utilizar esta arquitetura já que, i) os resultados foram tão bons quanto as estratégias e arquiteturas utilizadas anteriormente e ii) com o uso de aumentação de dados (técnica que artificialmente aumenta a variação das imagens por meio de transformações como rotação, ajuste de brilho e corte) é possível treinar redes neurais com o uso de relativamente poucos dados (CHOLLET, 2018). De maneira básica, essa arquitetura consiste de um caminho de contração, que captura o contexto da imagem (definição dos parâmetros que definem uma região de interesse em uma imagem), e de um caminho de expansão simétrica que proporciona a detecção de onde está esse objeto (a região de interesse).

Ao aprender as características contidas numa imagem que definem o objeto de interesse, a rede neural constrói um mapa de características. Uma das diferenças da UNet em relação a outras arquiteturas de AP é o uso desse mapa para reconstruir uma imagem contendo os objetos de interesse na localização exata na imagem. Essa arquitetura é constituída de três partes: caminho de contração (CC), gargalo e caminho de expansão (CE). Uma outra nomenclatura utilizada é o termo *encoder* para o caminho de contração e *decoder* para o caminho de expansão.

O caminho de contração é constituído de blocos de convoluções seguidas por operações de agrupamento (geralmente é utilizado *max pooling*) conforme um *kernel* (matriz de tamanho  $I \times J$  que será multiplicado nas regiões de tamanho  $I \times J$  da imagem, realizando o processo de extração de características e aprendizado). Após cada um desses blocos o número de mapas de características dobra. Assim a rede neural pode aprender características mais complexas contidas na imagem. Os resultados dessas convoluções (os mapas de característica) são concatenados no caminho de expansão. No caminho de expansão a imagem aumenta seus valores nas duas primeiras dimensões e sofre operações de convolução de forma que o que foi aprendido no caminho de contração possa ser representado em pixels. Dessa maneira a rede aprende onde estão os objetos de interesse de forma que haja segmentação dessas regiões. Para isso, o número de blocos do caminho de expansão deve ser igual ao número de blocos do caminho de contração. Dessa forma, as dimensões da imagem utilizada como entrada são reconstruídas e a imagem resultante da rede terá as mesmas dimensões do começo do treinamento.

## 2.6. Aplicações de aprendizado de máquina

Uma das mais notáveis e difundidas aplicações de algoritmos de aprendizagem de máquina se dá na detecção de objetos de interesse em imagens. Um dos exemplos mais bem-sucedidos é na direção de carros autônomos. Através de câmeras imagens são capturadas e simultaneamente o sistema reconhece o entorno, proporcionando que o carro siga o seu trajeto sem a ocorrência de acidentes (GRIGORESCU et al., 2019). Também no campo da segurança pública já é comum a utilização de algoritmos de reconhecimento facial baseados em arquiteturas de aprendizado de máquina (ABDELMINAAM et al., 2020). Outro exemplo bastante utilizado no nosso dia-a-dia é o reconhecimento de fala como a pesquisa por voz em mecanismos de busca como o Google, e assistentes virtuais como Siri da Apple, Cortana da Microsoft e Alexa da Amazon.

Os mesmos princípios de aprendizado de máquina são utilizados nas recomendações que os sites e aplicativos nos fazem. Quando utilizamos um mecanismo de busca para compras as informações são armazenadas e posteriormente servem como base para futuras recomendações. Essa prática é comum nas sugestões de compras e de produtos de entretenimento como séries, músicas e vídeos.

Uma das áreas do conhecimento científico em que os esforços que envolvem aprendizado de máquina mais são utilizados é na medicina. A aplicação pode se dar num momento mais simples, como os robôs utilizados em chats de atendimento que servem para o recolhimento de informações que vão ser utilizadas para o diagnóstico, ou mesmo aplicações mais complexas como a segmentação de estruturas que causam uma determinada patologia ou classificação de qual doença o paciente possui.

Um exemplo é a detecção de câncer de pele que é possível utilizando AP atingindo acurácia de até 84% (AMERI, 2020). Gulshan et al. (2016) apresentaram um promissor algoritmo de classificação de retinopatia diabética utilizando uma rede neural. Chai et al. (2018) demonstraram uma *multi-branch neural network* utilizada para auxiliar no diagnóstico de glaucoma. Esses autores atingiram cerca de 91% de acurácia na detecção dessa doença, resultado este superior às metodologias utilizadas anteriormente. Utilizando uma arquitetura de aprendizagem profunda, Ma et al. (2020) apresentaram um algoritmo de segmentação de rins que deve auxiliar na detecção de pedras presentes nesse órgão. Utilizando alguns marcadores e princípios de aprendizado de máquinas também é possível diagnosticar condições mentais como esquizofrenia (SHIM et al., 2016). O uso de tecnologias que envolvam aprendizado de máquina é sobretudo animador para condições em que o diagnóstico realizado pela observação do médico é bastante difícil como abscesso peritonsilar (WILSON et al., 2019). Utilizando aprendizado de máquina, Fernandes et al. (2021) apresentaram um algoritmo capaz de prever a severidade do quadro de covid-19 de pacientes baseado em dados obtidos de exames de rotina. Tal ferramenta é mais um exemplo de como estratégias que envolvam aprendizado de máquina podem auxiliar na tomada de decisão por parte do corpo médico e na melhor alocação de recursos na administração da saúde pública.

Mesmo sendo tão promissor, o uso de ferramentas de aprendizado de máquina no auxílio do diagnóstico de doenças ainda encontra barreiras. Por vezes, quando apresentado a novas condições de captação de imagem, os algoritmos que identificam doenças através da análise de imagens não desempenham tão bem quanto quando

utilizado os dados utilizados para a obtenção do modelo. Outra barreira é a difusão ainda lenta dessas tecnologias e a distância entre cientistas de dados e médicos de forma que o diagnóstico seja mais preciso (LYNCH e LISTON, 2018).

### 2.6.1 Aplicações na genômica e proteômica

Outro campo do conhecimento em que a aplicação de algoritmos de aprendizado de máquina pode revolucionar o estado atual da pesquisa é na biologia. Por exemplo, algoritmos de aprendizado de máquina não supervisionados podem ser utilizados para realizar o *clustering* (processo de agrupamento de dados semelhantes) para subdividir um conjunto gigante de dados de maneira mais rápida e eficiente, proporcionando que esses dados sejam analisados posteriormente de maneira mais ágil (KARIM et al., 2021).

Um exemplo prático disso se dá na genômica. Por conta dos avanços na extração e sequenciamento de DNA mais dados estão abertos à comunidade. Nesse contexto, a área da genômica pode sofrer avanços extraordinários utilizando ferramentas como o aprendizado de máquina. Dentro dessa área já se utiliza aprendizado de máquina para o alinhamento de sequências de DNA, classificação, *clustering* e mineração de dados (YANG et al., 2020). Por mineração de dados é entendido a extração e reconhecimento de padrões contidos em um conjunto de dados (MAHMUD et al., 2021). Dessa forma o pesquisador pode empreender menos esforço na detecção de padrões ao mesmo tempo em que a precisão dessa separação de dados pode aumentar.

Uma empolgante utilização de algoritmos de aprendizagem profunda é acoplar esses sistemas a análise de dados obtidos de instrumentos como o DNA microarray. Desta forma, LAHMER et al. (2020) puderam classificar genes em genes regulados pelo ciclo celular e genes não regulados pelo ciclo celular com acurácia de 93%, resultado que os autores alegam superar em 20% a abordagem de análise até então considerada a mais eficiente. Outra aplicação é a identificação de sítios de início de transcrição de DNA que também pode ocorrer utilizando AM (GEORGAKILAS et al., 2020).

Na proteômica, estudo das proteínas em sistemas biológicos, a utilização de algoritmos de aprendizagem profunda também tem um potencial revolucionário. A quantidade de dados produzida tem aumentado ao longo das últimas décadas graças à difusão de análises como a espectrometria de massa. A filtragem desses dados pode se dar pelo princípio da mineração de dados citado anteriormente. Atualmente, utilizando aprendizagem profunda, é possível prever características físico-químicas de peptídeos baseados na sequência, espectro de massa e tempo de retenção (MEYER, 2021). Outros exemplos são a predição do tempo de retenção que a proteína terá nas análises de espectrometria e o sequenciamento de peptídeos (WEN et al., 2020).

### 2.6.2 Aplicações na biologia e agricultura

É possível acoplar algoritmos de visão computacional a análise de amostras biológicas. Uma das aplicações que mais podem ajudar nas rotinas laboratoriais é a automação da análise de imagens para que assim seja possível identificar estruturas de interesse que facilmente são detectadas pelo olho humano, porém são dispendiosas em termos de tempo ou que necessitam de mão-de-obra altamente especializada. Muitas

vezes essas análises podem ser feitas mais rapidamente e com maior precisão através do uso de algoritmos. Essa necessidade se torna ainda mais urgente com o crescimento no volume de dados que resulta do uso de microscópios cada vez mais eficientes. Além disso, a análise de imagens muitas vezes está sujeita a *biases* (tendências não voluntárias) do observador o que causa variação na análise quando ela é realizada por observadores diferentes sem que, necessariamente, se trate de coisas diferentes.

Outro problema na análise de amostras biológicas se dá na perda de qualidade que elas podem sofrer conforme o tempo passa (BERNAS et al., 2005; YUAN et al., 2020). As próprias estruturas biológicas podem se tornar mais difíceis de distinguir, bem como artefatos podem avançar sobre a estrutura em questão tornando a análise mais difícil o que pode produzir resultados e avaliações menos precisas. Uma estratégia que pode ser utilizada para resolver esse problema é a captação imediata da imagem da amostra (após esta ser obtida) para posterior análise, permitindo assim o registro/preservação definitiva dos dados de interesse. Outra estratégia pode ser a análise em tempo real das imagens capturadas no processo de escaneamento da estrutura biológica (COLLINS et al., 2020; WANG et al., 2020).

A segmentação de estruturas biológicas utilizando algoritmos de aprendizagem profunda já é uma realidade (KAN, 2017). Ferramentas como MitoSegNet são versáteis na detecção de mitocôndrias em uma ampla gama de dados e também na medição de algumas características como área e perímetro dessas estruturas, superando o uso de ferramentas tradicionais baseadas em aprendizado de máquina como o Ilastik (FISCHER et al., 2020).

Algoritmos de aprendizado de máquina também podem ser utilizados no estudo de plantas. É possível diferenciar espécies vegetais utilizando algoritmos de visão computacional, tarefa importante sobretudo para imagens de plantas capturadas em campo aberto, e até mesmo realizar classificações taxonômicas através da análise de imagens de folhas (MOCHIDA et al., 2018; WILF et al., 2013).

Para fins de preservação do conhecimento uma série de herbários no mundo estão sendo digitalizados. Muitas vezes as imagens das plantas são digitalizadas sem as respectivas informações taxonômicas. Visando mitigar esse problema, Carranza-Rojas et al. (2017) apresentaram uma estratégia utilizando aprendizado profundo que foi capaz de classificar, através da análise de imagens de folhas, as espécies que elas pertenciam.

Através das técnicas de visão computacional, uma série de análises fenotípicas são realizadas. Por exemplo, existem artigos publicados que alcançaram a segmentação de órgãos como flores utilizando técnicas simples como *thresholding* (NAIJAR e ZAGROUBA, 2012), extração e contagem de folhas (KUMAR e DOMNIC, 2019) ou mesmo a segmentação e contagem de frutos (SADEGHI-TEHRAN et al., 2019). A segmentação desses órgãos pode ser utilizada posteriormente para análises numéricas como largura, altura e contagem. Através dessas análises, parâmetros que se relacionam, por exemplo, a produção podem ser avaliados, auxiliando na tomada de decisão do produtor agrícola (LI et al., 2020).

Ferentinos (2018) apresenta um excelente resultado no diagnóstico de doenças de plantas. Utilizando um banco de dados de mais de 80000 imagens de 58 classes (combinação de espécie de planta e doença), esse autor apresentou um algoritmo com 99% de precisão na classificação dessas classes. Porém uma das limitações desse trabalho

é o tempo necessário para que o modelo que resultou nesta precisão fosse obtido. Foram necessários cerca de cinco dias de treinamento da VGG (arquitetura de rede neural utilizada) para a obtenção desse resultado. Além disso, não houve tentativa de testar esse algoritmo em dados diferentes do conjunto de dados original, o que pode levar a uma queda considerável da precisão alcançada já que as imagens podem ser captadas em condições diferentes daquelas que serviram para treinamento do modelo.

No estudo de bactérias a aplicação de ferramentas de visão computacional podem ser revolucionária. Atualmente as análises realizadas no estudo de bactérias em placas de petri, geralmente, são apenas qualitativas ou semi-quantitativas. Além disso, a diferença entre bactérias de diferentes espécies em imagens é muitas vezes sutil demais para que o olho humano possa realizar essa distinção. O uso de ferramentas como o ImageJ pode auxiliar na diferenciação de bactérias diferentes, acrescentar a rotina análises como área, comprimento e intensidade luminosa, segmentação de células bacterianas com posterior contagem e diferenciação entre bactérias diferentes (DANIELSEN e NORDENFELT, 2017).

Uma série de análises de contaminação ou colonização por bactérias são realizadas por meio de contagem de colônias em placas de petri ou através de contadores de colônia automatizados. A primeira opção pode ser bastante cansativa, o que pode levar a erros por parte do observador. Já na segunda opção, esses produtos são considerados de custo elevado. Uma opção que pode ser utilizada para superar os problemas citados é a utilização de softwares de visão computacional que possuam baixo custo ou que sejam de livre acesso e que contem essas estruturas de maneira tão confiável quanto a tecnologia considerada o estado da arte (GROSSI et al., 2019).

Utilizando uma rede neural, Wang et al. (2020) desenvolveram um microscópio capaz de captar imagens de bactérias e classificá-las em tempo real nas três espécies de interesse do estudo. Além disso, para cada uma das análises realizadas foi estimado o valor de \$ 0,60 por teste. Esse fato traz à luz outra interessante potencialidade quando se utiliza ferramentas de visão computacional, sobretudo quando baseadas em aprendizado profundo na análise de amostras biológicas, que é a redução de custos enquanto os resultados são tão bons quanto ou até melhores que as análises tradicionais consideradas de maior eficácia.

Essa redução de custos também é relatada no estudo de fungos causadores de doenças. Zielinski et al. (2020) apresentam um algoritmo capaz de classificar diferentes espécies de fungos causadores de doenças utilizando um modelo de rede neural profunda chamado *deep bag of words algorithm*. Ao utilizar esse método de classificação, que foi bastante preciso, o uso de análises bioquímicas para a detecção de qual fungo estava presente no meio se tornou supérfluo, o que diminui os custos dessa análise ao mesmo momento que produz um diagnóstico mais rápido.

A detecção de fungos micorrízicos arbusculares (FMA) também já é relatada utilizando aprendizado de máquina. Evangelisti et al. (2021) apresentaram um software intitulado AMFinder, utilizando este algoritmo é possível detectar, quantificar a colonização por FMAs e encontrar hifas intraradiciais através do uso de uma RNC que analisa imagens.

O estudo de nematóides também pode se dar através de ferramentas de VC (CHEN et al., 2020). Um promissor resultado no estudo dos nematoides é apresentado por Hakim



et al. (2018). Esses autores apresentam uma rede neural de aprendizagem profunda que foi capaz de diferenciar o nematoide de vida livre *Caenorhabditis elegans* de artefatos presentes em placas de petri preenchidas com agarose. Adicionalmente foi apresentada a capacidade de diferenciação fenotípica (masculino e feminino) nesse algoritmo.

Para o nematoide parasita da soja, *Heterodera glycines*, utilizando AP, foi desenvolvido um algoritmo capaz de contar os ovos depositados por esse nematoide em amostras de solo de forma que a tomada de decisão para controle dessa peste seja mais eficiente (AKINTAYO et al., 2018). Thevenoux et al. (2021) apresentam uma estratégia de estudo de nematoides utilizando uma rede neural convolucional e *custom computer vision algorithm* (CCVA). Esses autores conseguiram distinguir duas espécies próximas de nematoide e apresentaram medidas de pixels para esses organismos (para a medida do comprimento e largura do “botão” desse organismo). A taxa de erro nas avaliações para um dos dois organismos avaliados foi maior que para o outro organismo, revelando possível problema de *overfitting*.

Há de se ressaltar que o uso de algoritmos de aprendizagem profunda envolvendo visão computacional dentro da microbiologia ainda encontra barreiras. O uso desses algoritmos pode revolucionar a realidade do estudo dos microorganismos como já tem revolucionado para uma série de outras áreas do conhecimento. Nesse contexto é necessário um maior engajamento da comunidade no que diz respeito à aprendizagem profunda. Outro problema é a, ainda, pequena disponibilidade de bancos de dados de imagens de microorganismos de acesso livre, o que poderia levar a avanços nesse campo de estudo. Outra possível limitação, é o engajamento ainda incipiente da comunidade com linguagens de programação (ZHANG et al., 2021).

### 3. Referências

ABDELMINAAM, D. S.; ALMANSORI, A. M.; TAHA, M.; BADR, E. A deep facial recognition system using computational intelligent algorithms. **PLoS ONE**, v. 15, n. 12 December, p. 1–27, 2020. Disponível em: <<http://dx.doi.org/10.1371/journal.pone.0242269>>.

AHMAD, M.; JOE, J.; HAN, D. CortexNet: Convolutional Neural Network with Visual Cortex in human brain. **2018 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2018**, p. 206–212, 2018. IEEE.

AKINTAYO, A.; TYLKA, G. L.; SINGH, A. K.; et al. A deep learning framework to discern and count microscopic nematode eggs. **Scientific Reports**, v. 8, n. 1, p. 1–11, 2018.

ALLEN, C.; MEHLER, D. M. A. Erratum: Open science challenges, benefits and tips in early career and beyond (PLoS Biology (2019) 17:5 (e3000246) DOI: 10.1371/journal.pbio.3000246). **PLoS Biology**, v. 17, n. 12, p. 1–14, 2019.

ALLOGHANI, M.; AL-JUMEILY, D.; MUSTAFINA, J.; HUSSAIN, A.; ALJAAF, M.A. A systematic review on supervised and unsupervised machine learning algorithms for data science. In: BERRY, M.W.; MOHAMED, A.; YAP, B.W. Supervised and Unsupervised Learning for Data Science. **Springer**, 2021.

AMERI, A. A deep learning approach to skin cancer detection in dermoscopy images. **Journal of Biomedical Physics and Engineering**, v. 10, n. 6, p. 801–806, 2020.  
BERNAS, T.; ROBINSON, J. P.; ASEM, E. K.; RAJWA, B. Loss of image quality in photobleaching during microscopic imaging of fluorescent probes bound to chromatin. **Journal of Biomedical Optics**, v. 10, n. 6, p. 064015, 2005.

ARTHUR, S. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210, 1959.

BAR-HILLEL, M. On the subjective probability of compound events. **Organizational Behavior and Human Performance**, v. 9, n. 3, p. 396–406, 1973.

BENGIO, Y. **Learning deep architectures for AI**. 2009.

CERNEK, P.; BOLLIG, N.; ANKLAM, K.; DÖPFER, D. Hot topic: Detecting digital dermatitis with computer vision. **Journal of Dairy Science**, v. 103, n. 10, p. 9110–9115, 2020.

CHAI, Y.; LIU, H.; XU, J. Glaucoma diagnosis based on both hidden features and domain knowledge through deep learning models. **Knowledge-Based Systems**, v. 161, p. 147–156, 2018. Elsevier B.V. Disponível em: <<https://doi.org/10.1016/j.knosys.2018.07.043>>.

CHEN, L.; STRAUCH, M.; DAUB, M.; et al. A CNN FRAMEWORK BASED ON LINE ANNOTATIONS Hans-Georg Luigs , Susanne Schultz-Kuhlmann , Stefan Kr Imaging and Computer Vision , RWTH Aachen University , Aachen , Germany Julius

Köln Institute : Federal Research Centre for Cultivated Plants , Elsdorf . , , n. 031, p. 508–512, 2020.

CHOLLET, F.; Deep Learning With Python. **Manning Publications co.**, 2018.

COLLINS, J. T.; KNAPPER, J.; STIRLING, J.; et al. Robotic microscopy for everyone: the OpenFlexure microscope. **Biomedical Optics Express**, v. 11, n. 5, p. 2447, 2020.

DANIELSEN, J.; NORDENFELT, P. Computer vision-based image analysis of bacteria. **Methods in Molecular Biology**, v. 1535, p. 161–172, 2017.

DREYFUS, H. L. *Alchemy and Artificial Intelligence.*, 1965.

DUTTA, A.; ZISSERMAN, A. The VIA annotation software for images, audio and video. **MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia**, p. 2276–2279, 2019.

FERENTINOS, K. P. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, v. 145, n. January, p. 311–318, 2018. Elsevier. Disponível em: <<https://doi.org/10.1016/j.compag.2018.01.009>>.

FERNANDES, F. T.; DE OLIVEIRA, T. A.; TEIXEIRA, C. E.; et al. A multipurpose machine learning approach to predict COVID-19 negative prognosis in São Paulo, Brazil. **Scientific Reports**, v. 11, n. 1, p. 1–7, 2021. Nature Publishing Group UK. Disponível em: <<https://doi.org/10.1038/s41598-021-82885-y>>.

FISCHER, C. A.; BESORA-CASALS, L.; ROLLAND, S. G.; et al. MitoSegNet: Easy-to-use Deep Learning Segmentation for Analyzing Mitochondrial Morphology. **iScience**, v. 23, n. 10, 2020.

FONSECA, A.J.; FREITAS, A.F.; CARVALHO, G.R.; et al. Arbuscular mycorrhizal fungus on the initial growth and nutrition of *Coffea arabica* L. genotypes. **Ciência e Agrotecnologia**, v. 43, 2019.

FORTUNATO, L.; GALASSI, M. The case for free and open source software in research and scholarship. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 379, n. 2197, 2021.

FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics**, v. 36, n. 4, p. 193–202, 1980.

GEORGAKILAS, G. K.; PERDIKOPANIS, N.; HATZIGEORGIOU, A. Solving the transcription start site identification problem with ADAPT-CAGE: a Machine Learning algorithm for the analysis of CAGE data. **Scientific Reports**, v. 10, n. 1, p. 1–12, 2020.

GRIGORESCU, S.; TRASNEA, B.; COCIAS, T.; MACESANU, G. A survey of deep learning techniques for autonomous driving. **Journal of Field Robotics**, v. 37, n. 3, p. 362–386, 2020.

- GROSSI, M.; PAROLIN, C.; VITALI, B.; RICCÒ, B. Computer vision approach for the determination of microbial concentration and growth kinetics using a low cost sensor system. **Sensors (Switzerland)**, v. 19, n. 24, 2019.
- GULSHAN, V.; PENG, L.; CORAM, M.; et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. **JAMA - Journal of the American Medical Association**, v. 316, n. 22, p. 2402–2410, 2016.
- HAILESILASSIE, T. Rule Extraction Algorithm for Deep Neural Networks: A Review. , v. 14, n. 7, p. 376–381, 2016. Disponível em: <<http://arxiv.org/abs/1610.05267>>..
- HAKIM, A.; MOR, Y.; TOKER, I. A.; LEVINE, A.; NEUHOF, M.; MARKOVITZ, Y.; RECHAVI, O. WorMachine: Machine learning-based phenotypic analysis tool for worms. **BMC Biology**, v. 16, n. 1, p. 1–11, 2018. BMC Biology.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, v. 2016-Decem, p. 770–778, 2016.
- HENDLER, E. J. A Letter from the Editor We ' d Like to Hear from You Letters to the Editor : **IEEE Intelligent Systems**, p. 2006–2006, 2006.
- HUTCHINS, J. ALPAC: the (in)famous report. , v. 14, n. December 1964, p. 131–135, 2003.
- KARIM, M. R.; BEYAN, O.; ZAPPA, A.; et al. Deep learning-based clustering approaches for bioinformatics. **Briefings in Bioinformatics**, v. 22, n. 1, p. 393–415, 2021.
- KAUL, V.; ENSLIN, S.; GROSS, S. A. History of artificial intelligence in medicine. **Gastrointestinal Endoscopy**, v. 92, n. 4, p. 807–812, 2020. American Society for Gastrointestinal Endoscopy. Disponível em: <<https://doi.org/10.1016/j.gie.2020.06.040>>.
- LAHMER, H.; OUESLATI, A. E.; LACHIRI, Z. Classification of DNA Microarrays Using Deep Learning to identify Cell Cycle Regulated Genes. **2020 International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2020**, p. 0–4, 2020.
- LE, M.; KAYAL, S. Revisiting Edge Detection in Convolutional Neural Networks. , 2020. Disponível em: <<http://arxiv.org/abs/2012.13576>>.
- LI, J.; MENG, B.; CHAI, H.; et al. Arbuscular Mycorrhizal Fungi Alleviate Drought Stress in C3 (*Leymus chinensis*) and C4 (*Hemarthria altissima*) Grasses via Altering Antioxidant Enzyme Activities and Photosynthesis. **Frontiers in Plant Science**, v.10, article 499, 2019.
- LI, Z.; GUO, R.; LI, M.; CHEN, Y.; LI, G. A review of computer vision technologies for plant phenotyping. **Computers and Electronics in Agriculture**, v. 176, n. July, p.

105672, 2020. Elsevier. Disponível em:  
<<https://doi.org/10.1016/j.compag.2020.105672>>.

LIGHTHILL, J. Part I Artificial Intelligence A general survey by Sir James Lighthill FRS Lucasian Professor of Applied Mathematics , 1 Introduction 2 The ABC of the subject. , , n. July, p. 1–22, 1972.

LIU, C.; ZOPH, B.; NEUMANN, M.; SHLENS, J.; HUA, W.; LI, L.; FEI-FEI, L.; YUILLE, A.; HUANG, J.; MURPHY, K. Progressive Neural Architecture Search. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 11205 LNCS, p. 19–35, 2018.

LYNCH, C. J.; LISTON, C. New machine-learning technologies for computer-aided diagnosis. , v. 24, n. September, p. 1304–1305, 2018.

MA, F.; SUN, T.; LIU, L.; JING, H. Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. **Future Generation Computer Systems**, v. 111, p. 17–26, 2020. Elsevier B.V. Disponível em:  
<<https://doi.org/10.1016/j.future.2020.04.036>>.

MAHMUD, M.; KAISER, M. S.; MCGINNITY, T. M.; HUSSAIN, A. **Deep Learning in Mining Biological Data**. Springer US, 2021.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity (reprinted from bulletin of mathematical biophysics, vol 5, pg 115-133, 1943). **Bulletin of Mathematical Biology**, v. 52, n. 1--2, p. 99–115, 1990. Disponível em:  
<[http://journals2.scholarsportal.info/pdf/00928240/v52i1-2/99\\_alcotiina.xml](http://journals2.scholarsportal.info/pdf/00928240/v52i1-2/99_alcotiina.xml)>.

MESA-VALLE, C.M.; GARRIDO-CARDENAS, J.A.; CEBRIAN-CARMONA, J.; et al. Global Research on Plant Nematodes. **agronomy**, v. 10, article 1148, 2020.

MEYER, J. G. Deep learning neural network tools for proteomics. **Cell Reports Methods**, v. 1, n. 2, p. 100003, 2021. Elsevier Company. Disponível em:  
<<https://doi.org/10.1016/j.crmeth.2021.100003>>.

MOCHIDA, K.; KODA, S.; INOUE, K.; et al. Computer vision-based phenotyping for improvement of plant productivity: A machine learning perspective. **GigaScience**, v. 8, n. 1, p. 1–12, 2018. Oxford University Press.

MUNAFÒ, M. R.; NOSEK, B. A.; BISHOP, D. V. M.; et al. A manifesto for reproducible science. **Nature Human Behaviour**, v. 1, n. 1, p. 1–9, 2017. Macmillan Publishers Limited. Disponível em: <<http://dx.doi.org/10.1038/s41562-016-0021>>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: **Internacional Conference on Medical Image Computing and Computer-Assisted Intervention**, p. 234-241, Springer, 2015.

SADEGHI-TEHRAN, P.; VIRLET, N.; AMPE, E.M.; REYNS, P.; HAWKESFORD, M.J. *DeepCount*: In-Field Automatic Quantification of Wheat Spikes Using Simple

- Linear Iterative Clustering and Deep Convolutional Neural Network. **Frontiers in Plant Sciences**, v. 10, 2019.
- SCHNEIDER, C. A.; RASBAND, W. S.; ELICEIRI, K. W. NIH Image to ImageJ: 25 years of image analysis. **Nature Methods**, v. 9, n. 7, p. 671–675, 2012. Nature Publishing Group. Disponível em: <<http://dx.doi.org/10.1038/nmeth.2089>>.
- SHIM, M.; HWANG, H. J.; KIM, D. W.; LEE, S. H.; IM, C. H. Machine-learning-based diagnosis of schizophrenia using combined sensor-level and source-level EEG features. **Schizophrenia Research**, v. 176, n. 2–3, p. 314–319, 2016. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.schres.2016.05.007>>.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings**, p. 1–14, 2015.
- THEVENOUX, R.; LE, V. L.; VILLESÈCHE, H.; et al. Image based species identification of Globodera quarantine nematodes using computer vision and deep learning. **Computers and Electronics in Agriculture**, v. 186, n. May, 2021.
- VÁZQUEZ, D.; BERNAL, J.; JAVIER SÁNCHEZ, F.; et al. A Benchmark for Endoluminal Scene Segmentation of Colonoscopy Images. **Journal of Healthcare Engineering**, v. 2017, 2017.
- VOULODIMOS, A.; DOULAMIS, N.; DOULAMIS, A.; PROTOPAPADAKIS, E. Deep Learning for Computer Vision: A Brief Review. **Computational Intelligence and Neuroscience**, v. 2018, 2018.
- WANG, H.; CEYLAN KOYDEMIR, H.; QIU, Y.; et al. Early detection and classification of live bacteria using time-lapse coherent imaging and deep learning. **Light: Science and Applications**, v. 9, n. 1, 2020. Springer US. Disponível em: <<http://dx.doi.org/10.1038/s41377-020-00358-9>>.
- WEN, B.; ZENG, W. F.; LIAO, Y.; et al. Deep Learning in Proteomics. **Proteomics**, v. 20, n. 21–22, 2020.
- WILEY, V.; LUCAS, T. Computer Vision and Image Processing: A Paper Review. **International Journal of Artificial Intelligence Research**, v. 2, n. 1, p. 22, 2018.
- WILSON, M. B.; ALI, S. A.; KOVATCH, K. J.; SMITH, J. D.; HOFF, P. T. Machine Learning Diagnosis of Peritonsillar Abscess. **Otolaryngology - Head and Neck Surgery (United States)**, v. 161, n. 5, p. 796–799, 2019.
- WOELFLE, M.; OLLIARO, P.; TODD, M. H. Open science is a research accelerator. **Nature Chemistry**, v. 3, n. 10, p. 745–748, 2011. Nature Publishing Group.
- WOLLMAN, R.; STUURMAN, N. High throughput microscopy: From raw images to discoveries. **Journal of Cell Science**, v. 120, n. 21, p. 3715–3722, 2007.
- YANG, L.; GHOSH, R. P.; MATTHEW FRANKLIN, J.; et al. NuSeT: A deep learning

tool for reliably separating and analyzing crowded cells. **PLoS Computational Biology**, v. 16, n. 9, p. 1–20, 2020. Disponível em: <<http://dx.doi.org/10.1371/journal.pcbi.1008193>>.

YUAN, J.; WANG, X.; ZHOU, H.; et al. Comparison of Sample Preparation Techniques for Inspection of Leaf Epidermises Using Light Microscopy and Scanning Electronic Microscopy. **Frontiers in Plant Science**, v. 11, n. February, p. 1–13, 2020.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 8689 LNCS, n. PART 1, p. 818–833, 2014.

ZHANG, Y.; JIANG, H.; YE, T.; JUHAS, M. Deep Learning for Imaging and Detection of Microorganisms. **Trends in Microbiology**, v. 29, n. 7, p. 569–572, 2021. Elsevier Ltd. Disponível em: <<https://doi.org/10.1016/j.tim.2021.01.006>>.

ZHANG, C.; WIJNEN, B.; PEARCE, J. M. Open-Source 3-D Platform for Low-Cost Scientific Instrument Ecosystem. **Journal of Laboratory Automation**, v. 21, n. 4, p. 517–525, 2016.

ZIELISKI, B.; SROKA-OLEKSIK, A.; RYMARCZYK, D.; PIEKARCZYK, A.; BRZYCHCZY-WOCH, M. Deep learning approach to describe and classify fungi microscopic images. **PLoS ONE**, v. 15, n. 6 June, p. 1–16, 2020.

## ARTIGO I

### Estágios automatizados de baixo custo e código livre para o imageamento de organismos do solo

#### Abstract

No estudo de organismos do solo, comumente, se faz necessário o uso de equipamentos que permita a observação destes organismos já que eles podem não ser visíveis ao olho nu. Um dos equipamentos que pode ser utilizado para este fim é o microscópio. Embora bastante utilizado, este equipamento pode apresentar custo financeiro elevado. Um dos componentes do microscópio que pode contribuir significativamente para esta realidade é o estágio, que, quando automatizado, apresenta custos elevados. Porém o uso deste componente se justifica pelo aumento na capacidade de análise do equipamento, por exemplo ao ampliar o campo de visão em relação a uma amostra de forma que mais informações sejam dali extraídas. Uma possível resolução para este problema é adotar o uso de estágios de código aberto. Para popularizar o uso desta tecnologia apresentamos neste artigo dois estágios automatizados e seus softwares de controle desenvolvidos no Laboratório de Microbiologia e Processos Biológicos do Solo da Universidade Federal de Lavras. O primeiro estágio (MicroStage) foi adaptado a um microscópio de campo claro comumente presente em laboratórios de análises biológicas. O segundo estágio (NemaStage) foi adaptado a um microscópio digital de baixo custo. Para testar o desempenho destes equipamentos, uma lâmina de microscopia contendo raízes micorrizadas foi avaliada para MicroStage e uma lâmina contendo raiz parasitada por nematoides foi avaliada para NemaStage. Os estágios automatizados e os seus softwares de controle foram eficientes no escaneamento total de lâminas de microscopia ou de regiões destas, enquanto mantinham detalhamento satisfatório em relação as estruturas presentes. Quando acoplado a uma estratégia de junção das imagens obtidas ao longo do escaneamento, foi possível realizar a ampliação do campo de visão em relação a amostra. Por se tratar de equipamentos de baixo custo e código livre, acreditamos que os equipamentos aqui apresentados podem constituir uma excelente opção sobretudo para laboratórios com menor poder aquisitivo.

Highlight 1: Estágios de baixo custo mantém resultados de qualidade no imageamento de amostras biológicas.

Highlight 2: Estágios automatizados podem ser obtidos por custo baixo

**Palavras-chave:** Microscopia, ciência aberta, nematoide, equipamento de baixo custo, NemaStage

O artigo científico se encontra nas normas da revista “Computers and Electronics in Agriculture” e está sujeito a alterações.



## 1. Introdução

O microscópio é uma ferramenta fundamental no estudo de microrganismos não visíveis a olho nu. Desde os primeiros microscópios utilizados com fins científicos por Robert Hooke e Antonj Van Leeuwenhoek no século XVII até os dias atuais, uma série de evoluções no uso e na capacidade de análise desses equipamentos vem ocorrendo (WOLLMAN et al., 2007).

Um dos componentes do microscópio que tem sofrido evoluções significativas é o estágio. Esse componente é o receptáculo da lâmina que será submetida à observação no aparelho e permite que o usuário mova a amostra para visualizar diferentes áreas de interesse. A necessidade do estágio se dá pelo fato de a ampliação reduzir o campo de visão do usuário, sendo, portanto necessário escanear a amostra manualmente pela movimentação do estágio. Por ser tradicionalmente de operação manual, a visualização de amostras em microscópio está sujeita a uma série de erros ou imprecisões por parte do operador (e.g. erro de posicionamento, análise repetida em algumas áreas, áreas sem imageamento).

Uma maneira de melhorar o manuseio, e conseqüentemente o resultado da análise, é através da automação do movimento do estágio (Ma et al., 2009), que permite um posicionamento preciso e facilmente replicável do campo visual. Outra vantagem do uso de estágio automatizado é a diminuição de danos à saúde do operador, já que dores no pulso e nas costas são comumente relatadas pelos usuários de microscópios (Thompson et al., 2003), especialmente quando necessário analisar amostras extensas. Operando nesse tipo de estágio, há menos movimento repetitivo por parte do operador, bem como menor necessidade de se aproximar do microscópio para movimentar a lâmina que está sendo observada, já que é possível utilizar um controle, ou mesmo enviar comandos via software que realizam esta movimentação.

Embora melhore a qualidade do fluxo de análises no laboratório, o uso de estágios automatizados encontra algumas barreiras. A mais notável é o custo financeiro. Em pesquisa de mercado realizada em julho de 2021, os estágios automatizados encontrados custavam mais de R\$ 40.000,00 (sem incluir o microscópio), tornando sua utilização inviável para muitos laboratórios. Outra dificuldade é o alto custo dos microscópios e de sua manutenção, especialmente problemática em países em desenvolvimento onde a assistência técnica se torna um fator limitante (Malkin, 2007; Malkin e Keane, 2010).

Uma alternativa para contornar essas limitações é trabalhar no escopo da ciência aberta. Nesta abordagem o usuário tem acesso a softwares e hardwares de livres, podendo fabricar e utilizar equipamentos a baixos custos, e até mesmo adaptá-los à realidade local e do organismo que deve ser analisado. Neste escopo os custos financeiros são drasticamente reduzidos uma vez que o usuário arca apenas com os custos do material e pode utilizar a própria mão-de-obra para fabricar o equipamento (Woelfle et al., 2011).

Com o uso de estágio automatizado, é possível a ampliação do campo visual quando se analisa uma amostra. Para tanto, o usuário pode utilizar uma câmera digital para capturar imagens precisamente posicionadas na amostra e, posteriormente ao escaneamento, fazer a junção das imagens individuais para compor uma imagem em alta definição compreendendo toda a amostra ou algumas regiões de interesse. Dessa forma é possível contextualizar melhor as informações contidas na lâmina que está sendo observada e perceber relações espaciais presentes na estrutura ou organismo analisado.

Essa estratégia de escaneamento imediato e completo de uma lâmina se torna imprescindível, sobretudo, para amostras biológicas que perdem qualidade com o passar do tempo, pois permite um registro definitivo que pode ser analisado em qualquer momento (Bernas et al., 2005; Yuan et al., 2020).

Neste contexto apresentamos duas alternativas de baixo custo para escaneamento de amostras biológicas usando estágios automatizados. Um deles baseado em um microscópio digital de baixo custo e outro acoplado a um microscópio de campo claro, representativo daqueles comumente encontrado em laboratórios voltados para análises biológicas. Apresentamos também softwares de código aberto para controle do estágio e para a aquisição automática de imagens.

## 2. Metodologia

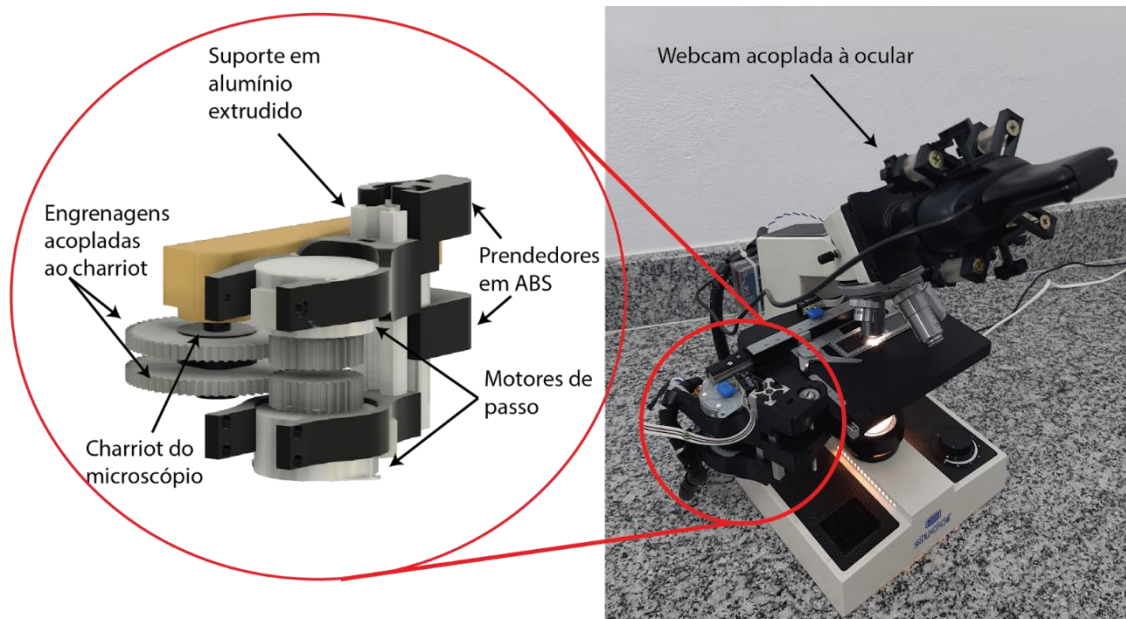
### 2.1 - Estágio automatizado acoplado a microscópio de campo claro (MicroStage)

Com o objetivo de automatizar o imageamento de lâminas com raízes colonizadas por fungos micorrízicos, nós desenvolvemos e adaptamos um aparelho em um antigo microscópio binocular, acromático, de campo claro (Studar M PZO). Esse aparelho é composto por dois motores de passo acoplados, via engrenagens impressas em 3D, ao Charriot do microscópio, controlando a movimentação do estágio, conforme ilustrado na **figura 1**. A movimentação dos motores é modulada por um microcontrolador Arduino Nano contendo o software grbl 0.9, utilizando a linguagem GCode para definir o posicionamento da amostra.

Para definir a grade de imagens a ser obtida e enviar os comandos para posicionar a amostra, nós desenvolvemos um software, StageControl, escrito na linguagem Python 3.0. Para usar este software, o usuário especifica a distância a ser percorrida ao longo dos eixos X e Y (resultando na área total que deve ser escaneada), e a distância entre imagens consecutivas. A aquisição das imagens pela câmera digital (modelo Logitech C720) acoplada ao microscópio também é controlada automaticamente pelo software StageControl.

As engrenagens que conectam os motores de passo ao Charriot foram desenhadas no software Autodesk Fusion360 e confeccionadas em uma impressora 3D (Ender 3 Pro) utilizando filamento de ABS por conta da satisfatória resistência mecânica que o material apresenta. O mesmo procedimento foi adotado para confecção do adaptador utilizado para acoplar a câmera digital à ocular do microscópio.

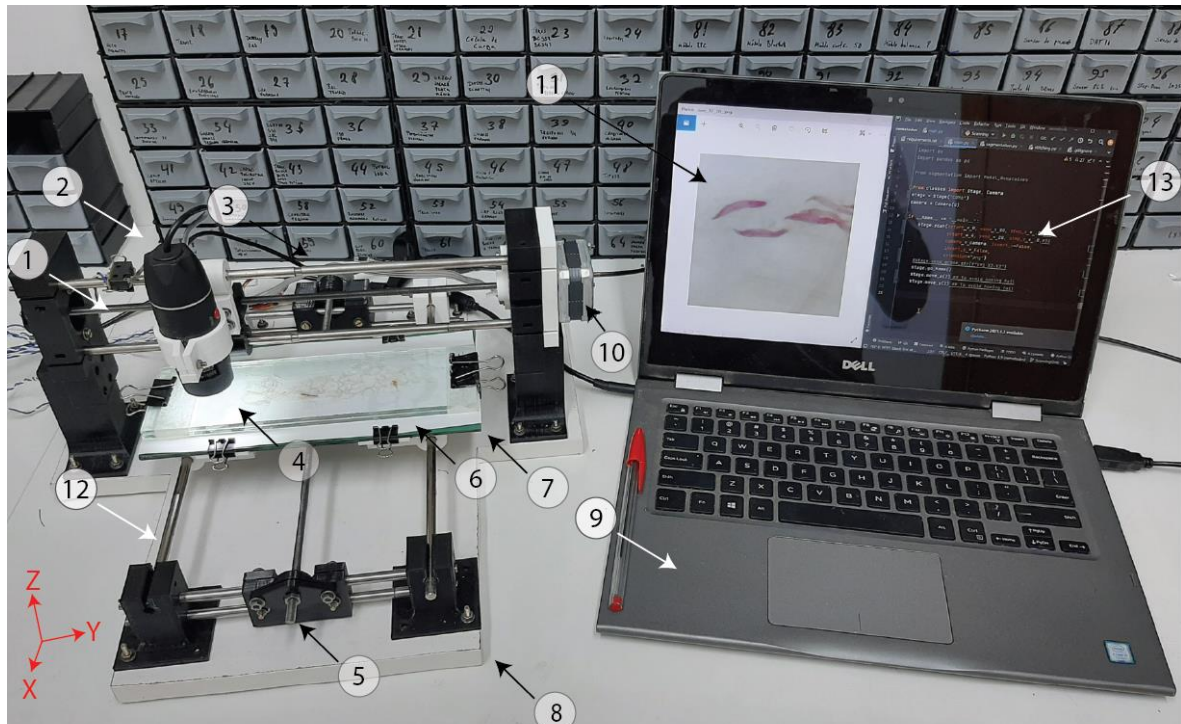
As distâncias máximas que podem ser percorridas utilizando esse estágio são 40 e 90mm ao longo dos eixos X e Y respectivamente, que é o suficiente para escanear completamente as lâminas de microscopia convencionais (com dimensões de 25 x 75 mm).



**Figura 1.** Automação de movimento de estágio de microscópio de campo claro. O aparelho acoplado ao Charriot do microscópio é composto por dois motores de passos que tracionam o Charriot através de engrenagens. Os motores são fixados por suportes impressos em 3D usando ABS e por uma peça de alumínio extrudido. O controle do movimento dos motores é realizado através de um microcontrolador, que por sua vez é controlado por um software escrito em Python rodando em um computador. A aquisição das imagens é feita por uma câmera acoplada à ocular do microscópio.

## 2.2 Microscópio digital de baixo custo acoplado a um estágio automatizado para escaneamento de amostras de maiores dimensões (NemaStage)

Com o fim de tornar acessível o imageamento de lâminas de maiores dimensões (como as usadas no estudo de nematóides parasitas de plantas) e criar alternativas aos microscópios mais caros frequentemente utilizados em análises microbiológicas, nós desenvolvemos um estágio automatizado acoplado a um microscópio digital, ambos de baixo custo.



**Figura 2.** O microscópio digital (2) é movido ao longo do eixo Y através de um motor de passo (10) acoplado a um fuso (1) e regulado por um microcontrolador Arduino Nano (não aparente na imagem). A lâmina de vidro com a amostra (4) é sustentada por uma base de vidro (7) que movimentada ao longo do eixo X através de um fuso (5) tracionado por um motor de passo (3), também ligado a um microcontrolador. A iluminação através de LEDs do microscópio digital é refletida por uma faixa reflexiva (6) colocada sob a amostra para aumentar o contraste da imagem. Para dar rigidez a ambos os eixos, foram utilizadas guias lineares (12) de 8mm de diâmetro, em aço inox retificado, reciclados de impressoras descartadas, assim como os motores de passo. Todo o sistema é montado sob uma base de MDF (8). Tanto a aquisição da imagem (11) pelo microscópio digital quanto o microcontrolador responsável pela movimentação dos eixos são controlados a partir de um software de código livre escrito na linguagem Python (13) rodando em um laptop (9).

NemaStage consiste, basicamente, de um microscópio digital de baixo custo acoplado a um estágio automatizado. O movimento do microscópio se dá no eixo Y. Esse movimento é proporcionado por um motor de passo acoplado a um fuso (**figura 2**). Esse movimento é controlado pelo usuário através de um microcontrolador do tipo Arduino Nano.

A lâmina posta acima de uma base de vidro se movimenta ao longo do eixo X (este movimento também é acionado por um motor de passo também ligado a um microcontrolador). A iluminação da amostra (da lâmina contendo raízes infectadas por nematoides) se dá através de LEDs do microscópio digital.

Para dar estabilidade a NemaStage, evitando perda de qualidade nas imagens captadas, foram utilizadas guias lineares em aço inox retificado de 8mm. Vale ressaltar que tanto estas guias como os motores de passo foram reaproveitados de impressoras.

Todo o processo de captação das imagens, quanto a movimentação, são controladas via laptop através do software StageControl citado na **seção 2.1**. A conexão entre o laptop e o software de controle se dá através de conexão USB.

Esse estágio tem capacidade de movimentação de 200 mm ao longo do eixo X e de 160 mm ao longo do eixo Y. Com estas dimensões é possível realizar o escaneamento

completo de placas grandes como as utilizadas no estudo de nematóides. Com estas dimensões, foi possível realizar o escaneamento da lâmina por completo em cerca de uma hora utilizando este equipamento. Esse microscópio também teria capacidade de escanear simultaneamente 16 lâminas convencionais de microscopia (com dimensões de 25 x 75mm), como as comumente utilizadas na microscopia de fungos micorrízicos arbusculares.

O custo total para construção do aparelho, incluindo o microscópio, ficou em torno de R\$ 268,00.

## **2.3 Preparo das amostras biológicas**

### **2.3.1 Amostras de fungo micorrízico arbuscular e escaneamento**

Plantas de milheto (*Pennisetum glaucum*) serviram como plantas isca para a obtenção de raízes micorrizadas. Para detalhes do processo de cultivo das plantas e espécies de fungos micorrízicos utilizados, consultar a **seção 2.1** do **artigo 3** desta dissertação.

Fragmentos de raízes de 1 cm de comprimento foram obtidos e submetidos posteriormente à microscopia. Inicialmente esses fragmentos foram corados conforme a metodologia descrita por Koske e Gemma (1989). Esses fragmentos foram, então, postos em lâminas de dimensões 2,6 cm x 7,6 cm contendo PVLG (polivinil-ácido láctico-glicerol) Regiões destas lâminas foram então escaneadas através do uso de MicroStage para verificar a capacidade do equipamento em realizar o escaneamento automatizado. Para a captação das imagens resultantes deste escaneamento foi utilizada uma câmera modelo Logitech C720. Posteriormente ao escaneamento foi verificada se a abordagem utilizada conseguiu captar imagens de raízes com estruturas fúngicas (arbúsculos, hifas e vesículas) que denotam a colonização micorrízica através da observação manual destas imagens.

### **2.3.2 Amostras de raízes infectadas por nematóides**

A metodologia completa está presente na **seção 2.1** do **artigo 2** desta dissertação. Brevemente, nematóides no estágio juvenil (J2) foram obtidos de coleções puras de *Meloidogyne javanica* multiplicadas em raízes de tomate (*Solanum lycopersicum* L.). A suspensão de ovos foi obtida conforme a metodologia descrita por Hussey e Barker (1973). Para a obtenção dos juvenis J2, esta suspensão de ovos foi incubada em câmara de eclosão à 28°C. Só foram utilizados J2 eclodidos após 24 horas de montagem da câmara.

Posteriormente estes juvenis foram inoculados em substrato que continha sementes de soja (*Glycines max*). Sete dias após esta infestação as raízes foram retiradas e lavadas em água corrente e submetidas ao clareamento em solução de NaClO (1,5%) por 10 minutos. Posteriormente ocorreu uma lavagem para retirar resíduos desta substância. Em seguida as raízes foram transferidas para tubos de vidro onde foram cobertas com uma solução corante preparada com 3,5 g de fucsina ácida + 250 mL de ácido láctico + 750 mL de água destilada (adaptado de Byrd et al., 1983) e diluída em 1:29 (1 mL de solução corante inicial + 29 mL de água destilada). Posteriormente os tubos

de vidro foram postos em chapa aquecedora e mantidos até ebulição da solução corante. Este resfriamento se deu em condição ambiente por uma noite. As raízes foram, então, lavadas para eliminar o excesso de corante e recolocadas nos tubos, onde foram cobertas com solução de glicerina 1:1 (glicerina pura + água destilada) e deixadas em repouso por 2 h. Em seguida, as raízes foram colocadas em lâmina de vidro, sobre as quais foram adicionadas gotas de glicerina pura. Outra lâmina de vidro foi sobreposta e, assim, estando pronta para observação.

### 3. Uso de StageControl

Para o uso básico de StageControl (para acessar o código verificar o **material suplementar 1** e **apêndice 1**) é necessário especificar a movimentação que deve ocorrer para determinar a área total de escaneamento ou ponto específico que se deseja imagear. A obtenção da área total de escaneamento se dá através dos argumentos `x_start`, `x_end`, `y_start` e `y_end` e a distância entre captações, em milímetros, é fornecida pelos argumentos `step_x` e `step_y`. Também é necessário especificar as portas USB que o estágio e a câmera estão conectados.

Caso o usuário deseje começar o escaneamento do ponto inicial padrão, `x_start` e `y_start` assumem valor 0. Sempre será necessário especificar os pontos finais do escaneamento (`x_end` e `y_end`). Caso deseje começar de outro ponto, o usuário pode movimentar o estágio manualmente usando softwares livres como o Universal GCode Sender para obter as coordenadas da região de interesse.

Para especificar a distância entre captações de imagens o usuário deve ajustar os valores dos argumentos `step_x` e `step_y`. Dessa maneira é possível capturar imagens no intervalo espacial especificado pelo usuário. Esses argumentos também proporcionam uma sobreposição entre as imagens para que seja possível, posteriormente, juntar as imagens obtidas resultando em uma única imagem (processo de junção ou *stitching*) assim ampliando o campo de visão em relação à amostra.

A imagem do microscópio capturada através da câmera é originalmente redonda (para MicroStage), já que a captura da imagem se dá através de uma câmera posta acima da lente ocular. Dessa forma, é necessário realizar um corte na imagem para facilitar a posterior junção. É possível realizar esse ajuste através do argumento `crop_bound` na função `main` (**apêndice 1**). Esse passo não é necessário para o NemaStage tendo em vista que as imagens são adquiridas automaticamente via microscópio digital. O código desse software está apresentado no **apêndice 2**.

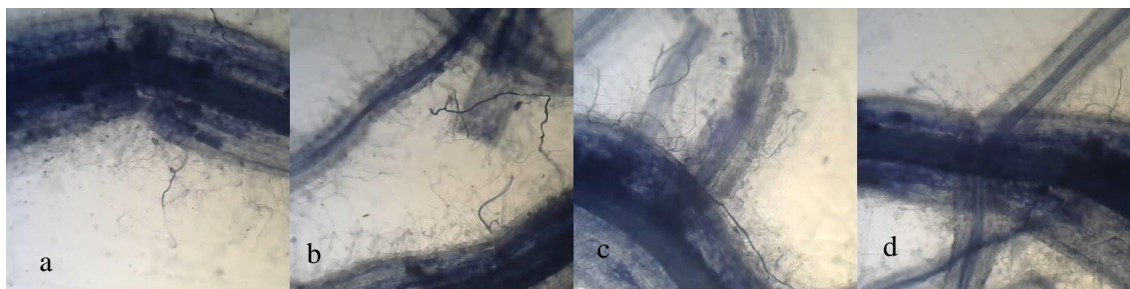
Para facilitar a junção de imagens utilizando o plugin *Stitching* (Preibisch et al., 2009) do software Fiji (Schneider et al., 2012), estas são automaticamente nomeadas no esquema "`nem_x_y_.png`", em que `x` é a posição da imagem ao longo do eixo X e `y` a posição no eixo Y. O formato PNG foi escolhido por preservar as informações originais da imagem e por ser amplamente utilizado.

## 4. Aplicação do método

### 4.1 Uso de MicroStage

Para testar o uso do estágio automatizado MicroStage, foi realizado o escaneamento de uma lâmina de microscopia contendo raízes de milho colonizadas por fungos micorrízicos arbusculares (FMA).

O escaneamento de uma lâmina ocorreu conforme as dimensões especificadas nos argumentos `xstart`, `xend`, `ystart` e `yend`, resultando em uma área total de escaneamento. Para que a junção das imagens ocorra de maneira satisfatória, se faz necessário o uso de sobreposição entre imagens consecutivas. Desta forma testamos valores para os argumentos `step_y` e `step_x`. Nestes testes foi observado que o valor 0.8 proporcionava junção satisfatória em tempo aplicável caso se deseje utilizar a abordagem de junção como rotina no laboratório (cerca de 20 minutos para junção destas imagens representando uma região). A **figura 3** traz alguns exemplos de imagens adquiridas no processo de escaneamento. Uma possível alternativa para melhoria da qualidade destas imagens é a utilização de uma câmera de melhor qualidade, como por exemplo uma de 1080p. Estas imagens foram obtidas em objetiva de ampliação de 5 vezes e ocular de 10 vezes..



**Figura 3.** Imagens de fragmentos de raízes de milho colonizadas por fungo micorrízico arbuscular. Nas figuras **3.a**, **3.b** e **3.c** é possível, por exemplo, observar a presença de vesículas de FMA. Nas figuras **3.a** e **3.b** é possível observar, também, algumas hifas.

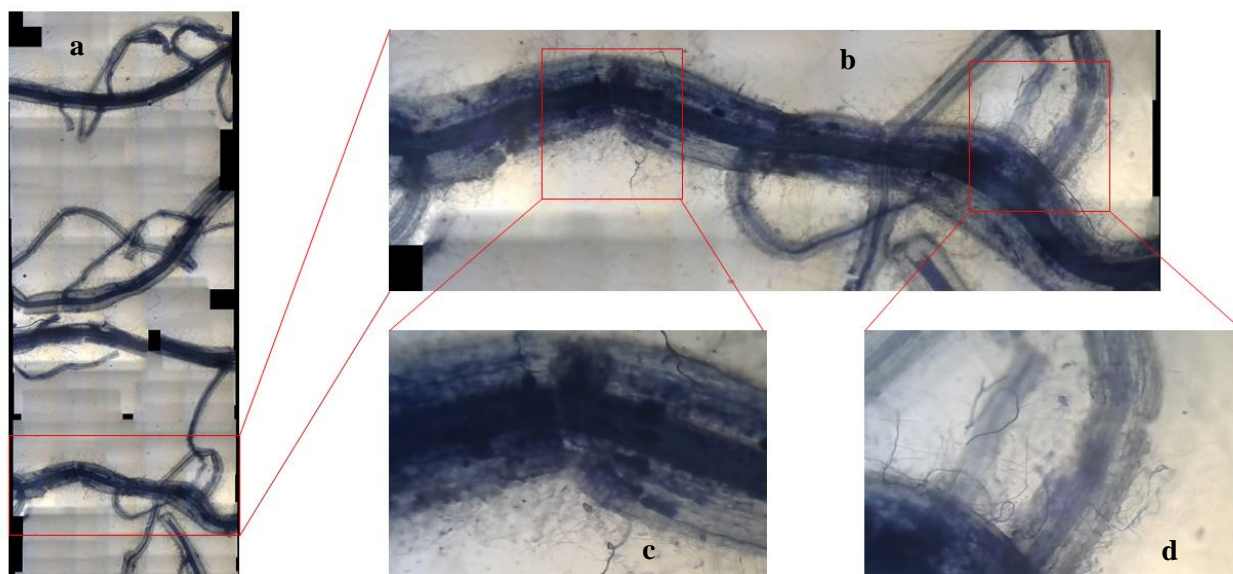
O processo de junção das imagens foi conduzido no software Fiji conforme a metodologia apresentada por Preibisch et al. (2009), utilizando uma grade na qual a posição das imagens é informada no próprio nome dos arquivos. Para que as imagens estivessem em formato passível de junção (imagens quadradas) foram utilizados no argumento `crop_bound` os valores 150, 650, 350, 850 o que resultou em imagens de dimensões 500 x 500 pixels, para essa câmera em particular. Para este teste de junção escolhemos apenas uma região e não a lâmina por completo devido a maior necessidade de processamento que a segunda opção representaria. O tempo de escaneamento da região especificada foi em torno de 20 minutos.

Como é possível escolher a lente objetiva e ocular que o usuário irá utilizar para observação da amostra e captura das imagens, uma série de ampliações são possíveis. Há de se ter em mente que conforme as ampliações aumentam, mais imagens são necessárias para cobrir a amostra por inteiro, o que demanda mais recursos

computacionais para o processamento, mais tempo para o escaneamento e mais espaço no armazenamento dos arquivos no computador. É necessário, então, que o usuário encontre um equilíbrio entre o detalhamento que as imagens proporcionam e o impacto que isso terá posteriormente em termos de uso de memória no computador. Para esse teste foi utilizada uma objetiva de 5x e ocular de 10x, obtendo-se resultados satisfatórios.

Utilizando essa abordagem de captação de imagens em uma área de varredura/escaneamento especificada para posterior junção, o campo de visão foi aumentado em comparação à abordagem convencional de captação de imagens em apenas alguns pontos. Dessa forma é possível retirar informações mais globais sobre a amostra, aumentando o poder de análise. Por exemplo, caso o usuário deseje criar um índice de colonização da raiz por fungos micorrízicos baseado na análise de imagens, será necessária uma estratégia de junção de imagens (caso se trabalhe com ampliações que não permitam que uma só imagem contenha todo o fragmento de raiz), já que esta reconstruiria a imagem da raiz (contendo estruturas fúngicas) que originalmente estava fragmentada. Outro exemplo é a simples medição de uma estrutura, como, por exemplo, de uma hifa. Nas abordagens tradicionais essa estrutura pode estar presente em múltiplas imagens. Neste contexto uma hifa só poderia ser considerada mais de uma hifa, e a medição estaria sujeita a erros. Essa estratégia também pode ser combinada a técnicas de visão computacional para fazer a segmentação automática das estruturas fúngicas e calcular a colonização micorrízica.

A **figura 4.a** traz um exemplo de junção de imagens de uma região da lâmina contendo raízes colonizadas por FMA que ampliou o campo de visão em relação à amostra. Nesta junção 120 imagens individuais foram fundidas para reconstrução de uma região da lâmina. Na **figura 4.b** uma das raízes ali contidas foi ampliada e nas **figuras 4.c e 4.d** estão ampliadas imagens nas quais é possível a observação de estruturas de micorriza como arbúsculos e hifas. Na imagem original de junção, que resulta do Stitching por posição do software (Preibisch et al., 2009), armazenada no computador, é possível alcançar esse detalhamento (das **figuras 4.c e 4.d**) através do zoom do visualizador de imagens utilizado pelo usuário.



**Figura 4.** Imagens de fragmentos de raízes obtidos através do escaneamento automático utilizando o software StageControl e o estágio MicroStage. Na **figura 4.a** estão contidos quatro fragmentos de raízes que foram reconstruídos através da junção de 120 imagens individuais. Na **figura 4.b** uma destas regiões foi ampliada. Nas **figuras 4.c e 4.d** imagens individuais que compõe a imagem ampla são exemplificadas.



### 4.1.1 Limitações da abordagem

Alguns problemas foram encontrados nos testes de MicroStage. Um deles foi que por vezes o foco não foi mantido uniformemente em todas as imagens captadas ao longo do escaneamento. Esse problema é negligível em magnificações menores (objetivas de 5, 10 ou 20x), mas pode ser um problema na objetiva de 40x. Uma maneira de evitá-lo é fazer pressão sobre a lâmina para que a altura da amostra seja homogênea.

Quanto às limitações da nossa abordagem podemos listar:

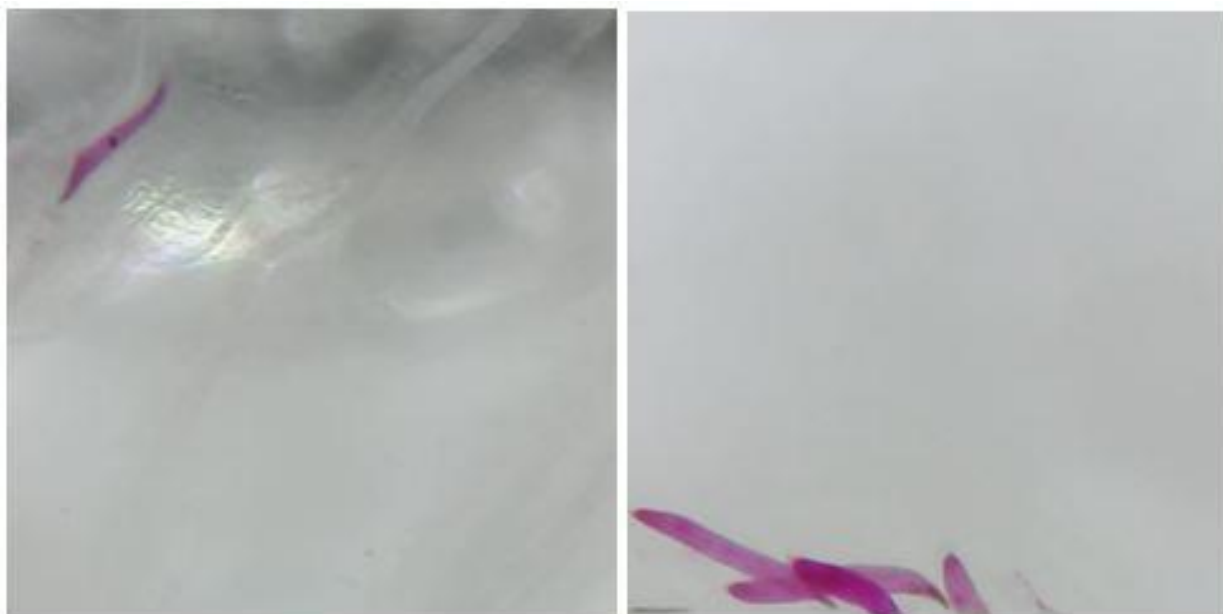
- i) Necessidade de o operador ter conhecimento de básico a intermediário na linguagem de programação Python para manusear o sistema de forma personalizada às necessidades da amostra analisada. Esse problema pode ser resolvido em versões futuras do software apresentado, quando poderá ser criada uma interface gráfica mais amigável;
- ii) Se faz necessário utilizar um software auxiliar para conhecer os pontos iniciais (quando o usuário não deseja começar o escaneamento no ponto inicial padrão) e os pontos finais. Softwares auxiliares como o Universal Gcode Sender podem ser utilizados para obtenção desses dados.
- iii) A junção de imagens é realizada em outro software. Na condução desses testes essa junção foi realizada no Fiji utilizando o plugin *Stitching* (Preibisch et al., 2009) por nome de arquivo. Em média foram necessários mais de 20 minutos para a realização dessa tarefa quando se juntou apenas uma região. Este tempo seria maior no caso de uma imagem que representasse uma região maior (como por exemplo a totalidade da lâmina).
- iv) Alguns erros ocorreram no processo de junção. Por vezes algumas regiões não foram juntas de maneira exata e ocorreram pontos coloridos ou pretos onde originalmente deveria haver uma imagem. A calibração do grau de sobreposição entre as imagens e de alguns parâmetros do plugin *Stitching* podem sanar esses problemas.

## 4.2 Uso de NemaStage

O método apresentado acima não é aplicável para as lâminas (com dimensões aproximadas de 60 x 200mm) frequentemente utilizadas para estudo de nematoides parasitas de plantas, devido às dimensões muito maiores (cerca de seis vezes) em relação às lâminas comuns (25 x 75mm). Portanto, nós desenvolvemos um equipamento, chamado de NemaStage, com área útil superior para acomodar essas amostras. Para controlar esse estágio também foi utilizado o software StageControl com algumas modificações. A função de escaneamento está contida no **apêndice 3** e o código que definiu as classes utilizadas no processo de escaneamento estão contidas no **apêndice 4**. O escaneamento total de uma lâmina utilizada neste experimento é de cerca de uma hora.

Na magnificação testada cerca de 1000x, foi necessário ajustar os argumentos `step_x` e `step_y` para 1 de forma que houvesse sobreposição satisfatória para que o

processo de junção ocorresse de forma apropriada. A figura 5 exemplifica imagens obtidas no escaneamento. Julgamos que a magnificação proporcionada pelo microscópio foi satisfatória para o imageamento de nematoides, permitindo uma visualização nítida dos nematoides adultos e juvenis. O microscópio digital utilizado constitui uma opção



**Figura 5.** Exemplo de imagens contendo nematoides captadas através do uso de NemaStage.

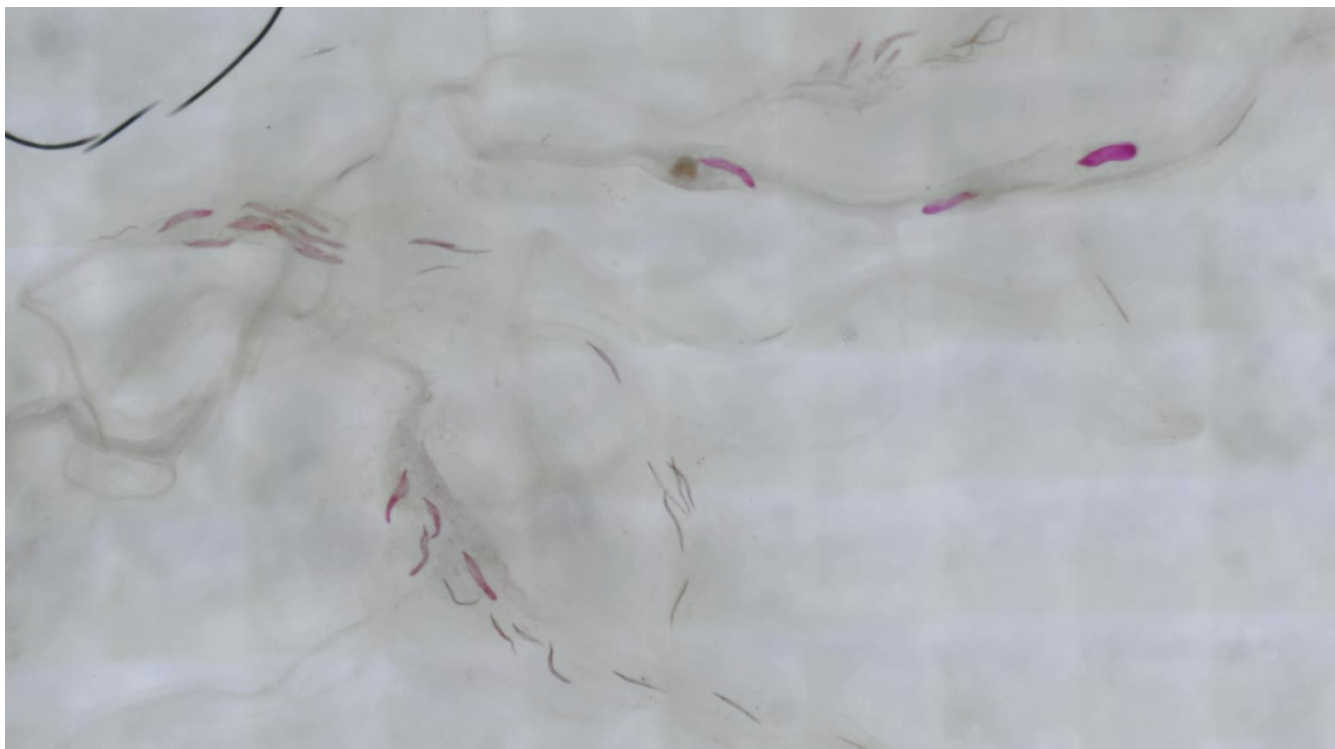
muito mais barata que os microscópios óticos tradicionalmente utilizados na análise destes organismos (Goulart, 2010), já que o microscópio utilizado neste experimento custa R\$112,00 reais enquanto os tradicionais custam pelo menos alguns milhares de reais. Além do custo, NemaStage apresenta a vantagem de não necessitar adaptações para o tamanho da lâmina trabalhada neste experimento. Além disso, nestes microscópios de estágio menor, a estratégia de imageamento automático se torna impraticável para nematoides. Este problema não ocorre com NemaStage, tornando-o mais atraente que as abordagens tradicionais utilizadas no imageamento de nematoides.

Foi necessário ajustar os argumentos `step_x` e `step_y` para 1 de forma que houvesse sobreposição satisfatória para que o processo de junção ocorresse de forma apropriada. A figura 5 exemplifica imagens obtidas no escaneamento. Julgamos que a magnificação proporcionada pelo microscópio (de até 100x) foi satisfatória para o imageamento de nematoides, constituindo uma opção mais barata que os microscópios óticos tradicionalmente utilizados na análise destes organismos (Goulart, 2010). Além disso, para o tamanho da lâmina utilizada (e, por conseguinte da necessidade de longa distância de travel necessária que o estágio deve possuir para realizar o imageamento destes organismos) NemaStage se apresenta como vantajoso ao não exigir adaptações na operação deste equipamento como pode ocorrer para o imageamento em microscópios óticos tradicionais nos quais o estágio pode ter uma dimensão menor que o da placa. Neste cenário (estágio menor que a placa) a ampliação do campo de visão através da junção de imagens capturadas em diferentes posições se torna impossível. Este problema não ocorre com NemaStage, tornando-o mais atraente que as abordagens tradicionais utilizadas no imageamento de nematoides.

Utilizando o hardware NemaStage e o software StageControl é possível especificar os pontos iniciais e finais para a coleta de imagens, o que proporciona a

captação de imagens de uma lâmina por completo, ou mesmo apenas uma região de interesse. Essa abordagem se mostrou eficiente para o estudo de nematoides e apresentou a vantagem de custos reduzidos para o imageamento já que utilizamos equipamentos de baixo custo e softwares e hardwares de uso livre que podem ser adaptados pelo usuário para melhor se adequarem às condições locais de acesso aos componentes do equipamento bem como da amostra que se deseja analisar. Desta forma, apresentamos uma interessante opção de estágio automatizado e software de controle, sobretudo para laboratórios e instituições com maior limitação financeira. A **figura 6** apresenta um exemplo do resultado de junção de imagens, que ampliou o campo de visão em relação à amostra, de uma região contendo nematoides. Para NemaStage o processo de junção de imagens se deu através do plugin Stitching (Preibisch et al., 2009).

Julgamos que a estratégia de junção de imagens capturadas durante o escaneamento da amostra contida na lâmina apresentada neste artigo é uma importante contribuição para o estudo de nematoides. Esta estratégia de junção torna-se imprescindível quando se utiliza escaneamento automático já que, quase invariavelmente, ocorrerá a presença de um (ou mais) nematóide que está presente em duas ou mais imagens. Sem a junção, operações de contagem e medição destes organismos se torna imprecisa já que o usuário pode contar um único nematoide como mais de um, além de realizar operações de medição de maneira incorreta já que o usuário pode medir apenas parte do organismo uma vez que sua parte complementar está presente em outra imagem. Através de análises de morfologia (como largura e comprimento) que são proporcionadas, em última instância pela junção de imagens que contenham estes organismos quando se trabalha com estratégia de escanemaento automático, podem ser tomadas melhores decisões já que características morfológicas como comprimento e largura podem ser utilizadas no auxílio da identificação do sexo e estágio dos nematóides (Garcia e Puerta, 2012).



**Figura 6.** Exemplo de imagem composta contendo regiões com nematoides.

### 4.2.1 Limitações da abordagem

Ao longo da condução dos testes de uso de NemaStage encontramos algumas limitações. A seguir elas estão listadas.

- i) Para a operação do software é necessário conhecimento básico da linguagem de programação Python, o que pode se tornar uma limitação para muitos usuários da comunidade científica que atua no campo da biologia e ciências agrárias, cuja maioria não possui essa familiaridade. Esse problema pode ser superado através da criação de uma interface gráfica.
- ii) É necessário utilizar um software auxiliar para obtenção das coordenadas em que se deseja iniciar (caso não queira começar do zero padrão) e terminar o escaneamento.
- iii) O método de junção de imagens utilizado nesse capítulo é muito ineficiente para amostras grandes, levando-se em conta que a maior parte da imagem não contém nematóides e, portanto, o esforço computacional para processar essas imagens é desperdiçado. Em razão disso, propomos um método muito mais eficiente baseado em inteligência artificial no próximo capítulo.

## 5. Custos de NemaStage

Um dos pontos mais importantes a favor do uso da ciência aberta para confecção de microscópios é o custo dos aparelhos convencionais. Um estágio automatizado pode custar mais de R\$40.000,00. Esse valor impede que muitos laboratórios, tanto privados quanto públicos, possam fazer uso desta tecnologia. Além disso, estes equipamentos são comprados muitas vezes no exterior, em dólar americano, e requerem mão-de-obra especializada para sua manutenção. Logo, o valor pode oscilar bastante conforme o câmbio em relação ao real brasileiro ou a moeda do país em que o laboratório está localizado. Esse cenário resulta em um menor poder de análise nessas instituições. Por exemplo, com o escaneamento e posterior obtenção de uma imagem única mais ampla de raízes micorrizadas, o estado da colonização pode ser melhor estudado do que quando as imagens são avaliadas individualmente. Apresentamos neste artigo equipamentos que mantêm elevadas capacidades de análise, mas que, ao mesmo tempo, possuem custos muito menores que os disponíveis no mercado. A **tabela 1** traz os custos aproximados para obtenção de cada um dos componentes utilizados para a construção do equipamento NemaStage. Além de seu baixo custo, por se tratar de um equipamento com software e hardware livres, o usuário pode realizar uma série de adaptações às condições locais, por exemplo de equipamentos pré-existentes no laboratório que podem ser reaproveitados para a fabricação deste equipamento, e do organismo que se deseja avaliar.

Item	Preço Unitário	Quantidade	Valor
Filamento ABS	R\$ 85,00	0,3	R\$ 25,50
Fuso 8mm	R\$ 0,07	1	R\$ 0,07
Barra Inox Usinado 8mm	R\$ 8,67	3	R\$ 26,01
Arduino Nano	R\$ 45,00	1	R\$ 45,00
Microscópio digital	R\$ 112,33	1	R\$ 112,33
CNC shield	R\$ 30,00	1	R\$ 30,00
Fonte 12V 3A	R\$ 20,00	1	R\$ 20,00
<b>Total</b>			<b>R\$ 258,91</b>

**Tabela 1.** Custos totais de confecção do estágio automatizado NemaStage

### Referências

- Bernas, T., Robinson, J.P., Asem, E.K.; Rajwa, B. 2005. Loss of image quality in photobleaching during microscopic imaging of fluorescent probes bound to chromatin. **Journal of Biomedical Optics**, v. 10, n. 6, p. 064015.
- Byrd, D.W., Kirkpatrick, T.J.R. Bakker, K.R. 1983. An improved technique for clearing and staining plant tissues for detection of nematodes. **Journal of Nematology**, v. 15, p. 142-143.
- García, L.E., Sánchez-Puerta, M.V. 2012. Characterization of a root-knot nematode population of *Meloidogyne arenaria* from Tupungato (Mendoza, Argentina). **Journal of Nematology**, v. 44, n. 3, p. 291–301.
- Goulart, A.M.C. 2010. Análise Nematológica: Importância e princípios gerais. **Documento 299, Embrapa Cerrados**.
- Hussey, R.S., Barker, K.R. 1973. A comparison of methods for collecting inocula of *Meloidogyne* spp. including a new technique. **Plant Disease Reporter**, v. 57, p.1025-1128.
- JetBrains. 2020. PyCharm. Disponível em <<https://www.jetbrains.com/pycharm>>.
- Koske, R.E.; Gemma, J.N. 1989. A modified procedure for staining roots to detect VA mycorrhizas. **Mycological Research**, v. 92, n. 4, p. 486–488. British Mycological Society. Disponível em: <[http://dx.doi.org/10.1016/S0953-7562\(89\)80195-9](http://dx.doi.org/10.1016/S0953-7562(89)80195-9)>. .
- Ma, X., An, G. Li, B. 2009. Design and implementation of an automated microscope stage. **Proceedings - 2009 International Forum on Information Technology and Applications, IFITA 2009**, v. 1, p. 603–605.
- Malkin, R.A. 2007. Barriers for medical devices for the developing world. **Expert Review of Medical Devices**, v. 4, n. 6, p. 759–763.
- Malkin, R., Keane, A. 2010. Evidence-based approach to the maintenance of laboratory and medical equipment in resource-poor settings. **Medical and Biological Engineering**

**and Computing**, v. 48, n. 7, p. 721–726.

Preibisch, S., Saalfeld, S., Tomancak, P. 2009. Globally optimal stitching of tiled 3D microscopic image acquisitions. **Bioinformatics**, v. 25, n. 11, p. 1463–1465.

Thompson, S.K., Mason, E., Dukes, S. 2003. Ergonomics and Cytotechnologists: Reported Musculoskeletal Discomfort. **Diagnostic Cytopathology**, v. 29, n. 6, p. 364–367.

Wollman, R., Stuurman, N. 2007. High throughput microscopy: From raw images to discoveries. **Journal of Cell Science**, v. 120, n. 21, p. 3715–3722.

Yuan, J., Wang, X., Zhou, H., Li, Y., Zhang, J., Yu, S., Wang, M., Hao, M., Zhao, Q., Liu, L., Li, M., Li, J. 2020. Comparison of Sample Preparation Techniques for Inspection of Leaf Epidermises Using Light Microscopy and Scanning Electronic Microscopy. **Frontiers in Plant Science**, v. 11, n. February, p. 1–13.

## ARTIGO II

### **Imageamento e estudo de nematoides assistido por visão computacional utilizando ciência aberta e equipamentos de baixo custo**

#### **Abstract**

Por conta dos prejuízos causados na atividade agrícola mundial, o estudo dos nematoides causadores de doenças em plantas é um tópico de enorme interesse. Várias abordagens e equipamentos podem ser utilizados, como a avaliação do efeito destes organismos sobre as plantas e avaliação de características morfológicas. Recentemente tecnologias que envolvem aprendizado de máquina foram desenvolvidas para esse fim. Algumas barreiras são encontradas no estudo destes organismos. Por exemplo, os estágios tradicionais dos microscópios são pequenos demais para o tamanho das lâminas utilizadas. Esse cenário impossibilita, por exemplo, o escaneamento total destas lâminas para construção de uma imagem ampla. Quando se trabalha com escaneamento total, se faz necessária uma abordagem de junção das imagens obtidas não apenas para a ampliação no campo de visão, mas também para que operações como medida dos organismos ali presente possam ser feitas. Outro problema quando se utiliza o escaneamento total é o elevado número de imagens que resultam desse processo, mas que não contém nenhum objeto de interesse, o que, em última instância torna o processo de junção de imagens bastante dispendioso tanto em relação ao tempo gasto quanto aos recursos computacionais necessários, o que pode tornar esse processo de junção impeditivo. Apresentamos neste artigo um algoritmo, baseado em uma rede neural artificial, capaz de realizar de forma eficiente a junção de imagens resultantes do escaneamento de raízes parasitadas por nematóides, reduzindo em mais de 90%, nos testes realizados, o número de imagens a serem avaliadas. Adicionalmente apresentamos um software para medida de comprimento e largura de nematoides presentes em imagens digitais. Todos os resultados desse trabalho foram obtidos utilizando equipamentos de baixo custo e software de código aberto, acessíveis a laboratórios com restrições orçamentárias.

**Palavras-chave:** Meloydogine javanica, Microscopia, Ciência Aberta, Aprendizado de Máquina, UNet.

O artigo científico se encontra nas normas da revista “Computers and Electronics in Agriculture” e está sujeito a alterações.

## 1. Introdução

Nematoides são importantes organismos do solo, com papel fundamental para o equilíbrio da teia alimentar (Kapagianni et al., 2010; Moura e Franzener, 2017). Apesar de sua importância ambiental, esses organismos são mais amplamente conhecidos pelos impactos negativos que podem trazer às culturas (soja, tomate, melão, etc.) ao parasitar e às vezes causar doenças a essas plantas (Abdeldaym et al., 2014; Gomes et al., 2011; Kirwa et al., 2018; Liu et al., 2012). Estima-se que o prejuízo financeiro causado por esses organismos, anualmente, na agricultura mundial é de quase 125 bilhões de dólares (Mesa-Valle et al., 2020).

Os esforços para o combate desse organismo são focados em detectar e controlar o mais rapidamente possível a infecção antes que ela se alastre pelo restante da lavoura. Para isso uma série de estratégias podem ser implementadas, como, por exemplo, a aplicação de defensivos agrícolas e a avaliação molecular do estado da planta de forma que seja verificada a ativação e ação dos mecanismos de defesa (Rahman et al., 2010; Sasanelli et al., 2021). Grande parte dos parâmetros avaliados nos estudos que visam entender o efeito desses organismos sobre as plantas diz respeito apenas aos efeitos dos nematoides sobre o desenvolvimento vegetal, como por exemplo sobre o vigor, comprimento, massa de frutos e os efeitos sobre o estado nutricional da planta através da análise do tecido vegetal (Bicalho et al., 2021; Conduta et al., 2020; Zinger et al., 2021). Quando são avaliados parâmetros relacionados aos nematoides, estes são simples como a contagem desses organismos, avaliação do estágio (geralmente verificar o número de juvenis), contagem de ovos e ovos chocados (Ngala et al., 2021; Seid et al., 2015). Avaliações como a medição do comprimento e largura podem ser realizadas, porém são bastante trabalhosas, já que a amostra tem que ser posta para observação no microscópio e cuidadosamente avaliada com o uso de uma lente objetiva específica com graduação (Skantar et al., 2008; Van Bezooijen, 2006). Além disso, esta lente não é um equipamento padrão que acompanha o microscópio quando comprado. Ou seja, representa mais um custo adicional. Uma alternativa para solucionar estas limitações é realizar essas análises baseadas em imagens.

Por mais simples que seja a análise realizada sobre o nematoide, é necessário o uso de microscópio. O custo financeiro desse equipamento é bastante variado, existindo as opções mais baratas (cerca de R\$ 100,00) até as opções de custo mais elevado (dezenas de milhares de reais). Uma das peças fundamentais para o funcionamento desse equipamento é o estágio, que é uma peça que permite a movimentação da lâmina que está sendo submetida à análise no microscópio, permitindo, assim, a observação das diferentes regiões da amostra.

É possível automatizar a movimentação desta peça, de forma que o usuário não precise movimentá-la manualmente. Com o uso dessa abordagem é possível explorar melhor a amostra analisada já que o usuário aumenta a capacidade de observação ao combinar imagens captadas separadamente, assim, ampliando o campo de visão. Outra vantagem é a possibilidade de analisar em diferentes intervalos de tempo a mesma estrutura já que o usuário pode especificar a localização de captação de maneira exata (Collins et al., 2020). Nesta condição, este escaneamento se torna mais eficiente já que é possível determinar a área total em que ele será realizado através do ajuste da movimentação e da distância para captação entre uma imagem e outra através de especificações do usuário via software.



Uma dificuldade para automatização do escaneamento é o alto custo do estágio automatizado, que pode custar mais de R\$40.000,00 tornando sua utilização impraticável para laboratórios com maior limitação de recursos financeiros. Uma possível solução para a popularização da utilização deste equipamento pode se dar no escopo da ciência aberta. Dessa forma o pesquisador pode utilizar hardwares e softwares de uso livre, o que torna o custo do estágio e demais peças muito menores (Campbell et al., 2014; Collins et al., 2020).

Posteriormente ao escaneamento, as imagens individuais precisam ser combinadas para formar uma imagem mais ampla. Tradicionalmente, quando se utiliza a estratégia de escaneamento de uma placa por completo, são armazenadas todas as imagens captadas no processo, a despeito de possuírem ou não o objeto de interesse, nesse caso o nematóide. Isso eleva desnecessariamente a demanda por recursos computacionais e tempo para processar essas imagens e fazer a junção delas. Uma das estratégias que podem ser utilizadas para solucionar este problema é realizar a junção apenas de imagens que possuem os objetos de interesse (nematóides).

Para a identificação de imagens contendo as estruturas de interesse, algoritmos podem ser treinados para automação desta tarefa como, por exemplo, algoritmos baseados em aprendizado de máquina (*machine learning*). Modelos de aprendizado de máquina já são utilizados na análise de amostras biológicas. Técnicas de visão computacional (uma subdivisão do aprendizado de máquina) acopladas a estes modelos são utilizadas amplamente para classificação e segmentação de estruturas biológicas (Fischer et al., 2020; Korfhage et al., 2020; Weber et al., 2020). Na nematologia há exemplos recentes de aplicação dessa técnica, como Chen et al. (2020), que relatam um algoritmo capaz de segmentar estruturas de nematoides, e Thevenoux et al. (2021), que reportam um algoritmo de classificação entre duas espécies de nematoides.

Apresentamos neste artigo software de acesso livre desenvolvidos nesta pesquisa para possibilitar o uso de um microscópio acoplado a um estágio automatizado, ambos de baixo custo. Além disso é apresentado um algoritmo de aprendizado de máquina, utilizando como arquitetura de rede neural artificial uma UNet modificada (Ronneberger et al., 2015), que identifica nematoides em imagens para otimizar o processo de combinação das imagens resultantes do escaneamento para compor imagens amplas. Adicionalmente, apresentamos um algoritmo que realiza medidas de comprimento e largura em nematoides em imagens digitais.

## **2. Metodologia**

### **2.1 Obtenção de juvenis de segundo estágio (J2) de *Meloidogyne javanica***

Para os testes citados à posteriori foram utilizadas duas lâminas de microscopia de nematoides. A descrição da obtenção destas lâminas é relatada nos parágrafos e no tópico seguinte.

Os juvenis de segundo estágio (J2) utilizados no experimento foram obtidos de populações puras de *Meloidogyne javanica* multiplicadas em plantas de tomate (*Solanum lycopersicum* L.) mantidas em casa de vegetação do Laboratório de Nematologia - UFLA.

A suspensão de ovos de *M. javanica* foi obtida conforme a técnica de Hussey e Barker (1973). Para tal, as raízes de tomate foram cuidadosamente lavadas em água parada para retirar partículas de substrato aderido, picadas em pedaços de aproximadamente 1 a 2 cm e trituradas em liquidificador com solução de hipoclorito de sódio (NaClO) 0,5% por, aproximadamente, 40 segundos. A suspensão de ovos e raízes foi vertida em jogo de peneiras de 0,074 mm de abertura (200 “mesh”), acoplada à de 0,025 mm (500 “mesh”), ficando os ovos retidos nesta última. Para a obtenção dos J2 a suspensão de ovos foi incubada em câmara de eclosão a 28 °C, sendo utilizados apenas os J2 eclodidos após 24 horas da montagem da câmara.

## **2.2 Penetração de juvenis de segundo estágio (J2) de *M. javanica* em raízes de soja**

Sementes de soja (*Glycine max*), cultivar Monsoy 5917, foram desinfestadas superficialmente em solução de NaClO (1%) por 1 minuto, colocadas para germinar em bandejas contendo areia umedecida e mantidas em sala climatizada a 28 °C. Decorridos cinco dias, as sementes germinadas foram transferidas para as células de 121 mL de bandejas de isopor com 72 cavidades, preenchidas com substrato Tropstrato. Após sete dias do transplante infestou-se o substrato com suspensão aquosa de três mL contendo 500 J2 de *M. javanica*. Em seguida, a bandeja contendo as plantas inoculadas foi colocada em sala climatizada com temperatura de 27 °C. Sete dias após a infestação, as plantas foram retiradas da bandeja. Em seguida, o sistema radicular foi lavado em água corrente e submetido ao clareamento em solução de NaClO (1,5%) por 10 min. Na sequência, as raízes foram lavadas em água corrente para eliminar o resíduo de NaClO. Em seguida, elas foram transferidas para tubos de vidro onde foram cobertas ( $\pm 15$  mL) com uma solução corante, preparada com 3,5 g de fucsina ácida + 250 mL de ácido láctico + 750 mL de água destilada (adaptado de Byrd et al., 1983) e diluída 1:29 (1 mL de solução corante inicial + 29 mL de água destilada). Então os tubos de vidro foram colocados em chapa aquecedora e mantidos neste equipamento até a ebulição da solução corante. O resfriamento foi feito em condições ambiente durante uma noite. As raízes foram, então, lavadas para eliminar o excesso de corante e recolocadas nos tubos, onde foram cobertas com solução de glicerina 1:1 (glicerina pura + água destilada) e deixadas em repouso por 2 h. Em seguida, as raízes foram colocadas em lâmina de vidro, sobre as quais foram adicionadas gotas de glicerina pura. Outra lâmina de vidro foi sobreposta e, assim, estando pronta para observação.

## **2.3 Microscópio, estágio automatizado e software de controle**

Para o escaneamento das lâminas (com dimensões de 6 x 20cm) contendo raízes parasitadas por nematóides, foi utilizado um estágio automatizado confeccionado no Laboratório de Microbiologia do Solo da Universidade Federal de Lavras (NemaStage), conforme apresentado no capítulo anterior. Brevemente, esse estágio consiste de dois eixos lineares montados sobre uma base de MDF. A movimentação desses eixos se dá através de fusos acoplados a motores de passo acionados por um microcontrolador arduino. Os comandos para controle do posicionamento desse estágio e o processo de escaneamento ocorre através do software Stagecontrol (Para mais informações sobre o software acessar a **seção 2.3** do **artigo 1** desta dissertação). Nesse software, escrito na linguagem de programação Python, comandos são enviados em g-code para o microcontrolador, responsável por acionar a movimentação dos motores. Esse software,

através da biblioteca OpenCV, simultaneamente controla a aquisição de imagens por meio de um microscópio digital conectado ao computador via USB. Com o uso desse software é possível automatizar o movimento do estágio, especificar as posições para captação da imagem (possibilitando que a região seja analisada em diferentes intervalos de tempo), bem como realizar o escaneamento completo de uma lâmina ou de uma determinada região de interesse. Mais informações sobre o estágio e o software estão contidas no **artigo 1**.

Para a captação das imagens foi utilizado um microscópio digital da marca Sazoley, importado da China, com câmera acoplada.

## 2.4 Segmentação e obtenção de medidas de nematoides em imagens

### 2.4.1 Segmentação de nematoides

Para a segmentação dos nematoides (processo de divisão da imagem em objetos de interesse e fundo) foi utilizado um algoritmo de rede neural convolucional (RNC) para a obtenção de um modelo que fosse capaz de prever a presença de nematoides. Foi escolhida como arquitetura para implementação desta rede uma UNet modificada (Ronneberger et al., 2015). Essa rede totalizou mais de 7 milhões de parâmetros para treinamento. A arquitetura completa da rede e o código, implementada através do pacote Keras, podem ser acessados no **apêndice 5**.

Posteriormente à obtenção do modelo, este é utilizado para confeccionar as máscaras de predição. Neste processo ocorrem algumas operações de filtragem (como exclusão de pequenas regiões). O código para tal está contido na segunda metade do **apêndice 5**.

Para o treinamento da RNC a rede foi alimentada com entradas (imagens) de dimensões 480 x 480 x 3. Escolheu-se trabalhar com três canais (imagem colorida), pois essa estratégia apresentou o melhor resultado nos testes de segmentação quando comparado a utilização de apenas um canal (escala de cinza). A escolha por essas dimensões se deu pelo melhor detalhamento proporcionado ao se trabalhar com essas medidas já que representava as dimensões originais em que as imagens foram captadas.

O modelo foi obtido através de 97 imagens usadas durante o treinamento, e suas respectivas máscaras, e 44 imagens de validação (com suas respectivas máscaras). Por máscara é entendido uma imagem binária em que todos os pixels da imagem são classificados em pertencentes ou não ao objeto de interesse (no nosso caso nematoides). Nesse experimento, os pixels da máscara assumiram valor 255 (representado pela cor branca) quando os pixels da imagem correspondente faziam parte de nematoides, ou 0 (representado pela cor preta) em casos negativos. As máscaras foram obtidas com o auxílio do software Via (Dutta e Zisserman, 2019). Um exemplo de máscara está contido na **figura 2.a**. Neste software são anotadas as coordenadas das regiões de interesse que constituem a máscara sendo necessário posteriormente reconstruir a máscara através destes pontos. O código utilizado para esse fim está presente no **apêndice 6**.

Como uma rede neural convolucional necessita de um número de dados muito maior para treinamento do que o conjunto de 141 dados (imagens) que utilizamos inicialmente, foi utilizada a estratégia de aumento de dados (*data augmentation*).

Nessa estratégia as imagens são artificialmente modificadas de forma que seja aumentada a variabilidade do conjunto de dados. Em última instância esse processo leva ao aumento de precisão da predição do modelo para conjuntos de dados diferentes daqueles utilizados no treinamento, evitando o sobreajuste (overfitting) (Shorten e Khoshgoftaar, 2019).

Para o treinamento da rede neural foi utilizado o Google Colab. Para que fosse possível o treinamento em tempo aceitável, foi utilizada uma unidade de processamento gráfico GPU Tesla T4 disponibilizada gratuitamente pela plataforma. Para verificar a eficiência do modelo quanto à segmentação (predição de estruturas de nematoides) foi utilizado o índice RMSE (raiz do erro quadrático médio) calculado conforme a equação a seguir:

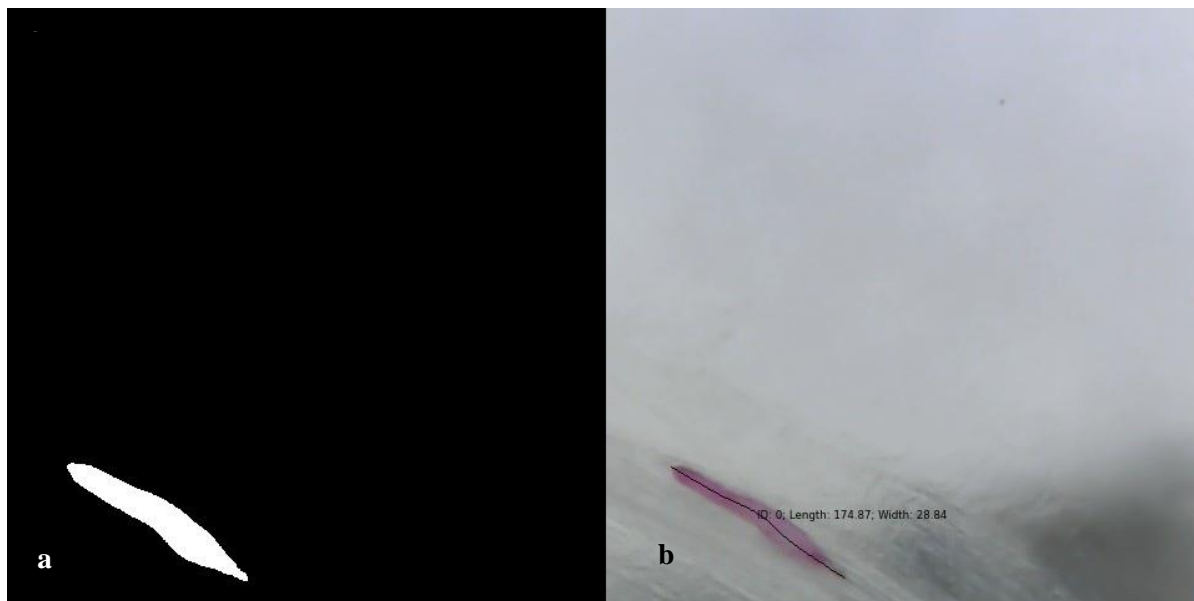
$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predito_i - Observado_i)^2}{N}}$$

Quanto menor for o valor de RMSE, mais similar é a máscara predita pelo modelo em relação à máscara gerada manualmente, logo mais acurado é o modelo. No Keras, foi utilizado como o valor 2 para o parâmetro *patience*, fazendo com que, após dois ciclos de treinamento (*epoch*), caso não fosse diminuído o valor de RMSE, o treinamento cessasse. Assim, o modelo com o último menor valor de RMSE obtido seria salvo para uso posterior e não haveria novas tentativas de obtenção de menor valor para este parâmetro. O modelo foi desenhado para ter 50 épocas de treinamento, porém foram necessários apenas 17. O tempo total para treinamento do modelo ficou entre 3 e 4 horas.

Após o modelo ter sido obtido foi necessária a testagem da capacidade de detecção e segmentação de nematoides em uma matriz de imagens não utilizadas para o treinamento do modelo, para verificar a capacidade de generalização dos resultados. Esses dados foram adicionais ao conjunto de dados usado na validação, compondo um conjunto de dados de teste, composto por oito imagens.

#### 2.4.2 Medição dos nematoides

Para análise de largura e comprimento de nematoides foi desenvolvido o software MeasureNema. Neste software as imagens originais são binarizadas (**figura 1.a**) e posteriormente esqueletonizadas (redução de um objeto a uma amplitude de pixels conectados conforme o objeto se expande conforme a **figura 1.b**) e são medidas quanto ao número de pixels do comprimento e da largura. O código de NemaMeasure é aberto e está disponível no **apêndice 7**.



**Figura 2** Exemplo de imagem binária obtida após o uso do modelo de precisão (**2.a**) e do esqueleto (linha preta dentro do nematoide) sobreposto ao nematoide na figura **2.b**.

Para verificar a precisão das medidas realizadas através do uso de NemaMeasure, inicialmente foi realizada a medição manual dos nematoides a partir das imagens utilizando o software ImageJ (Schneider et al., 2012). Um modelo de regressão linear foi aplicado para verificar o grau de concordância entre valores de largura e comprimento obtidos a partir de 83 nematoides medidos automaticamente (variável preditora) e os valores medidos manualmente (variável predita), utilizando o R (RStudio Team, 2020).

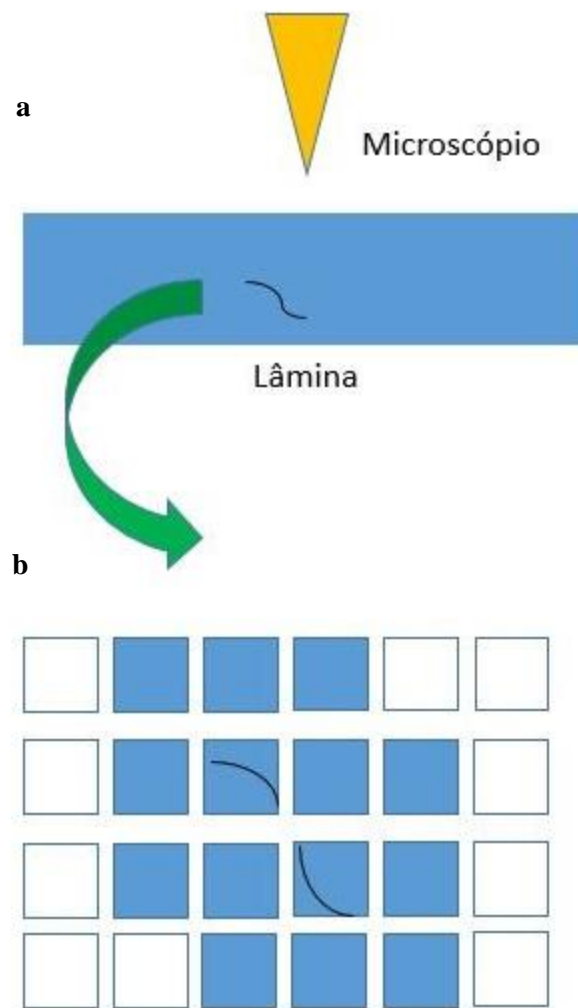
## 2.5 Detecção de imagens contíguas de nematoides e junção

No processo de escaneamento da lâmina, devido ao pequeno campo focal do microscópio em comparação com a área total da lâmina, é necessário gerar múltiplas imagens para varrer toda a área de interesse, resultando em um imageamento fragmentado. Entretanto, para posterior análise das imagens (e.g. contagem de nematoides, classificação quanto ao estágio e medidas), é necessário reconstruir uma imagem mais ampla da amostra através da combinação das imagens fragmentadas obtidas no escaneamento. A demanda de recursos computacionais para executar esse processo, entretanto, pode tornar-se proibitiva à medida que o número de fragmentos aumenta, o que depende do tamanho da amostra. Adicionalmente, grande parte destas imagens obtidas, não possuem os objetos de interesse (nematoides). Por essa razão, desenvolvemos um algoritmo que reconstrói apenas as áreas de interesse, ou seja, realiza a junção de imagens que contém nematoides. O código para esse software está disponível no **apêndice 8**, bem como na url (<https://github.com/teodecarvalho/nematodes/blob/main/stitching.py>).

No algoritmo desenvolvido, a junção se dá apenas entre imagens que contém nematoides e suas vizinhas imediatas. Estas vizinhas podem ser imagens que contém nematoides de maneira contígua (ou seja, um nematoide que, na verdade, tem parte em uma imagem e o complemento dela em outra) ou apenas imagens imediatamente vizinhas em que nenhuma estrutura de nematoide está presente. Essa junção com vizinhas imediatas se faz necessário para verificar a presença ou não de nematoides nas imagens. Para efetuar a junção das imagens o usuário deve executar o código presente no **apêndice 9**.

Para exemplificar essa junção apresentamos a **figura 2**. Após o escaneamento da lâmina (**figura 2.a**) são obtidas imagens separadas que, quando juntas, representam uma região (**figura 2.b**). Nessa ilustração reduzimos o número de imagens para apenas 24 para facilitar a compreensão, mas é importante ter em mente que, em cenários reais, milhares de imagens individuais são obtidas a partir do escaneamento de uma lâmina típica. A imagem individual que contém nematoide (imagens centrais dos quadrados azuis contendo estrutura curvilínea na **figura 2**) serve como pivô, ou seja, passa a ser o centro para seleção de imagens de junção. Nesse processo de junção, uma região que contenha nematoide terá no mínimo 9 imagens (o nematoide que serviu como pivô mais as oito vizinhas imediatas). Caso uma das vizinhas imediatas contenha uma estrutura considerada nematoide, o processo de junção continua e a imagem da junção terá mais de 9 imagens. Desta maneira é evitada a presença de nematoides nas bordas das imagens finais combinadas (as que sofreram junção), o que constituiria uma importante limitação para operações de medição e contagem. No exemplo da **figura 2** os quadrados (imagens) que sofrerão o processo de junção estão preenchidos em azul, e aquelas que não sofrerão junção estão representadas pelo preenchimento branco.

Posteriormente a aplicação deste algoritmo para definir quais imagens devem ser combinadas, a junção é efetuada por meio do plugin *Stitching* (Preibisch et al., 2009) do Fiji, executado através de macros automaticamente criadas no Python.

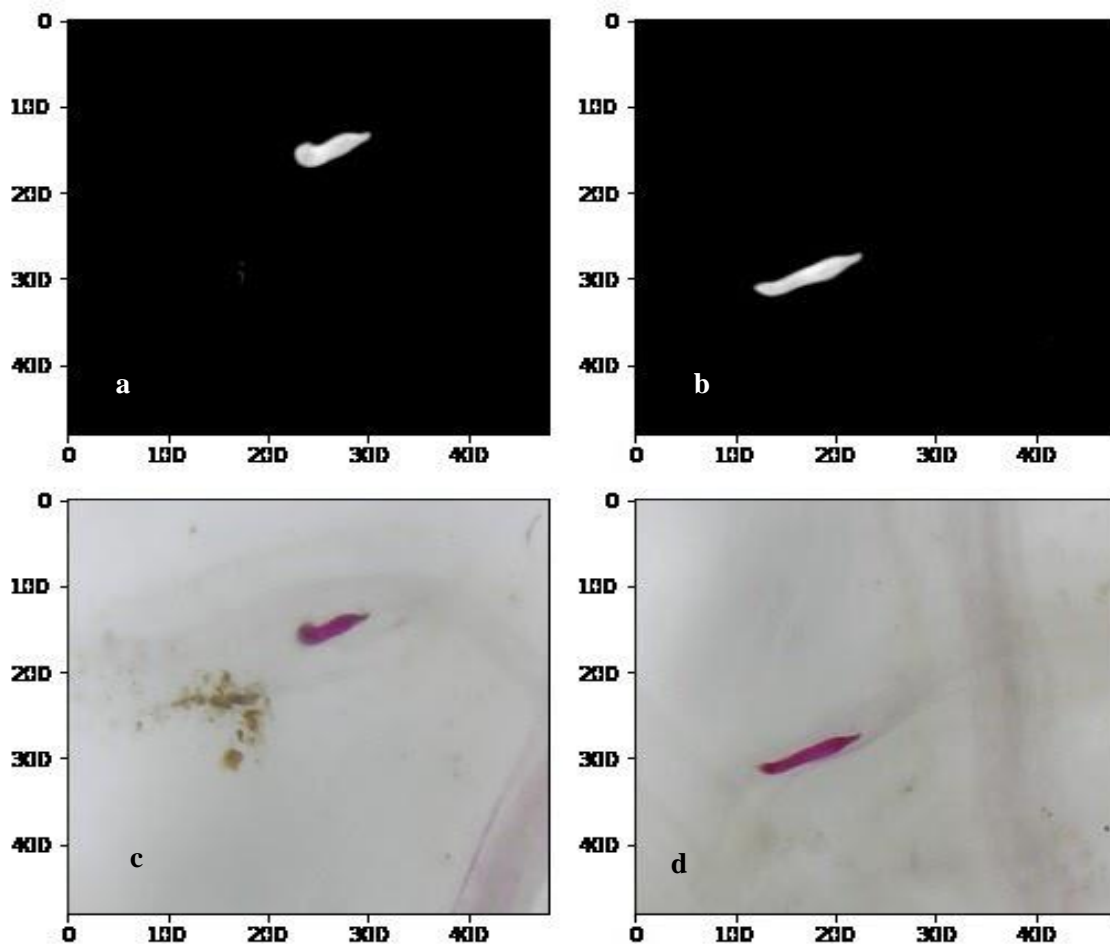


**Figura 2.** Exemplo de seleção de imagens para junção. Inicialmente uma lâmina é imageada inteiramente ou apenas uma região (**figura 2.a**). Posteriormente são selecionadas imagens que contém estruturas classificadas como nematoides (**figura 2.b**) e juntas com suas vizinhas imediatas. Caso alguma destas vizinhas contenha nematoide, o processo de seleção continua.

### 3. Resultados e Discussão

Após 17 ciclos de treinamento (como descrito na **seção 2.3.1**) foi salvo o modelo que teve o menor valor de RMSE (0,0109). O modelo com esse valor de perda resultou em acurácia de 0,9914. Embora extremamente elevado, é necessário interpretar esse valor com cautela. Como a maior parte da imagem de validação consiste de fundo (*background*), a predição por parte do modelo se torna facilitada pela menor complexidade da imagem e mesmo pela menor quantidade de pixels que podem ser classificados como o objeto de interesse (nesse caso nematoide). Estes fatores podem ter contribuído para este elevado valor de acurácia. De qualquer forma, conforme demonstrado nas imagens a seguir, a segmentação dos nematoides foi bastante precisa e atendeu aos objetivos desse trabalho.

Após a obtenção do modelo, a capacidade de predição foi testada em um novo conjunto, de teste (conjunto de imagens não utilizadas para o treinamento do modelo nem para o processo de validação), que continha 8 imagens. A **figura 3** apresenta alguns dos resultados destas predições.

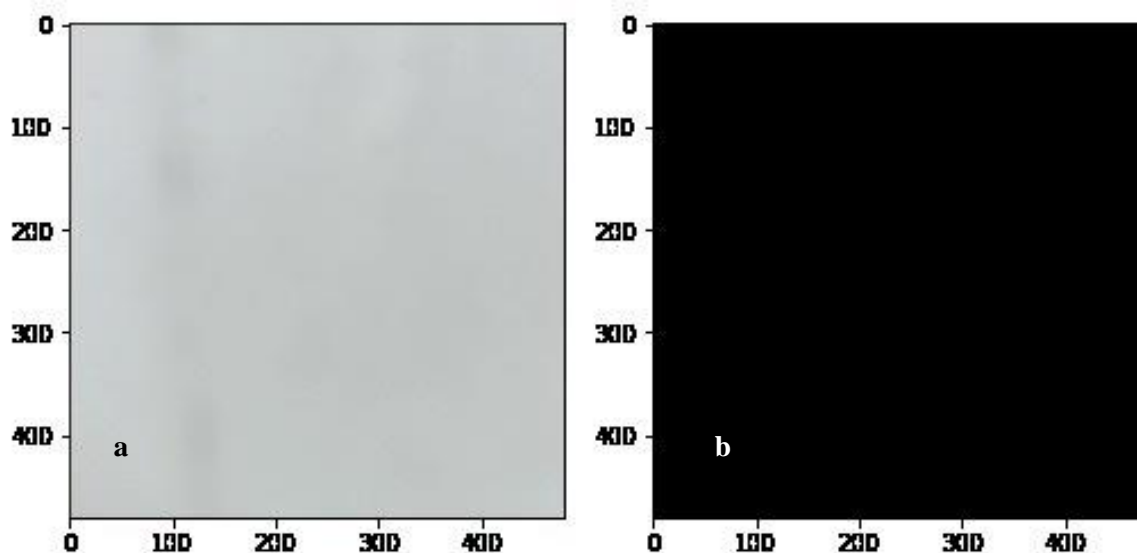


**Figura3.** Nematoides preditos (figuras 3.a e 3.b) das imagens originais de nematoides (figuras 3.c e 3.d).

Conforme visto na **figura 3**, a rede neural convolucional foi capaz de realizar segmentações de qualidade em situações onde o background estava mais homogêneo (predição **3.b** da imagem **3.d**) bem como quando artefatos (nesse caso fragmentos de solo) estavam presentes na imagem como na predição **3.a** da **figura 3.c**.

Invariavelmente, quando se escaneia uma lâmina na sua totalidade, são capturadas imagens em que nenhum objeto de interesse está contido (**figura 4.a**). Desta maneira se faz necessário que a rede neural convolucional que realiza as predições seja capaz de indicar corretamente que estas imagens não possuem nenhum nematoide. Como visto na máscara predita (**figura 4.b**), a rede que utilizamos foi capaz de prever corretamente que não haviam estruturas de nematoide naquela imagem.

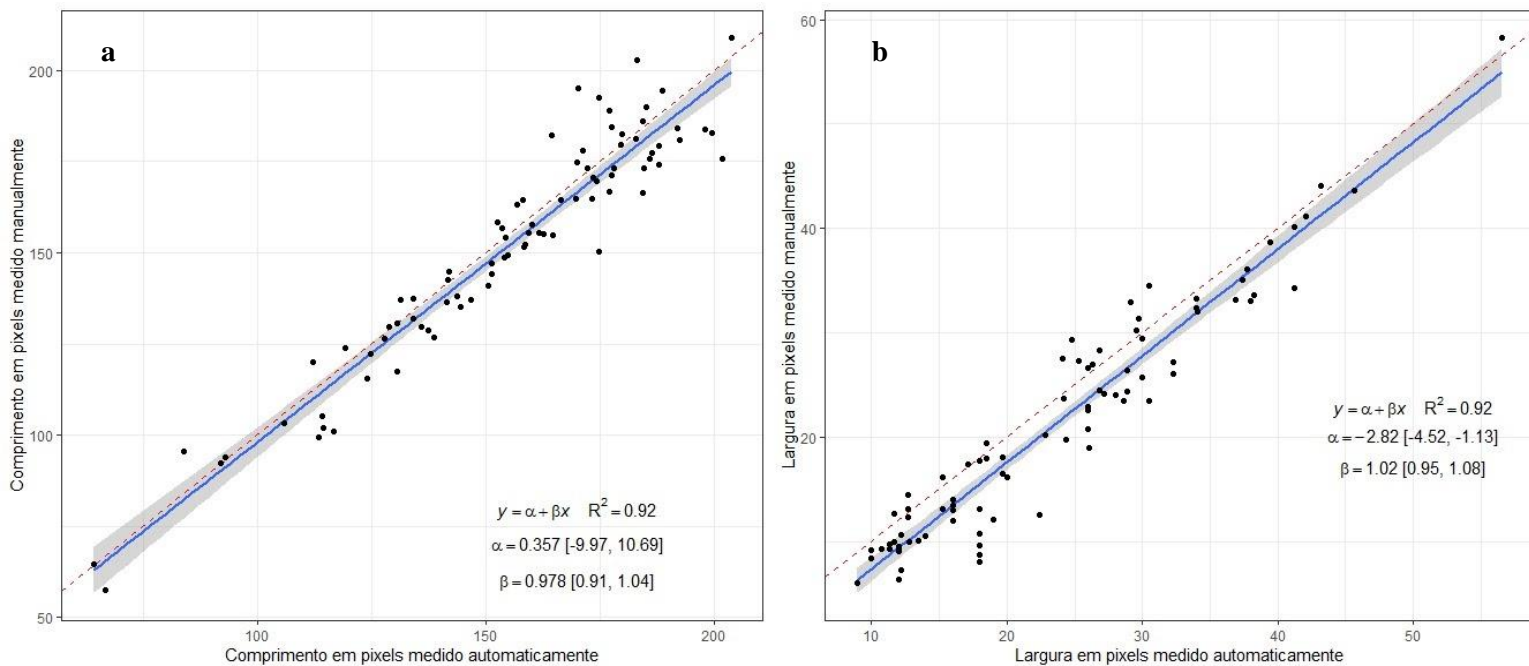




**Figura 4.** Exemplo de imagem (**figura 4.a**) que, invariavelmente, ocorrem no escaneamento e não contém nematoides. A **figura 4.b** traz a predição correta da ausência de nematoides na imagem avaliada.

Tipicamente a segmentação (separação da estrutura de interesse do restante da imagem) é o primeiro passo para a análise de estruturas biológicas quando se tenta estudar estes organismos utilizando visão computacional (Danielsen e Nordenfelt, 2017). Desta forma o nosso resultado de segmentação utilizando uma rede neural convolucional é um importante passo para o estudo dos nematoides fitoparasitas. Após a segmentação é possível realizar múltiplas análises sobre os objetos obtidos, como, por exemplo, verificar a acurácia do modelo quanto a detecção de estruturas. Esta análise pode ocorrer através da contagem das estruturas detectadas (segmentadas) pelo modelo comparada com a contagem de estruturas detectadas manualmente através da observação por parte do usuário, resultando em uma porcentagem de acurácia do modelo (Al-Kohafi et al., 2018). Caso o resultado desta comparação seja satisfatório, pode-se automatizar essa tarefa antes realizada de maneira manual. Desta maneira pode-se reduzir a variabilidade de análise conforme o observador que a realiza, bem como otimizar o tempo que o pesquisador dedica às atividades realizadas no laboratório. Estes objetos segmentados podem ser submetidos a outros tipos de análises, como, por exemplo, no artigo publicado por Fischer et al. (2020) que apresentaram uma ferramenta baseada em aprendizagem profunda capaz de medir o perímetro e a área de mitocôndrias. Neste artigo apresentamos o software NemaMeasure que realiza a medição do comprimento e largura de nematoides presentes em imagens.

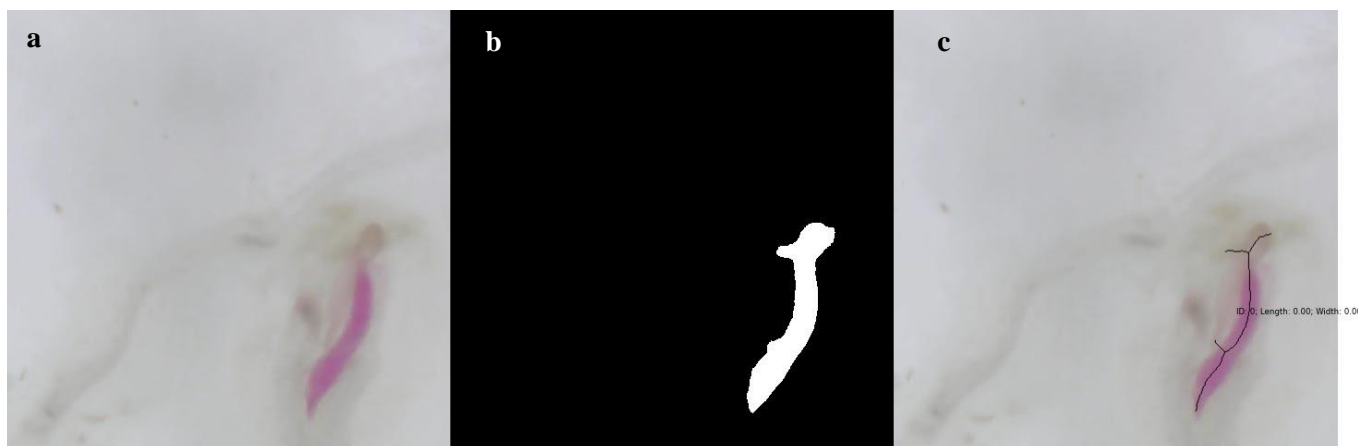
Inicialmente os nematóides presentes nas imagens foram medidos manualmente quanto à quantidade de pixels que os constituíam. Essas medições se deram nas imagens originais (antes de sofrerem binarização e esqueletização). Posteriormente os dados destas medições foram comparados com as medições automáticas realizadas através do software NemaMeasure. Os objetivos desta comparação foram verificar a eficiência do método automático, bem como verificar se havia necessidade de utilizar um fator de correção no seu uso para que os valores da análise automática fossem tão precisos quanto aqueles obtidos na análise manual (tida como a análise mais precisa). A **figura 5** traz o resultado da análise de regressão para um modelo linear que visou verificar a correlação entre as medições manuais e as medições automáticas de NemaMeasure.



**Figura 5.** Gráficos da análise de regressão implementada para verificar a consonância entre as medições manuais e automáticas. A **figura 6.a** traz a análise concernente ao comprimento e a **figura 6.b** a largura.

A **figura 5.a** traz a análise do comprimento e a **figura 5.b** a da largura dos nematoides em número de pixels. Conforme visto, para ambas as análises, o resultado do coeficiente de determinação ficou em torno de 0,92, indicando uma ótima acurácia do método automático, especialmente considerando que a medição manual está sujeita a variações.

Para a predição do comprimento dos nematoides, os parâmetros da equação de regressão não diferiram significativamente dos valores de uma linha 1:1. Isso significa que os valores obtidos automaticamente para o comprimento dos nematoides podem ser usados diretamente, sem necessidade de aplicar essa equação. Para a variável largura do nematoide (**figura 6.b**), o intervalo de confiança para o intercepto do modelo não admitiu o valor 0, indicando haver, portanto, um viés nos valores obtidos automaticamente, que pode ser corrigido pela aplicação da equação estimada  $y = \beta x$

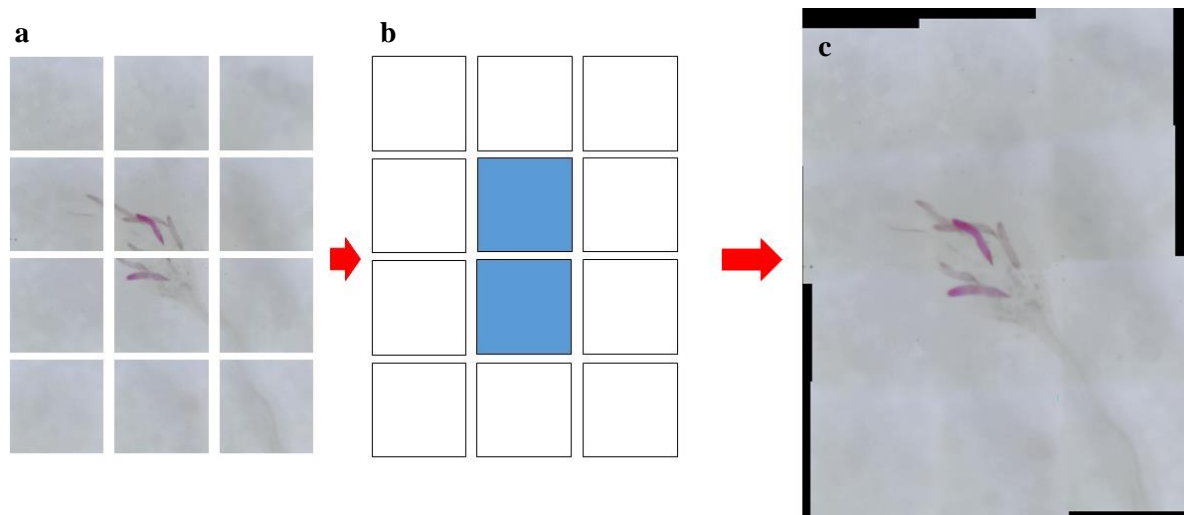


**Figura 6.** Exemplo de erro nas operações de medição. Por conta de uma região no topo do nematoide (**7.a**) ter sido erroneamente classificada como nematoide na máscara predita (**7.b**) o esqueleto apresentou bifurcação (**7.c**) impossibilitando que as medições fossem realizadas.

Os resultados das medições automáticas de largura e comprimento só foram possíveis em máscaras preditas (aquelas obtidas através do modelo) que não sofreram bifurcações. Nesse caso, por conta das ramificações, o algoritmo não foi capaz de prever valores de comprimento e largura. A **figura 6** traz um exemplo da imagem original (**6.a**), da máscara predita (**6.b**) e do esqueleto obtido (**6.c**), no qual não foi possível realizar as análises de comprimento e largura de maneira precisa por conta da bifurcação que o esqueleto calculado a partir da máscara predita apresentou. O mesmo problema pode ocorrer quando há nematoides sobrepostos na imagem.

Nossos resultados utilizando uma rede neural convolucional se soma a outros esforços no estudo de nematoides que utilizaram ferramentas de aprendizado de máquina. Chen et al. (2020) apresentaram um algoritmo de rede neural convolucional capaz de detectar nematoides presentes em imagens, separando-os do *background*, logo segmentando-os. Hakim et al. (2018) apresentaram resultado similar com um algoritmo de rede neural capaz de diferenciar o nematoide *Caenorhabditis elegans* de artefatos presentes na imagem. Além disso, o algoritmo foi capaz de diferenciar nematoides fêmeas de nematoides machos e realizar algumas medições como o comprimento. Utilizando algoritmos de visão computacional, é possível realizar outras análises além das citadas anteriormente quando se estuda nematoides. Akintayo et al. (2018) utilizando aprendizado profundo, apresentaram um algoritmo capaz de contar ovos depositados pelo nematoide parasita da soja, *Heterodera glycines*. Desta maneira essa tarefa de contagem pôde ser automatizada o que reduz o tempo para a análise, bem como elimina o problema de contagens diferentes entre si conforme mudança de observador que a realiza. Thevenoux et al. (2021) apresentaram, similarmente a este artigo, um algoritmo capaz de realizar medições em alguns órgãos de nematoides baseado na contagem de pixels.

A **figura 7** traz o processo de seleção de imagens e o resultado da junção descrita anteriormente na **seção 2.4** deste capítulo. Neste processo, imagens individuais (**figura 7.a**) são juntadas para criar uma imagem só que representa uma região contendo nematoide, ampliando o campo de visão sobre a amostra e permitindo que nematoides presentes em imagens contíguas estejam presentes em uma só imagem. Nesse caso como apenas duas imagens possuíam nematoides, a junção se deu em 12 imagens. Para que ocorra um encaixe satisfatório entre as imagens, é necessário aplicar sobreposição entre elas. No caso deste experimento utilizamos 20% de sobreposição.



**Figura 7.** Imagens individuais que contém nematoides (imagens centrais em 7.a e 7.b) são selecionadas e servem de “pivô”. A partir destas imagens suas vizinhas são selecionadas e juntas, até que nenhuma das vizinhas possuam nematoides, formando imagens compostas com maior amplitude de visão (**figura 7.c**).

Nesse experimento foi escaneada uma lâmina que continha fragmentos de raiz de soja parasitadas por nematoides. Ao fim do processo de escaneamento foram originadas 8.856 imagens. Após o processo de seleção de imagens que continham nematoides (realizado através do modelo criado pela UNet modificada descrita anteriormente) e do processo de junção, foram obtidas 153 imagens que representavam regiões que continham nematoides. Essa redução diminuiu consideravelmente o tempo e os recursos computacionais necessários para efetuar a análise.

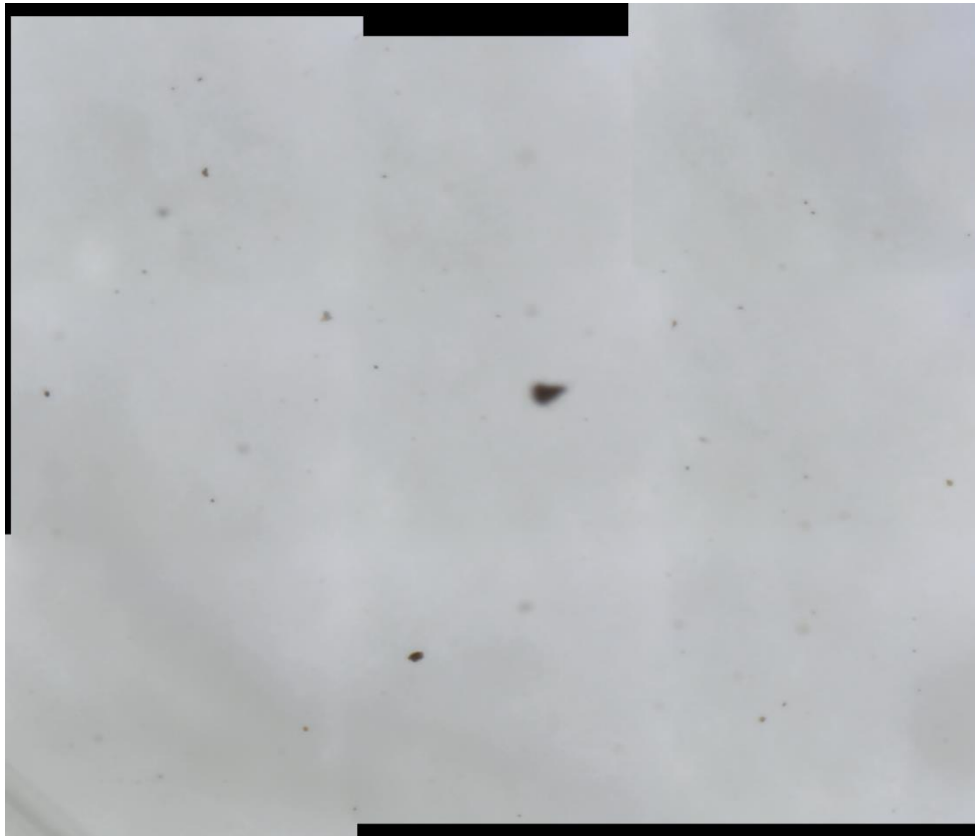
Acreditamos que a abordagem de junção de imagens que contem estruturas de interesse (no nosso caso nematoides) com as suas vizinhas imediatas para formar regiões que contém estes organismos pode facilitar o estudo de amostras biológicas. Inicialmente há uma redução na necessidade de capacidade de processamento computacional já que a junção se dará em um número menor de imagens, evitando, também, o armazenamento de dados supérfluos (por exemplo, imagens que contém apenas background). Outra vantagem é a otimização do tempo utilizado para as atividades do laboratório. Através dessa junção o número de imagens a serem avaliadas é diminuído drasticamente. Nesse trabalho, por exemplo, essa redução superou 90%.

Outra vantagem da abordagem aqui proposta é que, quando se utiliza o escaneamento completo de uma amostra acoplado a uma estratégia de junção de imagens, objetos que estavam “fragmentados” em mais de uma imagem podem ser reconstruídos. Por exemplo, quando se escaneia uma amostra contendo nematoides automaticamente, invariavelmente serão originadas imagens em que um nematoide (ou mais) está

parcialmente presente em duas ou mais imagens (como os nematoides da **figura 7**). Sem a adoção de uma estratégia de junção, não seria possível realizar medições neste nematoide já que ele estaria presente em mais de uma imagem, o que, na melhor das hipóteses, resultaria numa medição errada, por exemplo, do comprimento. Mesmo uma operação mais simples como a contagem estaria sujeita a erro já que o avaliador poderia contar o mesmo nematoide mais de uma vez.

Acreditamos que a abordagem de escaneamento total com posterior seleção de imagens para obtenção de imagens amplas que representem regiões que contem nematoides, apresenta vantagens em relação à microscopia convencional. Na abordagem convencional (sem escaneamento digital) o pesquisador deve percorrer toda a lâmina manualmente e, quando encontra o nematoide, capturar uma imagem deste organismo, ou apenas contá-lo. Desta forma o processo é dispendioso em termos de tempo (já que é necessário percorrer toda a lâmina na procura pelo organismo), bem como exige elevada atenção do usuário para evitar capturar/contar mais de uma vez o mesmo nematoide. Na abordagem proposta neste artigo o risco de imagens duplicadas do mesmo nematoide inexistente já que, mesmo nematoides presentes em mais de uma imagem, passam a formar uma imagem só (a imagem de região contendo nematoide). Outra vantagem é que, através do escaneamento automático digital, o usuário necessita apenas colocar a lâmina e especificar a movimentação do estágio automatizado. Desta maneira pode dedicar o tempo que utilizava para o escaneamento manual para outras tarefas laboratoriais.

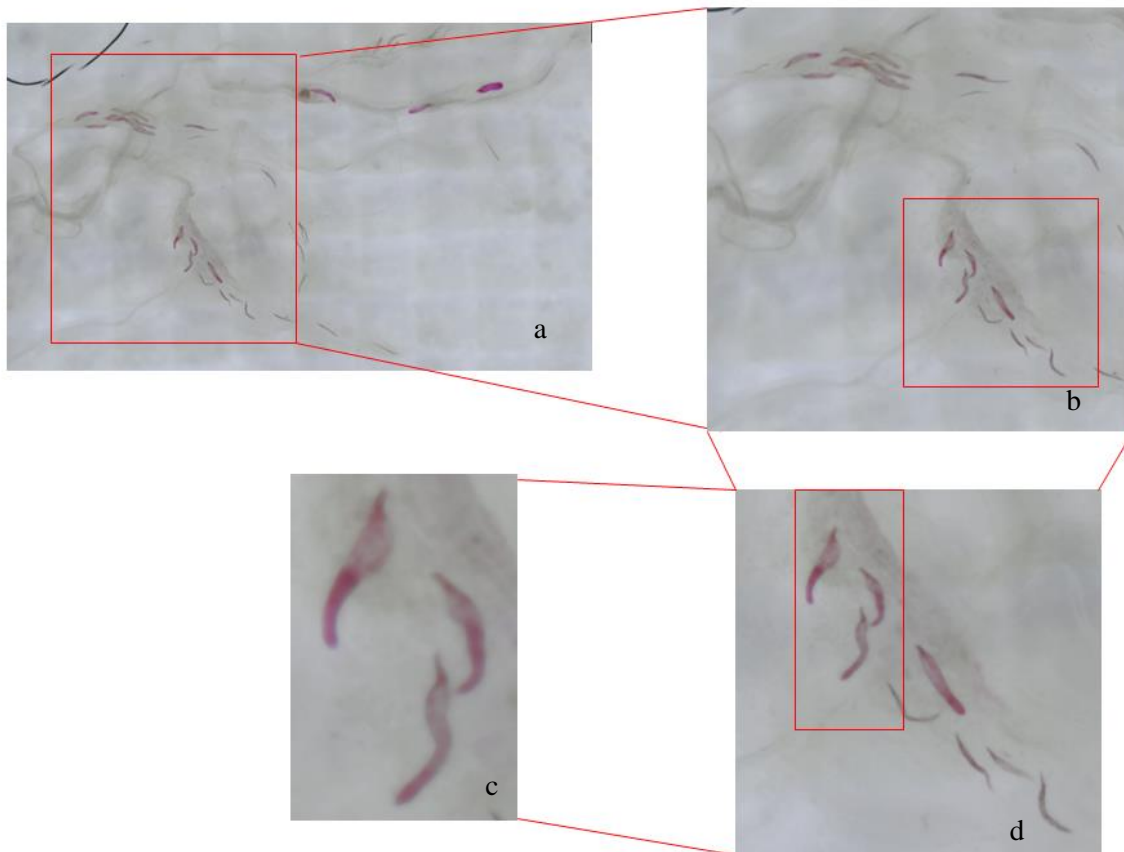
Há de se ressaltar que, dentre as imagens que sofreram junção, algumas delas não possuíam estruturas de nematoide como exemplificado na **figura 8**. Desta maneira, se faz necessário que o usuário realize uma avaliação de cada uma destas imagens para verificar se elas de fato contêm nematoides. Mesmo com esta limitação, ocorreu uma redução de cerca de 98% no número de imagens que o observador tem que avaliar para verificar a presença destes organismos.



**Figura 8.** Exemplo de imagem ampla que erroneamente sofreu o processo de junção.

Esses falsos positivos (imagens montadas sem nematoides) podem ser mitigados pelo re-treinamento da rede neural convolucional, ou pela utilização de outra arquitetura que pode levar a obtenção de resultados superiores (BRESSEM et al., 2020). Entretanto, consideramos desnecessária essa correção porque o impacto dos falsos positivos na qualidade da análise é negligível.

Na microscopia tradicional apenas uma imagem pode ser avaliada em cada momento. Com o uso de um estágio automatizado (que proporciona a captação de imagens que representam a lâmina de microscopia ou a região por inteiro) e uma estratégia de junção de imagens, este problema é superado já que a imagem (que sofreu junção) é composta por múltiplas imagens, proporcionando que todas estas imagens (outrora individuais) sejam analisadas em um só momento. Além disso, esta ampliação pode proporcionar um melhor entendimento das relações espaciais que pode ser de importância para o entendimento de um determinado fenômeno ocorrente na amostra. A **figura 9** traz um exemplo intuitivo desta ampliação no campo de visão. Originalmente uma só imagem contém todas as imagens que sofreram junção. Caso o usuário deseje mais detalhamento basta realizar zoom com o visualizador de fotos (sem que se perca qualidade). Como todas as informações contidas na imagem composta originalmente foram captadas em imagens individuais, não há perda de informação proporcionando que operações, como por exemplo medições, sejam realizadas de maneira eficiente.



**Figura 9.** Na figura 9.a a imagem que sofreu junção é exemplificada. Regiões mais específicas podem ser avaliadas pelo observador através do zoom (9.b, 9.c e 9.d).

Embora promissores, os resultados de experimentos que envolvem ferramentas de visão computacional e aprendizado de máquina no estudo de nematoides, devem ser interpretados com cautela. Como em outras áreas da biologia em que essas ferramentas já foram utilizadas, há uma preocupação que os modelos obtidos através de aprendizado de máquina não performem tão bem quando são submetidos a imagens que sofreram processo de captação diferente (Ferentinos, 2018, Ramcharan et al., 2019). Outro ponto importante é que a maioria dos estudos que envolvem nematoides e aprendizado de máquina são focados na espécie *Caenorhabditis elegans*, tida como uma espécie modelo, o que lança dúvidas se estes modelos conseguiriam performar satisfatoriamente na análise de outras espécies de nematoides como *Meloidogyne javanica*, que estudamos neste artigo. Outro ponto à se ressaltar é a ausência de estudos que identifiquem nematoides em raízes utilizando técnicas de visão computacional, tornando este experimento um pioneiro nesta abordagem.

A comunidade científica que atua no campo da biologia ainda não possui, de maneira geral, familiaridade com linguagens de programação e algoritmos como os de aprendizagem profunda. Desta maneira a adoção dessas técnicas ainda é incipiente (Zhang et al., 2021). Outro problema que pode limitar o uso destas técnicas é o custo para obtenção e uso de tecnologias que utilizem aprendizado de máquina, já que podem envolver o uso de equipamentos de elevado custo, seja na captação das imagens (necessidade de microscópio de alto custo e de câmera acoplada) ou mesmo no treinamento dos modelos de aprendizado de máquina (custo com computadores com elevada capacidade de processamento e unidade de processamento gráfico). Acreditamos

que nossa abordagem apresenta a vantagem de ser uma tecnologia de baixo custo, o que a torna mais acessível à comunidade do que outras abordagens que tradicionalmente dependem da utilização de equipamentos dispendiosos do ponto de vista financeiro, já que utilizamos em todas as etapas desta metodologia material de acesso livre e/ou de baixo custo. Além disso, todos os códigos, quer sejam para os softwares ou para os hardwares, são de acesso livre. Desta maneira o usuário pode adaptá-lo conforme suas necessidades (para mais informações sobre o software de controle de NemaStage e os custos deste equipamento, acessar o **capítulo I** desta dissertação). Outro ponto que pode ser interessante ao se utilizar NemaStage é o possível escaneamento simultâneo de placas de microscopia de dimensões menores (como por exemplo as utilizadas no estudo de fungos micorrízicos arbusculares). Para confirmar tal uso possível, serão necessários futuros testes.

#### **4. Conclusões**

Neste artigo foi demonstrada a capacidade de um modelo obtido através do treinamento de uma rede neural convolucional (implementada através do pacote Keras utilizando como linguagem de programação Python), nomeadamente uma Unet, em segmentar e identificar nematoides contidos em imagens. Assim sendo, este modelo foi utilizado para segmentar nematoides que posteriormente foram medidos quanto à largura e o comprimento através do software NemaMeasure. Este software foi tão preciso quanto a medição manual em número de pixels que constituíam os nematoides para a variável comprimento.

Adicionalmente, o modelo foi utilizado para classificar imagens que continham ou não nematoides, obtidas através do uso de NemaStage (que se mostrou capaz de captar imagens de qualidade) controlado por StageControl (que foi eficiente no controle de NemaMeasure). Estas imagens foram juntas de maneira satisfatória com suas vizinhas imediatas, em um processo que cessou quando estas vizinhas não possuíam mais nematoides. Esta estratégia de junção resultou em regiões contendo nematoides. Essa abordagem de trabalhar com regiões que contém o objeto de interesse no lugar de todas as imagens obtidas no escaneamento, reduziu significativamente as imagens a serem avaliadas manualmente pelo usuário. Além disso, esta abordagem proporcionou a ampliação do campo de visão em relação à amostra.

De maneira geral apresentamos neste artigo uma abordagem de baixo custo, baseada em ferramentas de visão computacional e dentro do escopo da ciência aberta, de baixa necessidade de processamento computacional e de código aberto que pode ser utilizada para o estudo de nematoides parasitas de raízes de maneira eficiente.



## Referências

- Abdelday, E.A., Erriquens, F., Sasanelli, N., Ceglie, F.G., Zaccone, C., Miano, T., Coccozza, C. 2014. Effects of several amendments on organic melon growth and production, *Meloidogyne incognita* population and soil properties. **Scientia Horticulturae**, v. 180, p. 156–160. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.scienta.2014.10.032>>.
- Akintayo A, Tylka GL, Singh, AK SINGH, Ganapathysubramanian B, Singh A, Sarkar S. 2018. A deep learning framework to discern and count microscopic nematode eggs. **Scientific Reports**, v. 8, n. 1, p. 1–11.
- Al-Kohafi, Y., Zaltsman, A., Graves, R., Marshall, W., Rusu, M. 2018. A deep learning-based algorithm for 2-D cell segmentation in microscopy images. **BMC Bioinformatics**, v. 19, n. 1, p. 1–11.
- Bressem, K.K., Adams, L.C., Erxleben, C., Hamm, B., Niehues, S.M., Vahldiek, J.L. 2020. Comparing different deep learning architectures for classification of chest radiographs. **Scientific Reports**, v. 10, n. 1, p. 1–16. Nature Publishing Group UK. Disponível em: <<https://doi.org/10.1038/s41598-020-70479-z>>.
- Campbell, R.A.A., Eifert, R.W., Turner, G.C. 2014. Openstage: A low-cost motorized microscope stage with sub-micron positioning accuracy. **PLoS ONE**, v. 9, n. 2.
- Cascione, A., Bicalho, G., Aleandro, S., Cristina, A., Machado, Z. 2021. Control of *Meloidogyne paranaensis* mediated by silicon. **Scientia Agricola**, v.78, n.3.
- Chen, L., Strauch, M., Daub, M., Jiang, X., Jansen, M., Luigs, H., Schultz-Kuhlmann, S., Krüssel, S., Merhof, D. 2020 . A CNN FRAMEWORK BASED ON LINE ANNOTATIONS FOR DETECTING NEMATODES IN MICROSCOPIC IMAGES. IEEE 17<sup>th</sup> International Symposium on Biomedical Imaging (ISBI), p. 508–512.
- Collins, J.T., Knapper, J., Stirling, J., Mduda, J., Mkindi, C., Mayagaya, V., Mwakajinga, G.A., Nyakyi, P.A., Sanga, V.L., Carbery, D., White, L., Dale, S., Lim, Z.J., Baumberg, J.J., Cicuta, P., McDermott, S., Vodenicharski, B., Bowman, R. 2020. Robotic microscopy for everyone: the OpenFlexure microscope. **Biomedical Optics Express**, v. 11, n. 5, p. 2447.
- Conduto, N.S., Silva, M.T.R., Rinaldi, L.K., Dias-Arieira, C.R. 2020. Interaction between resistance inducer and micronutrients on the control of root-lesion nematode and the development of soybean plants. **Revista Caatinga**, v. 33, n. 3, p. 591–598.
- Danielse, J., Nordenfelt, P. 2017. Computer vision-based image analysis of bacteria. **Methods in Molecular Biology**, v. 1535, p. 161–172.
- Ferentinos, K.P. 2018. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, v. 145, n. January, p. 311–318. Elsevier. Disponível em: <<https://doi.org/10.1016/j.compag.2018.01.009>>.
- Fischer, C.A., Besora-Casals, L., Rolland, S.G., Duchon, M., Conradt, B., Marr, C.

2020. MitoSegNet: Easy-to-use Deep Learning Segmentation for Analyzing Mitochondrial Morphology. **iScience**, v. 23, n. 10.
- Gomes, V.M., Souza, R.M., Mussi-Dias, V., Silveira, S.F., Dolinski, C. 2011. Guava decline: A complex disease involving *Meloidogyne mayaguensis* and *Fusarium solani*. **Journal of Phytopathology**, v. 159, n. 1, p. 45–50.
- Hakim, A., Mor, Y., Toker, I.A., Levine, A., Neuhof, M., Markovitz, Y., Rechavi, O. 2018. WorMachine: Machine learning-based phenotypic analysis tool for worms. **BMC Biology**, v. 16, n. 1, p. 1–11.
- Kapagianni, P.D., Boutsis, G., Argyropoulou, M.D., Papatheodorou, E.M., Stamou, G.P. 2010. The network of interactions among soil quality variables and nematodes: short-term responses to disturbances induced by chemical and organic disinfection. **Applied Soil Ecology**, v. 44, n. 1, p. 67–7.
- Kirwa, H.K., Murungi, L.K., Beck, J.J., Torto, B. 2018. Elicitation of Differential Responses in the Root-Knot Nematode *Meloidogyne incognita* to Tomato Root Exudate Cytokinin, Flavonoids, and Alkaloids. **Journal of Agricultural and Food Chemistry**, v. 66, n. 43, p. 11291–11300.
- Korfhage, N., Mühlhng, M., Ringshandl, S., Becker, A., Schmeck, B., Freisleben, B. 2020. Detection and segmentation of morphologically complex eukaryotic cells in fluorescence microscopy images via feature pyramid fusion. **PLoS Computational Biology**, v. 16, n. 9, p. 1–16.
- Liu, S., Kandath, P.K., Warren, S.D., Yeckel, G., Heinz, R., Alden, J., Yang, C., El-Mellouki, T., Juvale, P.S., Hill, J., Baum, T.J., Cianzio, S., Whitham, S.A., Korkin, D., Mitchum, M.G., Meksem, K. 2012. A soybean cyst nematode resistance gene points to a new mechanism of plant resistance to pathogens. **Nature**, v. 492, n. 7428, p. 256–260. Nature Publishing Group. Disponível em: <<http://dx.doi.org/10.1038/nature11651>>.
- Mesa-Valle, C.M., Garrido-Cardenas, J.A., Cebrian-Carmona, J., Talavera, M., Manzano=Agugliaro, F. 2020. Global research on plant nematodes. **Agronomy**, v. 10, n. 8.
- Moura, G.S., Franzener, G. 2017. Biodiversity of nematodes biological indicators of soil quality in the agroecosystems. **Arquivos do Instituto Biológico**, v. 84, n. 0, p. 1–8.
- Ngalam, B., Mariette, N., Ianszen, M., Dewaegeneire, P., Denis, M., Porte, C., Piriou, C., Robilliard, E., Couteil, A., Nguema-Ona, E., Yvin, J., Gobert, V., Beury, A., Le Roux, A., Montarry, J., Fournet, S. 2021. Hatching Induction of Cyst Nematodes in Bare Soils Drenched With Root Exudates Under Controlled Conditions. **Frontiers in Plant Science**, v. 11, n. January, p. 1–14.
- Preibisch, S., Saalfeld, S., Tomancak, P. 2009. Globally optimal stitching of tiled 3D microscopic image acquisitions. **Bioinformatics**, v. 25, n. 11, p. 1463–1465.
- Rahman, S.A.S.A., Mohamed, Z., Othman, R.Y., Swennen, R., Panis, B., De Waele, D.,

- Remy, S., Carpentier, S.C. 2010. In planta PCR-based detection of early infection of plant-parasitic nematodes in the roots: A step towards the understanding of infection and plant defence. **European Journal of Plant Pathology**, v. 128, n. 3, p. 343–35.
- Ramcharan, A., McCloskey, P., Baranowski, K., Mbilinyi, N., Mrisho, L., Ndalaha, M., Legg, J., Hughes, D.P. 2019. A mobile-based deep learning model for cassava disease diagnosis. **Frontiers in Plant Science**, v. 10, n. March, p. 1–8.
- Ronneberger, O., Fischer, P.; Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: **International Conference on Medical Image Computing and Computer-Assisted Intervention**, p. 234-241, Springer.
- RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, URL: <http://www.rstudio.com>.
- Sasanenelli, N., Konrat, A., Mingunova, V., Toderas, I., Iurcu-Straistaru, E., Rusu, S., Bivol, A., Andoni, C., Veronico, P. 2021. Review on control methods against plant parasitic nematodes applied in southern member states (C zone) of the European Union. **Agriculture (Switzerland)**, v. 11, n. 7.
- Seid, A., Fininsa, C., Mekete, T., Decraemer, W., Wesemael, W.M.L. 2015. Tomato (*Solanum lycopersicum*) and root-knot nematodes (*Meloidogyne* spp.)-a century-old battle. **Nematology**, v. 17, n. 9, p. 995–1009.
- Skantar, A.M., Carta, L.K., Handoo, Z.A. 2008. Molecular and Morphological Characterization of an Unusual *Meloidogyne arenaria* Population from Traveler's Tree, *Ravenala madagascariensis*. **Journal of Nematology**, v.40, n.3, p. 179-189.
- Shorten, C., Khoshgoftaar, T.M. 2019. A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data**, v. 6, n. 1. Springer International Publishing. Disponível em: <<https://doi.org/10.1186/s40537-019-0197-0>>.
- Thevenoux R, Le VL, Villessèche H, Buisson A, Burton-Aimar M, Folcher L, Parisey N. 2021. Image based species identification of Globodera quarantine nematodes using computer vision and deep learning. **Computers and Electronics in Agriculture**, v. 186, n. May. VAN BEZOOIJEN, J.; Methods and Techniques for Nematology. **Wageningen university**, p. 102, 2006.
- Weber, F., Weber, V.A. M., Menezes G.V., Oliveira Júnior, A.S., Alves, D.A., Oliveira, M.V.M., Matsubara, E.T., Pistori, H., Abreu, U.G. 2020. Recognition of Pantaneira cattle breed using computer vision and convolutional neural networks. **Computers and Electronics in Agriculture**, v. 175, n. May, p. 105548. Elsevier. Disponível em: <<https://doi.org/10.1016/j.compag.2020.105548>>.
- Zinger, F.D., Zinger, L.K.C.R., Moraes, W.B., Camara, G.D.R., Alves, F.R. 2021. Quantification of damage and yield losses and management of root-knot nematodes in conilon coffee. **Revista Caatinga**, v. 34, n. 2, p. 287–297.

### ARTIGO III

#### **Uso de aprendizado de máquina para segmentação de hifas de fungo micorrízico arbuscular em fragmentos de raízes**

##### **Abstract**

O estudo de fungos micorrízicos arbusculares (FMA) está em constante evolução no que concerne às metodologias utilizadas. Recentemente, como no restante da biologia, os pesquisadores tentam aplicar à análise destes organismos estratégias que envolvam aprendizado de máquina. Desta maneira, tarefas podem ser automatizadas, o que pode levar à redução no tempo gasto para estas análises e o aumento da precisão quando comparada a abordagens tradicionais. Uma das aplicações práticas que pode ser utilizada para o estudo dos FMAs é a detecção de estruturas destes organismos em raízes micorrizadas. Visando detectar hifas de fungos micorrízicos arbusculares colonizando raízes de milho, utilizamos uma rede neural convolucional do tipo UNet, junto a uma estratégia de aumento de dados, para obtenção de um modelo que fosse capaz de detectar a presença destas estruturas. O desempenho do modelo obtido neste trabalho foi bastante satisfatório, não apenas nas imagens de treinamento e validação, mas também no conjunto de imagens de teste, com alta correspondência entre a segmentação feita manualmente e a obtida através do modelo. O resultado aqui apresentado pode subsidiar aplicações mais complexas como, por exemplo, a automatização da determinação do percentual e intensidade de colonização de FMA em raízes micorrizadas.

**Palavras-chave:** Fungos, aprendizagem profunda, UNet, hifas, Novas Técnicas

O artigo científico se encontra nas normas da revista “Mycologia” e está sujeito a alterações.

## 1. Introdução

Fungos micorrízicos arbusculares (FMAs) constituem um importante grupo de microrganismos do solo. Esses organismos estabelecem simbiose com cerca de 85% das espécies de angiosperma, sendo o grupo dominante de micorriza em termos de estabelecimento de simbiose com espécies de planta (Wang e Qiu, 2006). Uma série de efeitos benéficos é relatada na literatura como resultante da simbiose entre fungos micorrízicos arbusculares e plantas. São exemplos destes benefícios o melhor desenvolvimento de mudas, redução do tempo necessário para o transplântio em mudas de espécies nativas, aumento da qualidade físico-química de frutos, auxílio na absorção de fósforo, aumento da produção de matéria seca e aumento da capacidade da planta em lidar com condições de estresse hídrico e osmótico (Abreu et al. 2018; Bressan et al. 2001; Cordeiro et al. 2019; Rosseto et al. 2020; Honorato et al., 2020).

Do ponto de vista do funcionamento mais saudável do solo, os FMAs podem apresentar papéis de fundamental importância. Em solos contaminados por metais pesados como cádmio, mercúrio e chumbo, esses fungos podem acumular estes metais em uma estrutura chamada vesícula evitando que estes componentes potencialmente tóxicos se acumulem nos tecidos vegetais (Bano e Ashfaq, 2013). Outro efeito positivo dos FMAs no combate à poluição do solo se dá através da formação de complexos glicoproteína-metais, evitando que esses poluentes estejam disponíveis para absorção por parte da planta (Dhalaria et al. 2020). Outra importante ação destes organismos no solo se dá através da formação de redes comuns de micorrizas. Nesse processo diferentes espécies de plantas são ligadas por simbioses em comuns (os fungos micorrízicos arbusculares). São exemplos de benefícios oriundos desse processo a transferência de nutrientes e água entre plantas (Egerton-Warbuton et al. 2007; Ingrassia et al. 2019).

Todos esses benefícios para o solo e plantas justificam o interesse da comunidade científica no estudo desses organismos.

A análise destes organismos muitas vezes ocorre através do estudo de fragmentos de raiz, de maneira que esta deve ser retirada do solo, constituindo um processo destrutivo para a amostra. Outra abordagem pode ser o uso de análises que objetivam verificar o efeito da simbiose sob a planta, como por exemplo através da pesagem da massa seca da planta que recebeu tratamento micorrizado (Feitosa e Santos, 2017; Melloni et al. 2020).

Na biologia recentemente vem crescendo o uso de ferramentas de aprendizado de máquina nas rotinas de análise de áreas como genômica, proteômica, fitopatologia e nematologia (Ferentinos, 2018; Lahmer et al. 2020; Meyer, 2021; Thevenoux et al.,2021). De maneira resumida, estes algoritmos de aprendizado de máquina visam treinar a máquina através do reconhecimento e aprendizado de padrões contidos nos dados a realizar determinada análise, como, por exemplo, a detecção de uma determinada estrutura de interesse como ovos de nematóides presentes em uma amostra (Akintayo et al. 2018).

Essas ferramentas também podem ser utilizadas no estudo de FMAs. Melo et al. (2019) apresentaram um algoritmo de rede neural artificial capaz de realizar a distinção entre esporos e outras estruturas contidas em imagens, com posterior contagem. Evangelisti et al. (2021) publicaram uma metodologia de identificação e quantificação da colonização de fungos micorrízicos arbusculares em raízes de algumas espécies baseado no uso de uma rede neural convolucional, que é um tipo de algoritmo de aprendizado de máquina.

Neste artigo visamos verificar a capacidade de uma UNet (Ronneberger et al. 2015) modificada (uma arquitetura de rede neural convolucional) segmentar hifas de fungos micorrízicos arbusculares em raízes de milho (*Pennisetum glaucum*).

## 2. Metodologia

### 2.1 Obtenção de raízes micorrizadas

Plantas de milho, *Pennisetum glaucum* (L.) R.Br., serviram como plantas armadilhas para obtenção de raízes micorrizadas. As informações concernentes às características edafoclimáticas da região em que o solo foi coletado, as coordenadas geográficas e a análise química do solo onde as plantas foram cultivadas estão contidas em Vieira (2021).

Para a obtenção de mudas de milho, inicialmente sementes foram imersas em álcool 70% por 30 segundos para a desinfestação superficial. Posteriormente as sementes foram lavadas em água destilada autoclavada. Após esse processo de desinfestação as sementes foram colocadas em bandejas contendo vermiculita esterilizada. Quando as mudas atingiram 5 centímetros, 10 mudas foram transplantadas para cada um dos microcosmos. Os microcosmos consistiram de vasos preenchidos com 3 litros de solo (para mais especificações consultar Vieira (2021)).

Em cada um dos transplantes, cada microcosmo foi inoculado com 100 esporos das seguintes espécies de FMAs: *Acaulospora colombiana* (Espanha & N.C. Schenck) Kaonongbua, J.B. Morton & Bever (isolado SCT115A), *Gigaspora albida* I.R. Hall & L.K. Abbott (isolado MGR521C) and *Claroideoglossum etunicatum* (W.N. Becker &

Gerdemann) C. 39 Walker & A. Schüßler (isolado SCT101A). Os isolados de fungos foram obtidos da Coleção Internacional de Culturas de Glomeromycota (CICG-<https://www.furb.br/cicg>, Blumenau, Brasil).

As plantas foram mantidas em casa de vegetação com temperatura média de 23,4°C e 65% de humidade. Os microcosmos foram irrigados com água destilada autoclavada e 100 mL de solução nutritiva de Hoagland e Arnon (1950) a partir de 60 dias após o início do experimento. 120 dias após o transplante, as plantas de milho foram retiradas dos microcosmos, as raízes lavadas em água corrente, secadas e posteriormente armazenadas à 4°C.

Três gramas de raízes frescas foram coletadas das plantas de milho, de cada microcosmo, e colorida de acordo com a metodologia descrita por Koske e Gemma (1989). Fragmentos de raiz de cerca de 1 centímetro foram selecionadas para serem colocadas em lâminas que passaram por posterior análise em microscópio. Dessa forma, os fragmentos de raízes foram postos em lâminas que continham PLVG (polivinil-ácido láctico-glicerol). A colonização de FMAs nas raízes foi considerada positiva quando pelo menos uma estrutura (hifa, vesícula ou arbúsculo) desse grupo de fungos estava presente nas raízes.

## **2.2 Obtenção das imagens de hifas de FMA**

As lâminas de microscopia que continham raízes micorrizadas, adquiridas conforme a metodologia descrita na **seção 2.1** deste capítulo, foram submetidas a análise



em microscópio. O microscópio utilizado foi um Nikon. As imagens foram obtidas através do acoplamento de uma câmera digital da marca Cannon a este microscópio.

Os fragmentos de raízes foram observados e algumas regiões que continham hifas de fungo micorrízico arbuscular foram escolhidas e imagens destas estruturas foram capturadas.

### **2.3 Rede Neural Convolutacional para extração de estruturas de FMA**

Para a segmentação de hifas de fungos micorrízicos arbusculares presentes nas imagens de raízes de milho foi utilizada uma rede neural convolutacional tendo como arquitetura uma UNet (Ronneberger et al. 2015). Esta rede sofreu algumas modificações como por exemplo nos valores de entrada e das camadas de convolução e de agrupamento. A arquitetura da rede pode ser acessada no **apêndice 11**.

Para essa rede neural foram utilizadas 28 imagens na matriz de treinamento e 11 imagens na matriz de validação. Ambas as matrizes continham máscaras (imagens binárias em que apenas o objeto de interesse está destacado em branco e todo o restante da imagem assume coloração preta) associadas para que o algoritmo aprendesse quais características eram importantes na definição do objeto de interesse (nesse caso hifas). Para aumentar a variabilidade e o número de dados para treinamento, foi utilizada a técnica de aumento de dados.

Foram utilizadas como entradas imagens em escala de cinza de dimensões 448 x 448 pixels. Ao todo a rede consistiu de 4.751.153 parâmetros treináveis. No treinamento a rede extrai informações importantes para o entendimento do contexto da imagem de

forma que aprenda quais são as características importantes para a definição dos objetos de interesse ali contidos (as hifas). O entendimento destas características, no caso deste experimento, se dá através da comparação da imagem original e as máscaras que contém as regiões de interesse marcadas. A matriz de validação diz respeito a imagens e suas respectivas máscaras que servem para o treinamento da capacidade de predição do modelo obtido através do processo de aprendizado que ocorre na matriz de treinamento. Ou seja, as máscaras criadas utilizando o modelo são comparadas com as máscaras confeccionadas manualmente na matriz de validação. As máscaras da matriz de treinamento e de validação foram obtidas através do uso do software Apeer da Carl Zeiss.

Como métrica de precisão do modelo foi utilizada a RMSE (equação abaixo). Esse valor diz respeito à diferença entre a predição realizada pelo modelo e os valores que representam as máscaras manuais criadas pelo usuário para a matriz de validação. O modelo que resultou no menor valor deste parâmetro foi salvo e utilizado posteriormente para a obtenção das máscaras preditas.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Predito_i - Observado_i)^2}{N}}$$

O último passo para a avaliação do modelo foi testá-lo em um conjunto de dados separado, ao qual a rede neural não havia sido exposta previamente, seja no processo de treinamento ou no processo de validação. Essa matriz de teste consistiu de 14 imagens para as quais o modelo deveria gerar uma imagem binária contendo somente as hifas (máscara predita).

Para o treinamento a rede foi desenhada para ter 50 ciclos de treinamento (*epochs*) e valor de *patience* 3 (esse valor significa que, se após três ciclos de treinamento o modelo

não sofreu redução de RMSE o treinamento deve parar e o modelo que obteve o menor valor desta métrica deve ser salvo).

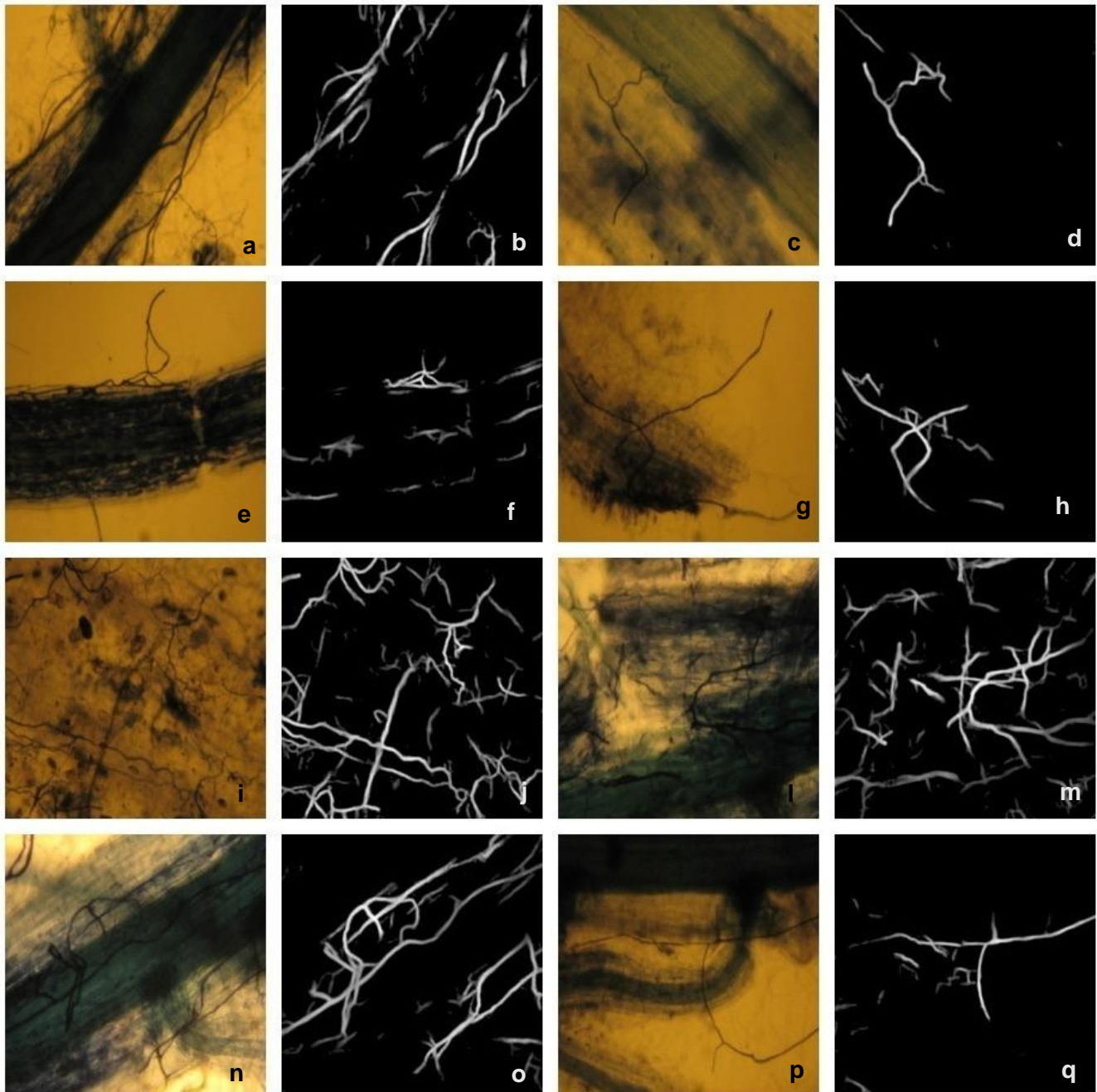
### 3. Resultados e Discussão

O menor valor de RMSE obtido foi 0,06877 e acurácia de 0.9778, logo, o modelo que resultou nestes valores foi salvo e utilizado posteriormente para a predição (obtenção de imagens de hifas segmentadas). A **figura 1** traz exemplos das predições que o modelo realizou na identificação de hifas presentes em imagens de raízes de milho colonizadas por FMAs.

Como percebido nas **figuras 1.b, 1.d e 1.j**, embora retilínea, as formas que as hifas assumem são bastante diversas. Além disso aglomerados de hifas podem ser formados (**figura 1.f**). A irregularidade de formas que as hifas podem assumir nas imagens pode ser um dos fatores que levaram a segmentação de hifas apresentada nesse artigo não ser tão eficiente em alguns casos.

Como visto nas **imagens 1.f, 1.h e 1.o**, o modelo obtido distinguiu satisfatoriamente, em alguns casos, hifas que estavam dentro das raízes daquelas que não estavam. Esse resultado se torna imprescindível caso, em um passo adiante, se deseje realizar um cálculo de colonização de raiz por FMA. Desta forma, seria possível, através da divisão do número de pixels classificados como FMA pelo número de pixels classificados como pertencentes a raiz, uma métrica de colonização da raiz por FMA. Para tal seria necessário anteriormente uma estratégia de segmentação de raízes. Esta poderia se dar em imagens com fundo claramente distinto da raiz, através de operações

de limiar (traduzido da palavra inglesa (*threshold*), ou mesmo em casos mais complexos, produzir um algoritmo de segmentação de raízes baseado em aprendizagem profundo.



**Figura 1.** Exemplos das predições obtidas através do uso do modelo (imagens à esquerda) e as hifas contidas nas raízes que se tenta prever (imagens à direita). Hifas externas à raiz foram propositalmente desconsideradas no treinamento, de modo que a rede neural segmentasse apenas hifas dentro ou imediatamente na superfície da raiz

Como visto em algumas imagens, (por exemplo **1.b** e **1.d**) pequenas regiões de pixels foram classificadas como pertencentes a hifas. Essas regiões seriam pequenas demais e não conectadas a outros pixels para formar filamentos o que excluiria a real possibilidade de elas serem hifas. Uma possível saída para resolver esse problema é a utilização de operações de filtragem que excluam objetos presentes na imagem que não atendam uma determinada condição como, por exemplo, não possuírem área (em termos de quantidade de pixels) menores que um valor limite.

Nosso resultado promissor utilizando técnica de segmentação semântica se soma a outros que já a utilizaram como na classificação de veículos presentes em imagens aéreas (Audebert et al. 2017), detecção de células cancerosas (Karabag et al. 2020), segmentação de núcleos de célula (Yang et al. 2020) e detecção de mutações cromossômicas (Pardo et al. 2018).

O resultado desse experimento é mais um resultado promissor alcançado utilizando uma UNet. Essa arquitetura, criada por Ronneberger et al. (2015), vem sendo bastante utilizada para a resolução de uma série de problemas como, por exemplo, em imagens de diagnósticos de medicina (Zhou et al. 2018), remoção de artefatos presentes em imagens de tomografia que dificultam o diagnóstico (Guan et al. 2018) e segmentação de minerais presentes em imagens do relevo oceânico (Song et al. 2019).

Uma das vantagens dessa arquitetura é a pequena necessidade de recurso computacional quando comparada a outras arquiteturas, principalmente quando se acopla técnicas de aumento de dados, o que leva a possibilidade de seu uso por uma série de

laboratórios que possuem limitação de processamento e de quantidade de dados disponíveis para treinamento (Ronneberger et al. 2015). Outras arquiteturas de aprendizado de máquina podem levar até dias para que o treinamento seja completado (Ferentinos, 2018), um problema que não ocorre quando se utiliza uma UNet.

O resultado apresentado nesse experimento se junta a outros trabalhos que utilizaram aprendizado de máquina no estudo de fungos. Zielinski et al. (2020) apresentam um algoritmo de aprendizado profundo que é capaz de classificar fungos causadores de doenças através da análise de imagens, de forma que o tempo para diagnóstico seja reduzido, bem como os custos. Outro exemplo se dá na aplicação de algoritmo de aprendizado de máquina para acelerar a classificação de fungos (Vu et al. 2020).

Os usos de outras abordagens de aprendizado de máquina também podem ser utilizados no estudo de fungos micorrízicos arbusculares. Scascia et al. (2021) apresentaram a utilização de um software de aprendizado de máquina (baseado na aplicação de filtros) para detectar e analisar a presença de FMAs em raízes. Esses autores propõem um índice de colonização micorrizica baseado no número de pixels na imagem considerados como estruturas de micorriza dentro da raiz. Embora promissor, este resultado ainda apresenta algumas limitações. Por exemplo, foi utilizado, para o processo de treinamento do modelo de aprendizado de máquina e seu uso, um software pago. Outro problema é que, em *backgrounds* menos distintos das estruturas micorrízicas ou da raiz, algoritmos baseados em simples filtros como Sobel e gaussian podem não ser suficientes para realizar essa distinção, já que, na condução do nosso experimento tentamos utilizar esse tipo de abordagem e os resultados não foram satisfatórios. Outra limitação é que a

presença de artefatos pode confundir o algoritmo nessa tarefa de classificação. Além disso, nesse experimento, não foi possível realizar a distinção entre estruturas de micorriza presentes na raiz, ou seja, hifas, arbúsculos e vesículas eram considerados as mesmas estruturas o que, a depender da análise que se deseja realizar, pode não ser o suficiente para resolver um determinado problema.

Uma estratégia mais robusta para a solução destes problemas pode se dar através do uso de algoritmos de aprendizado de máquina baseados em rede neural convolucional (RNC) como no nosso artigo e em Evangelisti et al. (2021). Os últimos autores citados apresentaram um algoritmo capaz de identificar e quantificar a colonização por FMA's em imagens de raízes colonizadas por este grupo de fungos. Uma das vantagens dessa abordagem utilizada é a possibilidade de o usuário treinar sua própria RNC o que aumenta a versatilidade de aplicação para dados que difiram dos dados utilizados para criar o modelo de aprendizado original.

A utilização de algoritmos de aprendizado de máquina apresenta algumas vantagens quando comparada com as metodologias tradicionais utilizadas no estudo de FMA's. As análises tradicionais de colonização são bastante dispendiosas em termos de tempo gasto e de impacto sobre a amostra já que estas podem ser destruídas no processo de forma que se algum erro ocorrer a amostra não pode ser reaproveitada. Além disso, os resultados destas análises podem variar conforme o observador que as realize (Vierheilig et al. 2005). Com o uso de ferramentas de aprendizado de máquina parte destes problemas podem ser solucionados. Por exemplo, a variabilidade pode ser extinta quando um software realiza a análise de colonização micorrízica ou conte as estruturas de micorriza presentes em uma imagem. Outra vantagem é a possibilidade de capturar as imagens em

um momento e somente posteriormente realizar a análise. Essa estratégia evita que sejam realizadas análises em amostras deterioradas, o que, em última instância, pode diminuir a precisão do resultado obtido.

Embora promissor, abordagens que utilizam ferramentas de aprendizado de máquina (quando se utiliza programação) encontram barreiras de aplicação no campo da biologia. Por exemplo, a comunidade científica que lida com biologia ainda não possui familiaridade com linguagens de programação (Zhang et al. 2021). Uma outra preocupação é que esses algoritmos obtidos performem bem apenas em condições muito específicas (em imagens obtidas em processo muito similar àquele em que ocorreu a captura das imagens que serviram como base para a obtenção do modelo) (Ramcharan et al. 2019)

Outro fator é a capacidade de processamento necessária para o treinamento e uso de algoritmos de redes neurais convolucionais. Muitos laboratórios não possuem a capacidade para tal. Um fator importante, também, é o preço de softwares baseados em aprendizado de máquina, que podem ser utilizados para estudo dos mais diferentes organismos. Nesse contexto, a adoção de softwares e códigos de acesso livre constitui-se em um importante avanço para a comunidade científica. Desta forma, um maior número de laboratórios tem acesso a análises e até mesmo equipamentos (*hardware*) que possuem a capacidade necessária para realizar determinada análise com qualidade.

#### **4. Conclusões**

Neste artigo demonstramos o potencial de uma técnica de aprendizado profundo, nomeadamente uma UNet modificada, para identificar hifas de fungos micorrízicos



arbusculares em imagens de raízes colonizadas. Nosso artigo se junta a uma crescente, mas ainda incipiente, tentativa de utilizar ferramentas computacionais no estudo de organismos do solo.

### Referências

- Abreu GM, Schiavo JA, Abreu PM, Bobadilha GS, Rosset JS. 2018. Crescimento inicial e absorção de fósforo e nitrogênio de *Enterolobium contortisiliquum* inoculada com fungos micorrízicos arbusculares. **Revista de Ciências Agrárias**, v. 41, n. 1, p. 156–164.
- Akintayo A, Tylka GL, Singh, AK SINGH, Ganapathysubramanian B, Singh A, Sarkar S. 2018. A deep learning framework to discern and count microscopic nematode eggs. **Scientific Reports**, v. 8, n. 1, p. 1–11.
- Audebert N, Le Saux B, Lefèvre S. 2017. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. **Remote Sensing**, v. 9, n. 4.
- Bano SA, Ashfaq D. 2013. Role of mycorrhiza to reduce heavy metal stress. **Natural Science**, v. 5, n. 12.
- Bressan W, Siqueira JO, Vasconcellos CA, Purcino AAC. 2001. Fungos micorrízicos e fósforo, no crescimento, nos teores de nutrientes e na produção do sorgo e soja consorciados. **Pesquisa Agropecuária Brasileira**, v. 36, n. 2, p. 315–323.
- Cordeiro ECN, Resende JTV, Córdova KRV, Nascimento DA, Saggin Júnior OJ, Zeist AR, Favaro R. 2019. Arbuscular mycorrhizal fungi action on the quality of strawberry fruits. **Horticultura Brasileira**, v. 37, n. 4, p. 437–444.
- Dhalaria R, Kumar D, Kumar, Nepovimova E, Kuca K, Islam MT, Verma R. 2020. Arbuscular mycorrhizal fungi as potential agents in ameliorating heavy metal stress in plants. **Agronomy**, v. 10, n. 6.
- Egerton-Warburton LM, Querejeta JI, Allen MF. 2007. Common mycorrhizal networks provide a potential pathway for the transfer of hydraulically lifted water between plants. **Journal of Experimental Botany**, v. 58, n. 6, p. 1473–1483.
- Evangelisti E, Turner C, McDowell AA, Shenhav L, Yunusov T, Gavrin A, Servante EK, Quan C, Schornack S. 2021. Deep learning-based quantification of arbuscular mycorrhizal fungi in plant roots. **New Phytologist**, p. 2207–2219.
- Feitosa AG, Santos DR. 2017. Fósforo e fungos micorrízicos no crescimento de mudas arbóreas da caatinga. **Agropecuária Científica no Semiárido**, v. 12, n. 4, p. 392–396. Disponível em: <<http://revistas.ufcg.edu.br/acsa/index.php/ACSA/index>>.

Ferentinos KP. 2018. Deep learning models for plant disease detection and diagnosis. **Computers and Electronics in Agriculture**, v. 145, n. January, p. 311–318. Elsevier. Disponível em: <<https://doi.org/10.1016/j.compag.2018.01.009>>.

Guan S, Khan AA, Sikdar S, Chitnis PV. 2020. Fully Dense UNet for 2-D Sparse Photoacoustic Tomography Artifact Removal. **IEEE Journal of Biomedical and Health Informatics**, v. 24, n. 2, p. 568–576.

Ingraffia R, Amato G, Frenda AS, Giambalvo D. 2019. Impacts of arbuscular mycorrhizal fungi on nutrient uptake, N<sub>2</sub> fixation, N transfer, and growth in a wheat/faba bean intercropping system. **PLoS ONE**, v. 14, n. 3, p. 1–16.

Karabag C, Jones ML, Peddie CJ, Weston AE, Collinson LM, Reyes-Aldoroso CC. 2020. Semantic segmentation of HeLa cells: An objective comparison between one traditional algorithm and four deep-learning architectures. **PLoS ONE**, v. 15, n. 10 October, p. 1–21.

Koske RE, Gemma JN. 1989. A modified procedure for staining roots to detect VA mycorrhizas. **Mycological Research**, v. 92, n. 4, p. 486–488. British Mycological Society. Disponível em: <[http://dx.doi.org/10.1016/S0953-7562\(89\)80195-9](http://dx.doi.org/10.1016/S0953-7562(89)80195-9)>.

Hoagland DR 1950. The water-culture method for growing plants without soil. **Circular. California agricultural experiment station**, v. 347, n. 2nd edit.

Honorato AC, Oliveira JRG, Passos AM, Yano-Melo AM. 2020. Mycorrhizal inoculation on the production of seedlings of native caatinga species. **Floresta e Ambiente**, v. 27, n. 2.

Lahmer H, Oueslati AE, Lachiri Z. 2020. Classification of DNA Microarrays Using Deep Learning to identify Cell Cycle Regulated Genes. **2020 International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2020**, p. 0–4.

Melloni R, Salomão FR, Silva LFO, Mescolotti DLC, Cardoso EJBN. 2020. Mycorrhiza formation on olive varieties (*Olea europaea* L.) fit for cultivation in the state of Minas Gerais. **Ciencia Florestal**, v. 30, n. 4, p. 1255–1265.

Meyer JG. 2021. Deep learning neural network tools for proteomics. **Cell Reports Methods**, v. 1, n. 2, p. 100003. Elsevier Company. Disponível em: <<https://doi.org/10.1016/j.crmeth.2021.100003>>.

Pardo E, Morgado JMT, Malpica N. 2018. Semantic segmentation of mFISH images using convolutional networks. **Cytometry Part A**, v. 93, n. 6, p. 620–627.

Ramcharan A, McCloskey P, Baranowski K, Mbilinyi, N, Mrisho L, Ndalaha M, Legg J, Hughes DP. 2019. A mobile-based deep learning model for cassava disease diagnosis. **Frontiers in Plant Science**, v. 10, n. March, p. 1–8.

Ronneberger O, Fischer P, Brox T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: **International Conference on Medical Image**

**Computing and Computer-Assisted Intervention**, p. 234-241, Springer.

Sciascia IA, Andrea C, Novero M. 2018. Quantifying Root Colonization in Arbuscular Mycorrhizas by Image Segmentation and Machine Learning, Research Square (in print).

Song W, Zheng N, Liu X, Qiu L, Zheng R. 2019. An Improved U-Net Convolutional Networks for Seabed Mineral Image Segmentation. **IEEE Access**, v. 7, p. 82744–82752.

Thevenoux R, Le VL, Villessèche H, Buisson A, Burton-Aimar M, Folcher L, Parisey N. 2021. Image based species identification of Globodera quarantine nematodes using computer vision and deep learning. **Computers and Electronics in Agriculture**, v. 186, n. May.

Thoma MA. 2016. Survey of Semantic Segmentation. , p. 1–16. Disponível em: <<http://arxiv.org/abs/1602.06541>>.

Vieira CV. 2021. Interação entre fungos micorrízicos arbusculares, comunidades de procaríotos e indicadores da qualidade do solo em áreas de mineração de carvão em recuperação. 2021. Tese (Doutorado). Programa de Pós-Graduação em Engenharia Ambiental, Universidade Regional de Blumenau, 2021.

Vierheilig H, Schweiger P, Brundrett M. 2005. An overview of methods for the detection and observation of arbuscular mycorrhizal fungi in roots. **Physiologia plantarum**, v. 125, p.393-404.

Vu D, Groenewald M, Verkley G. 2020. Convolutional neural networks improve fungal classification. **Scientific Reports**, v. 10, n. 1, p. 1–12. Nature Publishing Group UK. Disponível em: <<https://doi.org/10.1038/s41598-020-69245-y>>.

Wang B, Qiu YL. 2006. Phylogenetic distribution and evolution of mycorrhizas in land plants. **Mycorrhiza**, v. 16, n. 5, p. 299–36.

Yang L, Ghosh RP, Matthew Franklin J. 2020. NuSeT: A deep learning tool for reliably separating and analyzing crowded cells. **PLoS Computational Biology**, v. 16, n. 9, p. 1–20. Disponível em: <<http://dx.doi.org/10.1371/journal.pcbi.1008193>>.

Zhang Y, Jiang H, Ye T, Juhas M. 2021. Deep Learning for Imaging and Detection of Microorganisms. **Trends in Microbiology**, v. 29, n. 7, p. 569–572. Elsevier Ltd. Disponível em: <<https://doi.org/10.1016/j.tim.2021.01.006>>.

Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J. 2018. Unet++: A nested u-net architecture for medical image segmentation. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 11045 LNCS, p. 3–11.

Zielinski B, Sroka-Oleksiak A, Rymarczyk D, Piekarczyk A, Brzywczy-Woch M. 2020.. Deep learning approach to describe and classify fungi microscopic images. **PLoS ONE**, v. 15, n. 6 June, p. 1–16.

## Considerações Finais

Abordagens que envolvem o uso de ciência aberta vêm ganhando *momentum* por conta de uma crescente necessidade e demanda por transparência dos dados que norteiam a pesquisa científica. Desta forma, os resultados se tornam mais confiáveis e reprodutíveis. Dentro da ciência aberta, há um movimento que tenta incentivar a utilização e criação de softwares e hardwares de código livre. Estes irão proporcionar, em última instância, a popularização do uso de uma determinada metodologia.

Nesta dissertação todos os equipamentos utilizados (criados no nosso grupo de pesquisa) são de código aberto. Adicionalmente, houve uma preocupação em utilizar equipamentos de baixo custo (sejam eles os próprios equipamentos de código livre ou não). Desta maneira, acreditamos que esta dissertação contribui significativamente para a popularização de hardwares de baixo custo. Os estágios automatizados que aqui foram apresentados constituem uma opção mais barata que os estágios automatizados tradicionais. Esta realidade proporciona que um maior número de pesquisadores possa utilizar esta tecnologia.

Estes estágios automatizados aumentam o poder de análise dos laboratórios, bem como otimiza o tempo utilizado para as rotinas laboratoriais. Exemplos muito claros desta realidade são descritos ao longo dos três artigos desta dissertação. Por exemplo, NemaStage (controlado por StageControl) é bastante eficiente para o escaneamento automatizado de lâminas contendo raízes parasitadas por nematoides. Desta forma, o pesquisador não precisa mais operar o microscópio manualmente na procura por estes organismos parasitando raízes. O pesquisador pode optar pelo escaneamento automatizado enquanto dedica o tempo que utilizaria para esta operação, exaustiva e que necessita elevada atenção, para outra atividade laboratorial. Adicionalmente, pode-se fazer uso da abordagem utilizada no **artigo 3** para selecionar e juntar imagens contíguas contendo nematoides posteriormente ao escaneamento. Desta maneira é reduzido drasticamente o número de imagens que o pesquisador tem que avaliar.

Os resultados de junção de imagens contendo nematoides alcançados nos experimentos descritos anteriormente, se torna fundamental ao proporcionar operações de contagem e medições mais eficientes que abordagens tradicionais. Os softwares (e os códigos que os criaram) para seleção, junção e medição de nematoides, também estão disponíveis de maneira aberta. Esta é uma importante contribuição, especialmente pela adaptabilidade que o usuário pode dar a estes códigos. Por exemplo, mesmo um pesquisador que não milite no campo da microbiologia, pode adaptar esses códigos para um determinado objetivo que necessite selecionar imagens contendo objetos de interesse e posteriormente juntar imagens contíguas contendo estas estruturas.

As segmentações alcançadas nestes experimentos foram de altíssima qualidade para os objetivos traçados. Interessantemente, estes resultados foram alcançados utilizando técnicas de segmentação simples (com pouca necessidade de dados rotulados) e através de ferramentas abertas para o público (geralmente para a obtenção de modelos de segmentação se faz necessário computadores de elevada capacidade de treinamento). Para o modelo de segmentação de nematoides obtido, já temos uma aplicação prática, isto é, ele serve como base tanto para a seleção de imagens contendo nematoides que posteriormente sofrerão junção, bem como para segmentar os nematoides que sofrerão medições em NemaMeasure. Para o modelo que segmentou hifas de fungo micorrizico

arbuscular, conseguimos segmentações de qualidade que futuramente pode subsidiar, em parte, processos de medição e de índices de colonização micorrízica.

Vale ressaltar que os softwares e hardwares aqui apresentados são primeiras versões que constituem mínimos produtos viáveis. Desta forma, estes equipamentos e softwares, podem sofrer alterações que os otimizem e/ou facilitem seus usos, como, por exemplo, a criação de uma interface gráfica.

## Apêndices

### Apêndice 1

```

from classes import Stage, Camera
stage = Stage("COM7")
camera = Camera(1)
if __name__ == "__main__":
    camera.capture_during_scan(x= 0, y= 0, crop_bound= [150, 650, 350,
850])
    stage.scan(xstart = 12, xend =34, step_x= .8,
               ystart = 4, yend =15, step_y = .8,
               camera = camera)
    camera.convert2tif()

```

### Apêndice 2

```

import serial
from time import sleep
from math import ceil
import cv2
import os

class Stage():
    def __init__(self, port):
        self.connect(port)
        self.go_home()
        self.wait_move()

    def connect(self, port):
        self.serial = serial.Serial(port, 115200, timeout=3)
        sleep(3)

    def send_gcode_str(self, gcode_str, debug = False):
        debug = True
        self.serial.flushInput()
        sleep(.01)
        if debug:
            print(gcode_str)
        gcode_str += "\n"
        self.serial.write(gcode_str.encode())
        grbl_out = self.serial.readline() # Wait for robot response
        with carriage return
        if debug:
            print(grbl_out.strip())
        return (grbl_out.strip().decode("ascii"))

    def is_done(self):
        status = self.send_gcode_str("G4 P0")
        if status == '':
            self.send_gcode_str("?")
        return status == 'ok'

    def go_home(self):
        self.send_gcode_str("$H")
        #self.send_gcode_str("G10 P0 L20 X0 Y0 Z0")

    def wait_move(self):
        while not self.is_done():
            sleep(.9)

```

```

        pass
    return True

def move_x(self, x):
    self.send_gcode_str(f"G90 X{x}")
    return self.wait_move()

def move_y(self, y):
    self.send_gcode_str(f"G90 Y{-y}")
    return self.wait_move()

def scan(self, camera, xend, yend, step_x, step_y, xstart = 0,
ystart = 0):
    range_x = xend - xstart
    range_y = yend - ystart
    nsteps_x = ceil(range_x/float(step_x)) + 1
    nsteps_y = ceil(range_y/ float(step_y)) + 1
    flip = False
    for y in range(nsteps_y):
        self.move_y(ystart + y * step_y)
        if flip:
            x_seq = range(nsteps_x) [::-1]
        else:
            x_seq = range(nsteps_x)
        flip = not flip
        for x in x_seq:
            self.move_x(xstart + x * step_x)
            camera.capture_during_scan(x, y)
        camera.cap.release()

class Camera():
    def __init__(self, camera):
        self.connect(camera=camera)
        self.set_resolution()

    def connect(self, camera):
        self.cap = cv2.VideoCapture(camera)

    def set_resolution(self):
        self.cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
        self.cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)

    def capture(self):
        image = .1 * self.cap.read()[1]
        for i in range(9):
            image += .1 * self.cap.read()[1] # return a single frame
in variable `frame`
        return image

    def save(self, image_obj, filepath):
        cv2.imwrite(filepath, image_obj)

    def crop(self, image, xstart, xend, ystart, yend):
        return image[xstart:xend, ystart:yend]

    def convert2tif(self):
        images = [f for f in os.listdir() if ".png" in f]
        for img in images:
            imagem = cv2.imread(img)
            cv2.imwrite(img.replace("png", "tif"), imagem)
            os.remove(img)

```

```

    def capture_during_scan(self, x, y, crop_bound= [150, 650, 350,
850]):
        self._image = self.capture()
        self._image = self.crop(self._image, crop_bound[0],
crop_bound[1],
                                crop_bound[2], crop_bound[3])
        self.save(self._image, f"{x}_{y}.png")

```

### Apêndice 3

```

from classes import Stage, Camera
stage = Stage("COM3")
camera = Camera(0)

if __name__ == "__main__":
    stage.scan(xstart = 8, xend = 12, step_x = 1,
              ystart = 2, yend = 6, step_y = 1,
              camera = camera, invert_x=False,
              invert_y = False)

```

### Apêndice 4

```

import serial
from time import sleep
from math import ceil
import cv2

class Stage():
    def __init__(self, port):
        self.connect(port)
        self.go_home()
        self.wait_move()

    def connect(self, port):
        self.serial = serial.Serial(port, 115200, timeout=3)
        sleep(3)

    def send_gcode_str(self, gcode_str, debug = True):
        self.serial.flushInput()
        sleep(.01)
        if debug:
            print(gcode_str)
        gcode_str += "\n"
        self.serial.write(gcode_str.encode())
        grbl_out = self.serial.readline() # Wait for robot response
        with carriage return
        if debug:
            print(grbl_out.strip())
        return (grbl_out.strip().decode("ascii"))

    def is_done(self):
        status = self.send_gcode_str("G4 P0")
        if status == '':

```



```

        self.send_gcode_str("?")
        return status == 'ok'

def go_home(self):
    self.send_gcode_str("$H")

def wait_move(self):
    while not self.is_done():
        sleep(.5)
        pass
    return True

def move_x(self, x, invert_x = False):
    if invert_x:
        x = -x
    self.send_gcode_str(f"G90 X{x}")
    return self.wait_move()

def move_y(self, y, invert_y = False):
    if invert_y:
        y = -y
    self.send_gcode_str(f"G90 Y{y}")
    return self.wait_move()

def scan(self, camera, xend, yend, step_x, step_y, xstart = 0,
          ystart = 0,
          invert_x = False, invert_y = False):
    range_x = xend - xstart
    range_y = yend - ystart
    nsteps_x = ceil(range_x/float(step_x)) + 1
    nsteps_y = ceil(range_y/ float(step_y)) + 1
    flip = False
    for y in range(nsteps_y):
        self.move_y(ystart + y * step_y, invert_y = invert_y)
        if flip:
            x_seq = range(nsteps_x)[::-1]
        else:
            x_seq = range(nsteps_x)
        flip = not flip
        for x in x_seq:
            self.move_x(xstart + x * step_x, invert_x = invert_x)
            camera.capture_during_scan(x, y)
    camera.cap.release()

class Camera():
    def __init__(self, camera):
        self.connect(camera=camera)

    def connect(self, camera):
        self.cap = cv2.VideoCapture(camera)

    def capture(self):
        image = .1 * self.cap.read()[1]
        for i in range(9):
            image += .1 * self.cap.read()[1] # return a single frame
in variable `frame`
        return image

    def save(self, image_obj, filepath):
        cv2.imwrite(filepath, image_obj)

```

```

def crop(self, image, xstart, xend, ystart, yend):
    return image[xstart:xend, ystart:yend]

def capture_red_layer_and_make_square(self, image):
    return image[:, 0:480, 0]

def capture_during_scan(self, x, y):
    self.image = self.capture()
    self._image =
self.capture_red_layer_and_make_square(self._image)
    self.save(self._image, f"imagens/nem_{x}_{y}.png")

```

## Apêndice 5

```

from tensorflow.keras.layers import Conv2D, Dropout, MaxPooling2D,
Lambda, Input
from tensorflow.keras.layers import Conv2DTranspose, concatenate
from tensorflow.keras import Model
from skimage.measure import label, regionprops
from skimage.morphology import skeletonize, medial_axis, closing,
square
import matplotlib.pyplot as plt
import numpy as np
from skimage.io import imsave, imread
import os
from glob import glob

import pandas as pd
from skan import Skeleton

from stitching import Image_Matrix, Stitcher

plt.rcParams["figure.figsize"] = (10, 10)

class Model_Nematoides():
    def __init__(self):
        self.load_model(model_path = "model_for_nematodes.h5")

    def load_model(self, model_path):
        kw = dict(activation='relu', kernel_initializer='he normal',
padding='same')

        IMG_HEIGHT = 480
        IMG_WIDTH = 480
        IMG_CHANNELS = 3

        inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
        # s = Lambda(lambda x: x / 255)(inputs)

        # Contraction path
        c1 = Conv2D(32, (3, 3), **kw)(inputs)
        c1 = Dropout(0.1)(c1)
        c1 = Conv2D(32, (3, 3), **kw)(c1)
        p1 = MaxPooling2D((2, 2))(c1)

        c2 = Conv2D(64, (3, 3), **kw)(p1)
        c2 = Dropout(0.1)(c2)
        c2 = Conv2D(64, (3, 3), **kw)(c2)
        p2 = MaxPooling2D((2, 2))(c2)

```

```

c3 = Conv2D(128, (3, 3), **kw) (p2)
c3 = Dropout(0.2) (c3)
c3 = Conv2D(128, (3, 3), **kw) (c3)
p3 = MaxPooling2D((2, 2)) (c3)

c4 = Conv2D(256, (3, 3), **kw) (p3)
c4 = Dropout(0.2) (c4)
c4 = Conv2D(256, (3, 3), **kw) (c4)
p4 = MaxPooling2D(pool_size=(2, 2)) (c4)

c5 = Conv2D(512, (3, 3), **kw) (p4)
c5 = Dropout(0.3) (c5)
c5 = Conv2D(512, (3, 3), **kw) (c5)

kw_conv2transp = dict(strides=(2, 2), padding='same')
u6 = Conv2DTranspose(256, (2, 2), **kw_conv2transp) (c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(256, (3, 3), **kw) (u6)
c6 = Dropout(0.2) (c6)
c6 = Conv2D(256, (3, 3), **kw) (c6)

u7 = Conv2DTranspose(64, (2, 2), **kw_conv2transp) (c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(128, (3, 3), **kw) (u7)
c7 = Dropout(0.2) (c7)
c7 = Conv2D(128, (3, 3), **kw) (c7)

u8 = Conv2DTranspose(128, (2, 2), **kw_conv2transp) (c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(64, (3, 3), **kw) (u8)
c8 = Dropout(0.1) (c8)
c8 = Conv2D(64, (3, 3), **kw) (c8)

u9 = Conv2DTranspose(64, (2, 2), **kw_conv2transp) (c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(64, (3, 3), **kw) (u9)
c9 = Dropout(0.1) (c9)
c9 = Conv2D(64, (3, 3), **kw) (c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid') (c9)

self.model = Model(inputs=[inputs], outputs=[outputs])
self.model.compile(optimizer='adam',
loss='binary_crossentropy', metrics=['accuracy'])
self.model.summary()
self.model.load_weights(model_path)

def read_image(self, image_path):
img = imread(image_path)[:,:,:3]
return img

def preprocess(self, img):
img = np.expand_dims(img, axis=0)
return img

def predict(self, img, threshold = .5):
def resize_mask(mask):
return mask[0, :, :, 0]
mask = self.model.predict(img)
mask[mask < threshold] = 0

```

```

mask[mask >= threshold] = 1
return resize_mask(mask)

def close_mask(self, mask, square_size = 2):
mask = closing(mask, square(square_size))
return mask

def erase_small_region(self, image, min_area = 160):
labels = label(image)
for region in regionprops(labels):
    if region.area < min_area:
        for coord in region.coords:
            image[coord[0], coord[1]] = 0
return image

def segment(self, img_path, save_masks = False):
print(img_path)
img_or = self.read_image(img_path)
img = self.preprocess(img_or)
mask1 = self.predict(img, threshold=.5)
mask2 = self.close_mask(mask1, square_size = 3)
mask3 = self.erase_small_region(mask2, min_area = 1000)
if save_masks:
    filename = img_path.split("/") [2]
    print(filename)
    imsave(f"predicted_masks/{filename}", mask3)
return mask3

def detect_nematodes(self, img_path):
mask = self.segment(img_path)
area = mask.sum()
if area > 0:
    print("Yes")
return area > 0

def make_matrix(self, img_dir):
xs = [int(f.split("__")[1]) for f in os.listdir(img_dir) if
".png" in f]
ys = [int(f.split("__")[2]) for f in os.listdir(img_dir) if
".png" in f]
return (np.zeros(shape=(max(ys) + 1, max(xs) + 1)))

def populate_matrix(self, matrix, found_nematodes_list):
xs = [int(f.split("__")[1]) for f in found_nematodes_list]
ys = [int(f.split("__")[2]) for f in found_nematodes_list]
for x, y in zip(xs, ys):
    matrix[y, x] = 1
return matrix

def apply_segmentation(self):
with open("found_nematodes.txt", "w") as handle:
    for i, img in enumerate(glob("imagens_col/*__.png")):
        print(i)
        detected = model_nematodes.detect_nematodes(img)
        if detected:
            handle.write(img + "\n")

def stitch(self):
mat = model_nematodes.make_matrix("./imagens_col")
with open("found_nematodes.txt", "r") as handle:
    found_nematodes = handle.readlines()

```

```

mat = model_nematodes.populate_matrix(mat, found_nematodes)
image_matrix = Image_Matrix(mat)
regions = image_matrix.get_regions()
stitcher = Stitcher(regions)
for i, region in enumerate(regions.values()):
    stitcher.get_images("./imagens_col", i)
    stitcher.stitch_region(i)

if __name__ == "__main__":
    model_nematodes = Model_Nematoides()
    model_nematodes.apply_segmentation()
    model_nematodes.stitch()

```

## Apêndice 6

```

from skimage.io import imread, imsave
from glob import glob
import json
import numpy as np
import os
import shutil
from skimage.draw import polygon
import cv2
import pandas as pd

def combine_layers():
    images = glob("imagens/*_*.png")
    for image in images:
        img = imread(image)
        R = img[:, :, 0]
        G = img[:, :, 1]
        B = img[:, :, 2]
        RelR = .5 * B + .5 * G
        imsave('pre_processed/' + image.split("imagens/")[1],
        RelR.astype(int))

json_path= "/path/"
with open(json_path, "r") as file:
    txt = file.read()
    masks = json.loads(txt)

class Masks():
    def __init__(self, json_path):
        self.load_masks_from_json(json_path= json_path)

    def load_masks_from_json(self, json_path):
        with open(json_path, "r") as file:
            txt = file.read()
            masks = json.loads(txt)
            self.masks_metadata = masks["_via_img_metadata"]

    def get_polygons_coord(self, key):
        return self.masks_metadata[key]['regions']

    def draw_polygons(self, regions):
        img = np.zeros((480, 480), dtype = np.uint8)
        if regions:
            for region in regions:
                x = region['shape_attributes']['all_points_x']

```

```

        x = [c_x - 1 for c_x in x]
        y = region['shape_attributes']['all_points_y']
        y = [c_y - 1 for c_y in y]
        rr, cc = polygon(y, x)
        img[rr, cc] = 254
    return img

def make_binary_masks(self):
    if os.path.isdir("./masks"):
        shutil.rmtree("./masks")
    os.mkdir("./masks")
    for mask in self.masks_metadata.values():
        regions = mask['regions']
        img = self.draw_polygons(regions)
        imsave(f"./masks/{mask['filename']}", img)

if __name__ == "__main__":
    masks = Masks("/path")
    masks.make_binary_masks()

```

## Apêndice 7

```

from segmentation import Model_Nematoides
from skimage.transform import resize
from skimage.measure import label, regionprops
from skimage.morphology import skeletonize, medial_axis, closing,
square
from skimage.io import imread
from skimage.segmentation import clear_border
from skan import Skeleton
import pandas as pd
from glob import glob
import matplotlib.pyplot as plt

class Measure():
    def __init__(self):
        self._model = Model_Nematoides()

    def calculate_width(self, image):
        _, distance = medial_axis(image, return_distance=True)
        max_width = max(distance[image > 0]) * 2
        return(max_width)

    def measure_nematoide(self, img_path, mask, show_image=False):
        label_im = label(mask)
        results = []
        if regionprops(label_im):
            for count, region in enumerate(regionprops(label_im)):
                skel = skeletonize(region.image)
                Skel = Skeleton(skel)
                length = Skel.path_lengths()[0] if Skel.n_paths == 1
            else 0
            width = self.calculate_width(region.image) if
            Skel.n_paths == 1 else 0
            fatness = float(region.area) / Skel.path_lengths()[0]
        if Skel.n_paths == 1 else 0
        result = {"id": count,
                  "width": width,
                  "npaths": Skel.n_paths,

```

```

        "length": length,
        "coords": region.coords,
        "bbox": region.bbox,
        "skel": skel,
        "fatness": fatness,
        "centroid_x_pix": region.centroid[1],
        "centroid_y_pix": region.centroid[0],
        "centroid_x": region.centroid[1] /
float(mask.shape[1]),
        "centroid_y": 1 - region.centroid[0] /
float(mask.shape[0]),
        "image": region.image}
    results.append(result)
    data = pd.DataFrame(results)
    self.annotate_img(img_path, data)
    else:
        results = [{"id": None, "width": None, "npaths": None,
"length": None, "coords": None, "bbox": None, "skel": None, "fatness":
None, "centroid_x": None,
                    "centroid_y": None, "centroid_y_pix": None,
"centroid_x_pix": None, "image": None}]
        data = pd.DataFrame(results)
        return data

    def find_and_measure_one(self, img_path):
        mask = self._model.segment(img_path, save_masks=True)
        data = self.measure_nematoids(mask = mask, img_path =
img_path, show_image = True)
        return data

    def superimpose_skeleton(self, skel, bbox, image, coords):
        for x, y in coords:
            if skel[x - bbox[0], y - bbox[1]] > 0:
                image[x, y] = 1
        return image

    def annotate_img(self, img_path, data):
        img = imread(img_path)
        filename = img_path.split("/")[-2]
        for i, row in data.iterrows():
            img = self.superimpose_skeleton(skel = row.skel, bbox =
row.bbox, image = img, coords = row.coords)
            fig, ax = plt.subplots(figsize=(10, 10))
            ax.imshow(img, cmap="gray")
            for i, row in data.iterrows():
                text = f"ID: {row.id}; Length: {row.length:.2f}; Width:
{row.width:.2f}"
                ax.text(row.centroid_x_pix, row.centroid_y_pix, text,
color="black")
            plt.axis('off')
            fig.savefig("./annotated_imgs/" + filename,
                        bbox_inches='tight', pad_inches=0, dpi=624 / 10,
transparent=False)

    def find_and_measure_all(self, img_dir = "./nem_to_measure/"):
        results = []
        for file in glob(img_dir + "*.png"):
            data = self.find_and_measure_one(file)
            results.append(data)
        return pd.concat(results)

```





```

        self.check_new_cells()
    return self.regions

class Stitcher():
    def __init__(self, regions):
        self.regions = regions

    def make_region_square(self, region_idx):
        key = list(self.regions.keys())[region_idx]
        region = self.regions[key]
        i_s = [r[0] for r in region]
        j_s = [r[1] for r in region]
        topleft_corner = (min(i_s), min(j_s))
        bottom_right_corner = (max(i_s), max(j_s))
        for i in range(topleft_corner[0], bottom_right_corner[0] + 1):
            for j in range(topleft_corner[1], bottom_right_corner[1] +
1):
                if not (i, j) in region:
                    region.append((i, j))
        return (topleft_corner, bottom_right_corner, region)

    def get_images(self, image_dir, region_idx):
        topleft_corner, bottom_right_corner, square =
self.make_region_square(region_idx)
        if os.path.isdir("./temp"):
            shutil.rmtree("./temp")
        os.mkdir("./temp")
        empty_image = np.zeros(shape = (480, 480, 3))
        for i, j in square:
            file_name = f"{image_dir}/nem_{j}_{i}.png"
            if os.path.isfile(file_name):
                img = imread(file_name)[:,:0:3]
            else:
                img = empty_image
            out_file = f"./temp/nem_{j}_{i}.tif"
            imsave(out_file, img)

    def stitch_region(self, region_idx):
        topleft_corner, bottom_right_corner, square =
self.make_region_square(region_idx)
        max_x, max_y = bottom_right_corner
        min_x, min_y = topleft_corner
        fiji_macro = """run("Grid/Collection stitching", "" + \
        """ "type=[Filename defined position] order=[Defined by
filename
        ] "" + \
        f"""grid_size_x={(max_y-min_y) + 1} grid_size_y={(max_x -
min_x) + 1} "" + \
        f"""tile overlap=15 first file index x={min_y} "" + \
        f"""first file index y={min_x}
directory={os.path.abspath("./temp")} "" + \
        """file_names=nem_{x}_{y}.tif
output_textfile_name=TileConfiguration.txt "" + \
        """fusion_method=[Linear Blending] regression_threshold=0.30
max/avg_displacement_threshold=2.50 "" + \
        """absolute_displacement_threshold=3.50 compute_overlap "" + \
        """computation_parameters=[Save memory (but be slower)] "" + \
        """image_output=[Fuse and display]"; "" + \
        f"""saveAs("PNG",
"{os.path.abspath("./")}/Fused_{region_idx}.png");"" + \
        """close();"""
        with open("./temp/macro.ijm", "w") as handle:

```

```

        handle.write(fiji_macro)
    cmd = '/Applications/Fiji.app/Contents/MacOS/ImageJ-macosx --
headless ' +\
        f'-macro {os.path.abspath("./temp")}/macro.ijm'
    os.system(cmd)

```

## Apêndice 9

```

from stitching import Image_Matrix
from stitching import Stitcher

if __name__ == "__main__":
    model_nematodes.stitch()

```

## Apêndice 10

```

from measure_nematodes import Measure

if __name__ == "__main__":
    measure = Measure()
    data = measure.find_and_measure_all()

```

## Apêndice 11

```

import tensorflow as tf
from tensorflow.keras.layers import Conv2D, Dropout, MaxPooling2D,
Lambda, Input
from tensorflow.keras.layers import Conv2DTranspose, concatenate
from tensorflow.keras import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping,
TensorBoard, ReduceLROnPlateau
import os
import random
import numpy as np
from tqdm import tqdm
from skimage.io import imread, imshow, imsave
from skimage.transform import resize
import matplotlib.pyplot as plt

data_gen_args = dict(rescale = 1./255,
                    rotation_range=90,
                    shear_range = 0.2,
                    horizontal_flip = True,
                    vertical_flip = True,
                    width_shift_range = [-20,20],
                    height_shift_range = [-20, 20],
                    brightness_range=[0.2,1.0],
                    zoom_range=[0.5,1.0])

mask_data_gen_args = dict(rescale = 1./255,
                        rotation_range=90,

```

```

        shear_range = 0.2,
        horizontal_flip = True,
        vertical_flip = True,
        width_shift_range = [-20,20],
        height_shift_range = [-20, 20],
        brightness_range=[0.2,1.0],
        zoom_range=[0.5,1.0],
        preprocessing_function = lambda x: np.where(x>0,
1, 0).astype(x.dtype))

train_datagen = ImageDataGenerator(**data_gen_args)
valid_datagen = ImageDataGenerator(mask_data_gen_args, rescale=1./255)
seed = 1
batch_size = 10
target_size = (448, 448)
training_path = "/path/training/"
valid_path = "path/test/"
train_image_generator = train_datagen.flow_from_directory(
    training_path + "images",
    class_mode=None,
    color_mode='grayscale',
    target_size = target_size,
    batch_size = batch_size,
    seed = seed)

train_mask_generator = train_datagen.flow_from_directory(
    training_path + "masks",
    class_mode=None,
    color_mode='grayscale',
    target_size = target_size,
    batch_size = batch_size,
    seed = seed)

valid_image_generator = valid_datagen.flow_from_directory(
    valid_path + "images",
    class_mode=None,
    color_mode='grayscale',
    target_size = target_size,
    batch_size = batch_size,
    seed = seed)

valid_mask_generator = valid_datagen.flow_from_directory(
    valid_path + "masks",
    color_mode='grayscale',
    class_mode=None,
    target_size = target_size,
    batch_size = batch_size,
    seed = seed)

train_generator = zip(train_image_generator, train_mask_generator)
train_generator = (pair for pair in train_generator)
val_generator = zip(valid_image_generator, valid_mask_generator)
val_generator = (pair for pair in val_generator)

IMG_HEIGHT = 448
IMG_WIDTH = 448
IMG_CHANNELS = 1

kw = dict(activation='relu', kernel_initializer='he_normal',
padding='same')

```

```

inputs = Input((IMG_HEIGHT, IMG_WIDTH, IMG_CHANNELS))
# s = Lambda(lambda x: x / 255)(inputs)

# Contraction path
c1 = Conv2D(32, (3, 3), **kw)(inputs)
c1 = Dropout(0.1)(c1)
c1 = Conv2D(16, (3, 3), **kw)(c1)
p1 = MaxPooling2D((2, 2))(c1)

c2 = Conv2D(64, (3, 3), **kw)(p1)
c2 = Dropout(0.1)(c2)
c2 = Conv2D(32, (3, 3), **kw)(c2)
p2 = MaxPooling2D((2, 2))(c2)

c3 = Conv2D(128, (3, 3), **kw)(p2)
c3 = Dropout(0.2)(c3)
c3 = Conv2D(64, (3, 3), **kw)(c3)
p3 = MaxPooling2D((2, 2))(c3)

c4 = Conv2D(256, (3, 3), **kw)(p3)
c4 = Dropout(0.2)(c4)
c4 = Conv2D(128, (3, 3), **kw)(c4)
p4 = MaxPooling2D(pool_size=(2, 2))(c4)

c5 = Conv2D(512, (3, 3), **kw)(p4)
c5 = Dropout(0.3)(c5)
c5 = Conv2D(256, (3, 3), **kw)(c5)

kw_conv2transp = dict(strides=(2, 2), padding='same')
u6 = Conv2DTranspose(256, (2, 2), **kw_conv2transp)(c5)
u6 = concatenate([u6, c4])
c6 = Conv2D(256, (3, 3), **kw)(u6)
c6 = Dropout(0.2)(c6)
c6 = Conv2D(256, (3, 3), **kw)(c6)

u7 = Conv2DTranspose(64, (2, 2), **kw_conv2transp)(c6)
u7 = concatenate([u7, c3])
c7 = Conv2D(64, (3, 3), **kw)(u7)
c7 = Dropout(0.2)(c7)
c7 = Conv2D(64, (3, 3), **kw)(c7)

u8 = Conv2DTranspose(128, (2, 2), **kw_conv2transp)(c7)
u8 = concatenate([u8, c2])
c8 = Conv2D(128, (3, 3), **kw)(u8)
c8 = Dropout(0.1)(c8)
c8 = Conv2D(128, (3, 3), **kw)(c8)

u9 = Conv2DTranspose(64, (2, 2), **kw_conv2transp)(c8)
u9 = concatenate([u9, c1], axis=3)
c9 = Conv2D(64, (3, 3), **kw)(u9)
c9 = Dropout(0.1)(c9)
c9 = Conv2D(64, (3, 3), **kw)(c9)

outputs = Conv2D(1, (1, 1), activation='sigmoid')(c9)

model = Model(inputs=[inputs], outputs=[outputs])
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])
model.summary()

callbacks = [ModelCheckpoint(training_path + 'nome do modelo.h5',

```

```
verbose=1, save_best_only=True),
    EarlyStopping(patience=3, monitor='val_loss'),
    ReduceLROnPlateau(monitor='val_loss', factor=.5,
patience=3, min_lr=1e-6)]

STEP_SIZE=2000

results = model.fit_generator(generator = train_generator,
                             #class_weight=weight_dict,
                             validation_data = val_generator,
                             steps_per_epoch = STEP_SIZE,
                             validation_steps = STEP_SIZE,
                             epochs=50,
                             use_multiprocessing=True,
                             callbacks=callbacks,
                             verbose = 1)
```