Universidade Estadual de Campinas
Instituto de Computação

Edson Riberto Bollis

Métodos Fracamente Supervisionados para Classificação de Ácaros na Citricultura

CAMPINAS
2022

Edson Riberto Bollis

Métodos Fracamente Supervisionados para Classificação de Ácaros na Citricultura

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Orientadora: Profa. Dra. Sandra Eliza Fontes de Avila
Coorientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Tese defendida por Edson Riberto Bollis e orientada pela Profa. Dra. Sandra Eliza Fontes de Avila.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Bollis, Edson Ribeiro, 1985-
B638m Métodos fracamente supervisionados para classificação de ácaros na citricultura / Edson Riberto Bollis. – Campinas, SP : [s.n.], 2022.

Orientador: Sandra Eliza Fontes de Avila.
Coorientador: Hélio Pedrini.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado fracamente supervisionado. 2. Redes neurais profundas. 3. Modelos baseados em atenção. 4. Ácaro. 5. Agronomia. I. Avila, Sandra Eliza Fontes de, 1982-. II. Pedrini, Hélio, 1963-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Weakly supervised methods for citrus mite classification
Palavras-chave em inglês:
Weakly supervised learning
Deep neural networks
Attention-based models
Mites
Agronomy
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Sandra Eliza Fontes de Avila [Orientador]
Filipe de Oliveira Costa
Thiago Teixeira Santos
Esther Luna Colombini
Marcelo da Silva Reis
Data de defesa: 29-06-2022
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-0993-784X
- Currículo Lattes do autor: http://lattes.cnpq.br/4343524372032105

Universidade Estadual de Campinas
Instituto de Computação

Edson Riberto Bollis

Métodos Fracamente Supervisionados para Classificação de Ácaros na Citricultura

Banca Examinadora:

- Profa. Dra. Sandra Eliza Fontes de Avila
  IC/UNICAMP
- Dr. Filipe de Oliveira Costa
  CPqD
- Dr. Thiago Teixeira Santos
  EMBRAPA
- Profa. Dra. Esther Luna Colombini
  IC/UNICAMP
- Prof. Dr. Marcelo da Silva Reis
  IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 29 de junho de 2022

Agradecimentos

Agradeço aos meus orientadores, Sandra e Hélio, pelo trabalho desenvolvido e pelos ensinamentos e conhecimentos providos durante o tempo de doutorado. Agradeço, também, à minha noiva, Ana Paula, por me ajudar com o suporte emocional e psicológico que necessitei durante esse período. Agradeço a minha mãe, Silvia, ao meu pai, Riberto, e a outros membros de minha família, pois eles me ensinaram que sem educação não somos nada.

Agradeço aos que contribuíram durante o desenvolvimento inicial de minha tese na Fazenda São José, o administrador da área agronômica André, o gerente do manejo Guilherme, o técnico agronômico Cristiano e as inspetoras Sandra, Marcia e Denise.

Agradeço aos estudantes do Recod.ai pelas discussões e conhecimentos compartilhados, Alceu, Akari, Victor, Antônio, Ramon, Áurea, Manuel[2], Rafael[2], José, Luis, Giuliano e João Felipe, que sempre estiveram disponíveis para discutirmos.

Por fim, agradeço a todos que conheço e que me incentivaram neste trabalho, inclusive aos que já não estão mais presentes neste mundo físico.

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

Resumo

Pragas da agricultura de citros são fatores relevantes da perda de produção e, por isso, o motivo de grande investimento em prevenção de seus agentes causadores. No Brasil, o Manejo Integrado de Pragas (MIP) é o método mais utilizado para prevenir e amenizar os estragos causados pelas pragas e pelos vetores de doenças nas lavouras. No entanto, seus resultados são propensos a erros humanos, pois são executados por trabalhadores que vão aos pomares para identificar visualmente sintomas de doenças, insetos e ácaros. Nesta tese, foram propostos métodos de aprendizado fracamente supervisionados e aprendizado profundo para ajudar a automatização da identificação de ameaças presentes no MIP. Para isso, a base de dados *Citrus Pest Benchmark* foi coletada utilizando dispositivos móveis e lupas para contemplar os requisitos necessários do reconhecimento das pragas mais relevantes do interior paulista invisíveis a olho nu. Os métodos propostos para classificação binária, chamados *MIL-Guided* e *Attention-based MIL-Guided*, utilizam dois tipos de abordagens fracamente supervisionadas: a primeira abordagem, aprendizado por múltiplas instâncias, é guiada pela segunda, mapas de saliências criados por mapas de ativação. Os dois métodos mostraram que são capazes de reconhecer e classificar imagens com regiões de interesse muito pequenas relativamente ao tamanho total da área capturada, como as imagens da base coletada. Além disso, uma nova formulação matemática, chamada *Two-WAM*, referente aos mapas de ativação baseados em atenção foi proposta e utilizada, produziu localizações fracamente supervisionadas capazes de chegar ao nível de métodos totalmente supervisionados. A atenção, como foi proposta, ajudou no processo de encontrar regiões muito pequenas, como as dos ácaros. Para o problema de diferenciar entre várias espécies de ácaros (classificação multirrótulos), os experimentos mostraram que o uso do rótulo referente à classe negativa influenciou negativamente na geração e treinamento de modelos. Por outro lado, o uso da adaptação de domínio não supervisionada, conjuntamente com os mapas de ativação baseados em atenção, teve um impacto positivo nos resultados de classificação de ácaros. Também, foram executados experimentos com outra base de dados chamada IP102, que contém imagens de pragas grandes e salientes. Esses experimentos avaliaram a classificação de diferentes tipos de insetos e ácaros e quantificaram as localizações fracamente supervisionadas. As propostas desta tese, além de produzirem ótimos resultados para pragas pequenas e minúsculas, sempre estiveram ao mesmo nível que os melhores classificadores para a IP102, porém com uma rede mais leve.

Abstract

Pests in citrus yields are relevant factors in production losses and, therefore, the reason for significant investments in preventing their threats. In Brazil, Integrated Pest Management (IPM) is the most used method to prevent and mitigate the damage caused by pests and disease vectors in crops. However, their results are prone to human error because workers perform the management by going to orchards to identify symptoms of diseases, insects, and mites visually. This thesis proposes weakly supervised learning and deep learning methods to help automate threat identification present in IPM. For this, the *Citrus Pest Benchmark* dataset was collected using mobile devices and magnifying glasses to contemplate the requirements for classifying the most relevant pests in the São Paulo countryside that are invisible to the naked eye. The proposed methods for binary classification, called *MIL-Guided* and *Attention-based MIL-Guided*, use two types of weakly supervised approaches: the first approach, multi-instance learning, is guided by the second, saliency maps created by activation maps. Both methods showed that they could recognize and classify images with proportionally tiny regions of interest, such as the images of the collected dataset. Furthermore, a new mathematical formulation, called *Two-WAM*, referring to attention-based activation maps, was proposed and used, producing weakly supervised locations capable of reaching the level of fully supervised methods. Attention, as proposed, helped in finding tiny regions, such as mites. To distinguish between various mite species (multi-label classification), the experiments showed that the label referring to the negative class negatively influenced the models' training. On the other hand, unsupervised domain adaptation, in conjunction with attention-based activation maps, positively impacted mite classification results. Moreover, experiments were conducted with another database called IP102, which contains images of salient pests. These experiments evaluated the classification of different types of insects and mites and quantified the weakly supervised locations. In addition to producing excellent results for tiny pests, these thesis proposals have always been at the same level as the best classifiers for IP102 but with a smaller deep neural network.

Lista de Figuras

Lista de Tabelas

Lista de Abreviações e Siglas

ADNS	adaptação de domínio não supervisionada
Attention-based MIL-Guided	método fracamente supervisionado de aprendizado por múltiplas instâncias baseado em atenção e guiado por mapas de saliências (*attention-based weakly supervised multiple instance learning guided by saliency maps*)
BIDAF	rede de fluxo de atenção bi-direcional (*bi-directional attention flow*)
CAM	mapas de ativação por classes (*class activation map*)
CPB	base de dados de referência para pragas dos citros *Citrus Pest Benchmark*
DCML	aprendizado baseado em métrica usando divisão e conquista (*divide and conquer metric learning*
DD	densidade diversa (*diversity density*)
DMF-ResNet	rede residual profunda de fusão de múltiplos ramos (*deep multi-branch fusion residual network*)
DMIL-WDDS	sistema de diagnóstico de doenças do milho baseado em múltiplas instâncias (*multiple instance learning based wheat disease diagnosis system*)
DSMIL	aprendizado por múltiplas instâncias com fluxo duplo (*dual-stream multiple instance learning*)
Embrapa	Empresa Brasileira de Agropecuária
EM	maximização de expectativa (*expectation-maximization*)
EM-DD	maximização de expectativa de densidade diversa (*expectation-maximization - diversity density*)
FPN	redes de pirâmides de mapas de características (*feature pyramid networks*)
FR-ResNet	rede residual de reúso de características (*feature reuse residual network*)
GAEnsemble	*ensemble* com pesos produzidos por algoritmos genéticos (*genetic algorithm ensemble*)
GAN	redes adversárias generativas (*generative adversarial networks*)
GAT	redes de atenção baseadas em grafos (*graph attention networks*)
GHCID	base de detecção de gafanhotos (*GrassHopper deteCtIon Dataset*)
GLAM	mapas de ativações locais e globais (*global-local activation maps*)
GMIC	classificador de múltiplas instâncias com reconhecimento global (*globally-aware multiple instance classifier*)
Grad-CAM	mapas de ativação por classes baseados nos gradientes (*gradient-based activation map*)
HLB	Huanglongbing ou Greening
KNN	K-vizinhos mais próximos (*K-nearest neighbors*)
IAM	mapa de ativação de instâncias (*instance activation mapping*)
i.i.d.	variáveis aleatórias independentes e identicamente distribuídas *independent and identically distributed*)

IoT	internet das coisas (*Internet of Things*)
IP102	pragas de insetos com 102 classes (*insect pest 102*)
IPFC	pragas das culturas alimentícias do campo (*in-field pest in food crop*)
LKA	atenção de núcleo grande (*large kernel attention*)
LSTM	rede recorrente bidirecional de memória curta e longa (*long short-term memory*)
MIDCN	rede convolucional profunda de múltiplas instâncias (*multiple instances deep convolutional network*)
MIL	aprendizado por múltiplas instâncias (*multiple instance learning*)
MILCNN	rede neural convolucional de aprendizado por múltiplas instâncias (*multiple instance learning convolutional neural network*)
MIL-Guided	método fracamente supervisionado de aprendizado por múltiplas instâncias guiado por mapas de saliências (*weakly supervised multiple instance learning guided by saliency maps*)
MIP	manejo integrado de pragas
mi-SVM	formulação do padrão de margem máxima do MIL (*maximum pattern margin formulation of MIL*)
MI-SVM	formulação da margem máxima de *bags* para MIL (*maximum bag margin formulation of MIL*)
MMAL-Net	rede de atenção multirramos e multiescalas (*multi-branch and multi-scale attention networks*)
NCPB	base de dados de referência para as pragas dos citros construída sem ruídos (*Noiseless Citrus Pest Benchmark*)
NLP	processamento de linguagem natural (*natural language processing*)
NT	transformadores neurais (*neural transformers*)
Patch-SaliMap	algoritmo de seleção de múltiplos *patches* baseado em mapas de saliências (*multi-patch selection strategy based on saliency maps*)
PCA	análise de componentes principais (*principal component analysis*)
PDD271	base de doenças de plantas com 271 classes (*plant disease dataset 271*)
RAM	modelo de atenção recorrente (*recurrent attention model*)
RAN	rede de atenção residual (*residual attention networks*)
RI	regiões de interesse
RNNSearch	busca com redes neurais recorrentes (*recurrent neural network search*)
RPN	redes de proposição de regiões (*region proposal network*)
SACNN	rede neural convolucional de autoatenção (*self-attention convolutional neural network*)
SA-CAM	mapeamento de ativação de classes de atenção espacial (*spatial attention class activation map*)
Score-CAM	mapeamento de ativação de classes por pontuações (*score class activation map*)
SE	camadas de atenção por compressão e excitamento (*squeeze-and-excitation*)
SMPEnsemble	*ensenble* da soma das máximas probabilidades (*sum of maximum probabilities ensemble*)
SP-CAM	mapeamento de ativação de classes para o agrupamento de superpixels (*superpixel-pooled class activation map*)
SSD	detector de disparo único (*single shot detector*)
STN	rede de transformadores espacial (*spatial transformer network*)
SVM	máquinas de vetores de suporte (*support vector machines*)

T-CAM	mapeamento de ativação de classes pelo tempo (*temporal class activation map*)
Two-WAM	mapas de ativação com dois pesos (*Two-Weighted Activation Mapping*)
VAN	rede de atenção visual (*visual attention network*)
ViT	transformadores visuais (*visual transformers*)
WILDCAT	aprendizado fracamente supervisionado de redes neurais convolucionais profundas (*Weakly supervIsed Learning of Deep Convolutional neurAl neTworks*)

Sumário

Capítulo 1

Introdução

Este capítulo descreve os principais tópicos que serão abordados durante o decorrer deste texto. A Seção 1.1 especifica os principais problemas relativos ao tema de classificação de pragas da lavoura de citros. A Seção 1.2 apresenta as motivações e os desafios relacionados ao uso de redes neurais nesse domínio. As Seções 1.3, 1.4 e 1.5 descrevem, respectivamente, os objetivos, as questões de pesquisa e as contribuições do trabalho que se relacionam aos avanços científicos para a computação e seus benefícios para a área de citricultura. As publicações geradas durante o doutorado são listadas na Seção 1.6. Finalmente, a organização do texto desta tese de doutorado é apresentada na Seção 1.7.

1.1 Descrição do Problema

As pragas da lavoura e os vetores de doenças são extremamente danosos para a agricultura mundial e causam perdas significativas. Por exemplo, o inseto psilídeo (*Diaphorina citri*) que propaga a doença *Greening* (*Diaphorina citri*) — doença mais importante e destrutiva da citricultura [65] —, também conhecida como *Huanglongbing* (HLB), causou perdas de 13,2 bilhões de dólares na Flórida entre 2005 e 2016 [77] e, no Brasil, mais precisamente em São Paulo e Minas Gerais, a perda da produção em 2017 foi de 16,7%, o que corresponde a 32 milhões de plantas [17]. As perdas reais são muito maiores ao se considerar as plantas infectadas por outras doenças disseminadas por insetos e ácaros, como: a Clorose Variegada dos Citros (*Xylella fastidiosa*) com 2,6% da infecção; Cancro Cítrico (*Xanthomonas axonopodis*) com 12,9% da infecção [17]; e a Leprose (*Citrus leprosis vírus*) com 10,9% de infecção [48].

Os insetos e ácaros causadores de danos, denominados de *pragas*, e os vetores dispersores de doenças são normalmente muito pequenos e extremamente difíceis de serem classificados, como é o caso do ácaro da ferrugem que mede em média 0,3 mm de largura e 0,7 mm de comprimento [51] (a Figura 1.1d exemplifica o tamanho de um dos maiores ácaro sem aumento óptico). Uma das maneiras, incentivada pela Empresa Brasileira de Pesquisa Agropecuária (Embrapa) [159], para encontrar as pragas e doenças é pela inspeção visual descrita no *Manejo Integrado de Pragas* (MIP) [66] exemplificado na Figura 1.1, que é realizado pelas inspetoras de pragas (Figura 1.1a), também conhecidas como pragueiras. Geralmente, nessa profissão, as mulheres se destacam por serem mais atentas e distinguir melhor entre diferentes cores [164, 181]. As inspetoras andam pelos pomares de citros colhendo amostras de árvores (Figura 1.1c), verificando-as com uma lupa (Figura 1.1d) e descrevendo os resultados de suas análises em

(a) Inspetora de pragas (pragueira) (b) Lupa utilizada hoje. (c) Indivíduo da coleta (d) Ácaro em uma folha.

Figura 1.1: Manejo Integrado de Pragas como é realizado atualmente.

fichas ou sistemas de software especializados para dispositivos móveis [159]. No entanto, como esperado, esse processo é propenso a falhas, uma vez que o cansaço das inspetoras ao final do dia influencia o trabalho de análise e classificação.

A utilização de dispositivos móveis empregados em várias funções tem se tornado comum na agricultura, como a coleta de dados em campo, o controle de funcionários nas lavouras e o pedido e lançamento de notas de maquinário agrícola. Diante desse cenário, a utilização dos dispositivos móveis, como máquinas para adquirir imagens de ameaças e auxiliar sua classificação, é um caminho natural, principalmente por causa do advento dos métodos de *aprendizado profundo* (*deep learning*) [95]. Portanto, a matéria prima para automatização da classificação de pragas descritas no MIP é fácil de se encontrar em campo, entretanto, há outros desafios e necessidades para um projeto real que usa o aprendizado profundo. Dentre esses desafios, pode-se citar o tempo dispendido com a ida a campo para coleta de imagens, o número elevado de imagens representativas do domínio para treinar os algoritmos, a disponibilidade do tempo e conhecimento de especialistas para rotulação das imagens e, no caso específico do domínio da classificação de pragas, a obtenção de imagens com granularidade fina das características relativas às *regiões de interesse* (RI) — regiões onde os objetos de interesse realmente estão — muito pequenas. Sem uma forma prática de lidar com estes problemas, torna-se impraticável o uso do aprendizado profundo em campo.

Para lidar com esses desafios, torna-se necessário o uso de uma abordagem que: (i) economize o tempo de rotulação das imagens, pois a anotação de bases é dispendiosa; (ii) aumente o número de imagens para a etapa de treinamento, diminuindo o tempo de coleta em campo; e (iii) maximize a capacidade de extrair características de pequenas regiões não uniformes, reduzindo a necessidade de geração de rótulos e obtenção de imagens com RI centralizadas e visualmente grandes ou salientes. Nesse caso, o uso de abordagens *fracamente supervisionadas* foi proposto nesta tese, pois métodos de *aprendizado por múltiplas instancias* (*multiple instance learning*, MIL) utilizam múltiplos recortes ou, como serão chamados, *patches* — regiões das imagens contendo ou não objetos de interesse — sem a necessidade de rotulação manual, o

que aumenta o número de imagens independentemente do tempo de especialistas. Além disso, os métodos de *mapas de saliências* fracamente supervisionados, como os *mapas de ativação*, são capazes de encontrar as localizações das RI com um certo grau de certeza sem o uso ativo de especialistas. Adicionalmente, modelos ou abordagens baseadas em atenção — abordagens que incorporam a noção de relevância de características [30] — aumentam a influência desse tipo de regiões no processo de classificação.

1.2 Motivações e Desafios

No contexto da automatização do MIP ou automatização da classificação de pragas — termos designados para automatização da classificação de pragas e vetores de doenças da lavoura de citros descrita pelo MIP —, esta pesquisa apresentou grandes desafios devido à carência de dados anotados. A falta de bases de dados específicas para esse contexto foi o primeiro desafio e um dos principais motivos para diversas aplicações de aprendizado profundo não serem bem sucedidas diretamente em campo [13]. No caso da análise e classificação de pragas, quando os trabalhos desta tese foram iniciados, as bases públicas eram escassas e, mesmo as existentes atualmente, não satisfazem as necessidades específicas das pragas da agricultura citrícola paulista, pois muitas das pragas são regionais. Inclusive, muitas dessas bases foram coletadas de fontes diversas disponíveis na Internet. Por exemplo, a PlantVillage [81] é uma base com cerca de 55 mil imagens de sintomas de doenças e plantas variadas (uma exceção em relação ao número de imagens) e contém, para os citros, somente imagens coletadas em laboratório da doença *Greening*. A base IP102 [197] contém diversas imagens de pragas, entre elas, várias relacionadas aos citros, mas suas imagens foram coletadas da Internet e não contemplam todas as pragas necessárias para a execução deste trabalho.

Outro desafio importante é a inadequação das técnicas existentes de aprendizado profundo para extrair características de áreas muito pequenas durante o decorrer do processo de classificação de imagens [102, 139, 190]. As arquiteturas são normalmente propostas para tarefas gerais e de ponta a ponta — treinadas de forma única e monolítica —, pois os extratores de características, na maior parte das redes, são construídos com *downsampling* (reduções), que descartam características mais finas em detrimento de dados semanticamente mais gerais. Isso diminui a probabilidade de classificação correta quando as RI são muito pequenas em relação ao tamanho total das imagens. Então, é aceitável pensar que existe uma proporção espacial mínima para que as características de objetos sejam classificadas corretamente. Métodos de geração de múltiplos *patches* — métodos de criação de novas imagens a partir de um recorte das imagens originais —, aliados ao MIL, possibilitam as condições necessárias para o aprendizado profundo ser aplicado neste contexto. Porém, aumentam o processamento das redes devido à inferência de várias imagens no lugar da imagem original.

Além dos problemas inerentes às redes neurais e RI muito pequenas, as condições de coleta presentes em campo afetam a produção de imagens e influenciam o aparecimento de ruídos tais como borramento e luminosidade. Esses ruídos são naturais (intrínsecos) ao trabalho de campo e, em uma aplicação, sempre estarão presentes na produção. Portanto, torna-se necessário lidar com imagens ruidosas no treinamento de redes neurais profundas para entender se seu uso influencia o aprendizado. Além disso, deve-se compreender qual o impacto dos ruídos quando as RI são muito pequenas.

As redes de aprendizado profundo de propósito geral para o MIL são escassas, principalmente por utilizarem premissas relativas aos domínios específicos de seu uso, como ocorre em estudos na medicina [38]. Pelo fato do MIL trabalhar com supervisão inexata — tipo de supervisão em que há um grau de rotulação, entretanto, não suficiente para a tarefa desejada (Seção 2.4) —, nota-se, na maior parte dos trabalhos disponíveis, a necessidade de adaptar as arquiteturas de aprendizado profundo conhecidas. Essa adaptação é necessária porque muitos *patches* não recebem a rotulação correta e o aprendizado profundo requer rótulos corretos das imagens para modificar seus parâmetros. Assim, torna-se necessário o uso de métodos de aprendizado de máquina que guiem a produção de *patches* para diminuir os erros de rotulação, como os mapas de ativação.

Os mapas de ativação têm a propriedade de inferir localizações mesmo sem o treinamento com rótulos para esse fim, chamados retângulos delimitadores (*bounding boxes*). Porém, os métodos de geração de mapas de ativação, por serem fracamente supervisionados, trazem outros desafios, tais como a inexatidão das saliências geradas e apenas o realce das características mais discriminativas das RI. Por exemplo, quando uma imagem contém um grupo de objetos com a mesma classe, os mapas de ativação podem apontar uma região entre esses objetos e, quando uma região contém um objeto como, por exemplo, um boi, os mapas de ativação podem salientar somente a cabeça e os chifres como sendo as regiões mais discriminativas para sua classificação. Há um senso comum de que aprimorando a classificação melhora-se a localização, entretanto, as características que discriminam uma classe podem não ser as mais apropriadas para inferir as localizações dos objetos dessa classe.

1.3 Objetivos

Os principais objetivos deste trabalho de pesquisa são:

O1. Criar uma base de dados com diferentes tipos de ácaros da citricultura paulista;

O2. Verificar a importância do tamanho das regiões de interesse nas imagens;

O3. Avaliar a integração dos métodos de múltiplas instâncias e mapas de ativação;

O4. Melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas;

O5. Investigar a viabilidade do uso de métodos fracamente supervisionados na localização de pragas em imagens.

1.4 Questões de Pesquisa

As principais questões de pesquisa que esta tese visa responder são:

Q1. Métodos de aprendizado fracamente supervisionados por múltiplas instâncias são eficazes para a classificação de pragas da citricultura em regiões pequenas?

Q2. Métodos de aprendizado fracamente supervisionados de mapas de ativação são eficazes para a localização de pragas?

Q3. É possível desenvolver uma arquitetura ponta a ponta para gerar mapas de ativação que selecionem múltiplas regiões de interesse para a classificação automática de pragas?

Q4. Ruídos, tais como luminosidade e borramento presentes na captura de imagens em campo, afetam o treinamento dos modelos de redes neurais profundas?

Q5. Uma localização fracamente supervisionada mais acurada das pragas em um modelo causa uma maior taxa de classificação?

1.5 Contribuições

A automatização da classificação de pragas tem grande potencial para melhorar a produção agrícola e o gerenciamento do processo como um todo. As principais contribuições deste trabalho para a área da computação são:

C1. Um novo conjunto de dados de referência (*benchmark*) para o problema do reconhecimento de pragas dos citros, em que pequenas regiões de interesse contendo diferentes tipos de ácaros estão presentes nas imagens. O conjunto de dados, denominado *Citrus Pest Benchmark* (CPB), está disponível em https://github.com/edsonbollis/Citrus-Pest-Benchmark;

C2. Métodos fracamente supervisionados para classificação e localização de pragas. Os métodos, denominados *MIL-Guided* e *Attention-based MIL-Guided*, estão disponíveis em https://github.com/edsonbollis/Weakly-Supervised-Learning-Citrus-Pest-Benchmark e https://github.com/edsonbollis/Attention-based-MIL-Guided, respectivamente;

C3. Uma nova formulação matemática, denominada *Two-Weighted Activation Mapping* (Two-WAM), que utiliza dois pesos de treinamento para criação de mapas de ativação baseados em atenção;

C4. Uma estratégia eficaz de seleção de múltiplas regiões de interesse, denominada *Multi-patch Selection Strategy based on Saliency Maps* (Patch-SaliMap), baseadas em mapas de saliências para localizar automaticamente pragas dos citros;

C5. Resultados competitivos em relação aos métodos disponíveis na literatura para duas bases de dados desafiadoras (CPB e IP102).

1.6 Publicações

As seguintes publicações em conferências e periódicos foram produzidas diretamente no contexto desta tese de doutorado:

- E. Bollis, H. Pedrini e S. Avila. *Weakly Supervised Learning Guided by Activation Mapping Applied to a Novel Citrus Pest Benchmark*. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, p. 310–319,

2020. DOI: https://doi.org/10.1109/CVPRW50498.2020.00043. *h5-index*: 89.

- E. Bollis, H. Maia, H. Pedrini e S. Avila. *Weakly Supervised Attention-based Models using Activation Maps for Citrus Mite and Insect Pest Classification*. Computers and Electronics in Agriculture, volume 195, 2022. DOI: https://doi.org/10.1016/j.compag.2022.106839. Fator de impacto: 6,757.

A seguinte publicação foi produzida em colaboração com outro projeto de pesquisa durante o período de doutorado:

- A. Ishikawa, E. Bollis e S. Avila. *Combating the Elsagate Phenomenon: Deep Learning Architectures for Disturbing Cartoons*. 7th International Workshop on Biometrics and Forensics (IWBF), Cancun, Mexico, p. 1–6, 2019. DOI: https://doi.org/10.1109/IWBF.2019.8739202. *h5-index*: 15.

1.7 Organização do Texto

Os capítulos subsequentes deste texto estão organizados como segue. O Capítulo 2 descreve os principais conceitos relacionados às pragas, às doenças e aos vetores de doenças, ao Manejo Integrado de Pragas e aos métodos de aprendizado fracamente, semi e totalmente supervisionados. O Capítulo 3 revisa a literatura relativa ao aprendizado por múltiplas instâncias, mapas de ativação, métodos baseados em atenção e trabalhos de agricultura relacionados com o aprendizado profundo. O Capítulo 4 descreve as principais características da base coletada, denominada *Citrus Pest Benchmark*. O Capítulo 5 apresenta e discute as abordagens propostas nesta tese de doutorado e as medidas de avaliação para as redes de aprendizado profundo utilizadas. O Capítulo 6 reporta e avalia os resultados obtidos a partir dos métodos desenvolvidos. O Capítulo 7 conclui o texto desta tese com comentários finais e sugestões para trabalhos futuros.

Capítulo 2

Conceitos Relacionados

Este capítulo descreve os principais conceitos relacionados à classificação de pragas e doenças em campo (Seção 2.1) e ao manejo integrado de pragas (Seção 2.2), aos tipos de classificação (Seção 2.3) e aos tipos de aprendizado (Seção 2.4).

2.1 Pragas, Doenças e Vetores de Doenças

As pragas são insetos ou ácaros que causam (são o motivo direto de) perdas para a lavoura, ou seja, são a razão principal da perda de produção e do aparecimento de sintomas. As pragas possuem diversos tamanhos, entretanto, algumas são extremamente pequenas e impossíveis de serem vistas a olho nu, tais como os ácaros. Para isso, torna-se necessário o auxílio de aumento ótico (Figura 2.1). Por esse motivo, insetos são explorados em diversos trabalhos sobre classificação automática de pragas [5, 52, 133] ou as pragas são identificadas apenas por seus sintomas como se fossem doenças [60, 132].

Ovo do bicho lixeiro (*Chrysoperla*) | Ácaro predador maçã (*Iphiseiodes zuluagai*) | Ácaro da leprose (lupa 60×) (*Brevipalpus phoenicis*)

Figura 2.1: Insetos e ácaros (pragas) muito pequenos ou invisíveis a olho nu.

As doenças são causadas por fungos, vírus ou bactérias. Como não são identificáveis a olho nu ou com aumento ótico em campo, elas são reconhecidas por seus sintomas característicos (Figura 2.2). As doenças são muito mais devastadoras do que as pragas e, em alguns casos, como o do *Greening*, não têm tratamento e os agricultores somente atenuam os danos causados. O cancro cítrico, por outro lado, tem tratamento, mas seus frutos não são próprios para o mercado consumidor quando os sintomas aparecem. A Figura 2.2 mostra os sintomas tanto do *Greening* quanto do cancro cítrico.

Greening ou *Huanglongbing* (*Diaphorina citri* — HLB)

Cancro cítrico (*Xanthomonas axonopodis*)

Figura 2.2: Sintomas de doenças. Figuras presentes nos trabalhos [49, 50].

Os vetores de doenças são agentes, normalmente insetos ou ácaros, que levam as doenças de uma planta para outra. Isso ocorre quando um inseto ou ácaro está em uma árvore doente e se infecta com o fungo, vírus ou bactéria de uma doença e depois voa ou é carregado pelo vento até uma árvore sadia que, ao final, infecta-se com a doença trazida pelo vetor [67]. Os vetores de doenças são normalmente pragas ou são tratados como pragas para efeito de classificação [197].

2.2 Manejo Integrado de Pragas e a Classificação e Prevenção de Pragas e Vetores de Doenças

O Manejo Integrado de Pragas (MIP), definido pela primeira vez em uma reunião na Itália na década de 1950, foi descrito como um processo que, no ambiente associado à dinâmica das populações de pragas, utiliza técnicas e métodos adequados para manter as pragas em níveis populacionais abaixo dos que causam danos econômicos [66]. Posteriormente, o MIP foi generalizado para considerar vetores de doenças e sintomas iniciais de doenças. Assim, o MIP descreve as diretrizes do processo de análise, gerência e classificação de pragas, doenças e vetores de doenças para a aplicação de insumos e prevenção de perdas.

Segundo o MIP, seu processo reduz os riscos de infestação ao controlar as populações de pragas e vetores de doenças. Para prevenir a Leprose, por exemplo, é necessário verificar se o número de seus ácaros disseminadores aparece em mais do que 2% das frutas de uma unidade de produção e, em caso positivo, é necessária a aplicação de insumos para mantê-los abaixo desse limiar. O MIP parte do princípio de que, se os vetores de uma doença e os insetos ou ácaros de uma praga estiverem abaixo de um limiar, a probabilidade de disseminação e do aparecimento de sintomas será baixa. Se sintomas aparecerem, será necessário efetivar o contingenciamento de danos, porque perdas já estarão ocorrendo. Na Figura 2.3, alguns sintomas visíveis são ilustrados, em que provavelmente perdas já ocorreram.

O processo de manejo é realizado por amostragem estatística, ou seja, a inspetora de pragas vai a campo e seleciona árvores para a análise e classificação visual das pragas e sintomas de doenças. Em pomares de citros, uma percentagem de árvores de cada unidade de produção é analisada, dependendo da praga e da doença, e, dessa análise, são calculadas a taxa de infecção média por árvore e a taxa de infecção da unidade de produção, estabelecendo um valor relativo ao limiar proposto pelo MIP [159].

Ferrugem
(*Polyphagotarsonemus latus*)

Larva minadora dos citros
(*Phyllocnistis citrella*)

Figura 2.3: Sintomas de Pragas. Figuras modificadas do trabalho de Santos Filho et al. [159].

No caso da Lima Ácida do Tahiti, os especialistas caminham entre as árvores e utilizam diferentes estratégias para selecionar indivíduos para amostragem. Uma das estratégias é a escolha aleatória das árvores, que é feita por meio do andar em "ziguezague" por entre as ruas da unidade de produção. Outra estratégia é a chamada três por trinta, em que a inspetora de pragas analisa uma rua, verificando uma árvore a cada trinta, e desconsidera outras duas ruas. Uma prática comum é iniciar por ruas ou lugares distintos em datas distintas e não avaliar indivíduos presentes nas bordas da unidade.

Os técnicos normalmente inspecionam cada planta observando os frutos, as folhas e o caule com suas lupas e depois descrevem os resultados de suas análises em fichas de papel ou em ferramentas computacionais especializadas para dispositivos móveis. Um padrão é fornecer um valor numérico para o grau de infecção de cada uma das doenças e pragas por planta visitada, cujos valores geralmente variam entre 0 e 2, em que 0 indica ausência total de insetos ou ácaros específicos por praga e área de análise, 1 indica número abaixo de um limiar definido pelo MIP e 2 indica grande número de insetos ou ácaros.

2.3 Classificação Binária, Multiclasses e Multirrótulos

Nesta tese de doutorado, classes são os elementos existentes em um indivíduo de pesquisa que compartilham características semelhantes. Em nosso contexto, os indivíduos de pesquisa são imagens e cada elemento pertencente a uma classe possui uma região de interesse (RI) relacionada. Um rótulo é a anotação de uma classe segundo uma imagem. Cada base de dados composta por imagens pode conter um número finito de classes e cada imagem pode conter um número finito de rótulos.

Uma classificação binária é uma tarefa que relaciona cada imagem a duas classes distintas, uma classe para representar a existência de um elemento e outra para não existência. Matematicamente, uma tarefa de classificação binária é uma função $F : X \rightarrow Y$ que leva uma imagem de um conjunto X qualquer em um conjunto binário $Y = \{0, 1\}$, que é o conjunto de rótulos para as classes, em que 1 indica a existência do elemento desejado e 0 caso contrário. Exemplos de notações vetoriais para representar essas classes podem ser: $[0]$ ou $[1]$, quando é usado apenas um rótulo para representar as duas classes; e $[0, 1]$ ou $[1, 0]$, quando são usados dois rótulos para as representações.

A classificação de múltiplas classes, multiclasses ou multinomial, é a tarefa de classificar

imagens em uma de três ou mais classes, ou seja, a tarefa de gerar $F : X \rightarrow Y$, em que X é um conjunto de imagem qualquer e $Y = \{1, \ldots, n\}$ para $n \in \mathbb{N}$, com $n > 2$ o número total de classes. Um exemplo de notação vetorial para representar essas classes pode ser $[0, 0, 0, 1, 0, \ldots, 0]$, nesse caso, somente um valor 1 aparecerá no vetor inteiro e o índice relativo a esse valor será o índice da classe referente à imagem.

A classificação de múltiplos rótulos ou multirrótulos é a tarefa de classificação em que uma imagem pode ter um ou mais rótulos, o que difere da classificação multiclasses. Matematicamente, uma classificação multirrótulos é a tarefa de encontrar uma função $F : X \rightarrow Y$, em que X é o conjunto de imagens e $Y = \{0, 1\}^n$, onde 1 indica a presença de determinada classe, 0 a sua ausência e $n \in \mathbb{N}$ com $n > 2$ o número de classes. Pode-se ainda definir Y para a tarefa multirrótulos como sendo um conjunto de subconjuntos de $\{1, \ldots, n\}$, ou seja, $Y = \{y, y \subset \{1, \ldots, n\}\}$. Um exemplo de notação vetorial para representação multirrótulos pode ser $[0, 0, 1, 1, 0, \ldots, 1]$. Neste caso, mais de um valor 1 pode aparecer no vetor e os índices relativos a cada valor determinam as classes presentes na imagem. Porém, quando existe uma classe negativa entre os múltiplos rótulos, não é necessário um rótulo específico para representá-la. O vetor cuja ausência de um índice que contenha o valor 1 pode representar a classe negativa: $[0, 0, 0, \ldots, 0]$ ($n - 1$ índices).

2.4 Aprendizado Totalmente Supervisionado, Fracamente Supervisionado e Semissupervisionado

O *aprendizado fracamente supervisionado* é um termo geral que abrange uma variedade de trabalhos, cujo objetivo é construir modelos preditivos através de dados com supervisão fraca ou não totalmente confiável. Tipicamente, há três tipos de supervisão fraca [229]: (i) *supervisão incompleta*, que parte do pressuposto de que dois conjuntos de dados de treinamento são considerados, sendo o primeiro geralmente menor e rotulado, enquanto o segundo não rotulado; (ii) *supervisão imprecisa*, com apenas um conjunto de dados rotulados, entretanto, sem a certeza de que os rótulos foram criados corretamente; (iii) *supervisão inexata*, cujo conjunto de treinamento é único e provido com rótulos de granularidade grossa, por exemplo, o uso de rótulos de classe para segmentação ou localização de objetos em imagens. A Figura 2.4 mostra a relação entre os diferentes tipos de aprendizado.

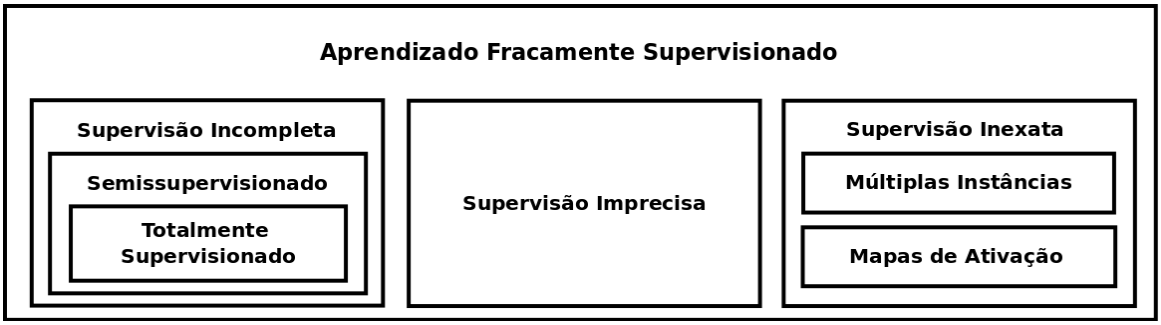


Figura 2.4: Ilustração do aprendizado fracamente supervisionado e seus subtipos de aprendizados para tarefas sem intervenção humana. Figura baseada no trabalho de Zhou [229].

A definição formal de uma tarefa para o aprendizado totalmente supervisionado é uma restrição da definição de aprendizado semissupervisionado, cuja definição recai sobre a definição

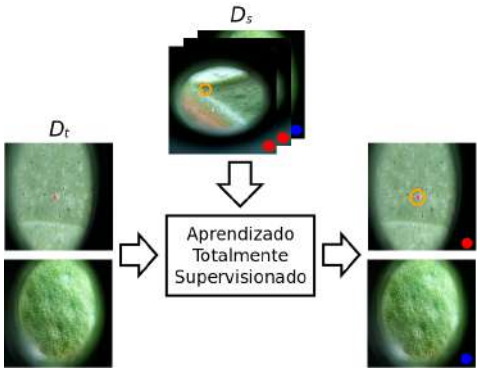
de tarefas para o aprendizado com supervisão incompleta. Assim, pode-se dizer que tanto o aprendizado semissupervisionado quanto o totalmente supervisionado são formas de aprendizado fracamente supervisionado.

No *aprendizado totalmente supervisionado* (Figura 2.5a), usa-se um conjunto de treinamento rotulado para encontrar os melhores parâmetros de uma tarefa (função ou modelo), que será avaliada em um conjunto de teste. Dado um conjunto de dados de treinamento $D_s = \{(x_i, y_i), y_i = f(x_i), x_i \in X, y_i \in Y, i \in \mathbb{N}\}$, em que $X \subset \mathbb{R}^m$ é o espaço m-dimensional de características reais de D_s e f é uma função que mapeia cada elemento de X para o conjunto de rótulos Y, uma tarefa totalmente supervisionada é definida formalmente como uma aproximação da função f por uma função $F : X \rightarrow Y$ que prevê os rótulos para um conjunto previamente escolhido de dados de teste D_t, não conhecidos durante a geração de F. A ação de calcular F é conhecida como treinamento. Exemplos de tarefas totalmente supervisionadas são comuns na literatura, tais como as tarefas de classificação e detecção de objetos na base de dados ImageNet [91] e a tarefa de segmentação da base de dados MS COCO [108].

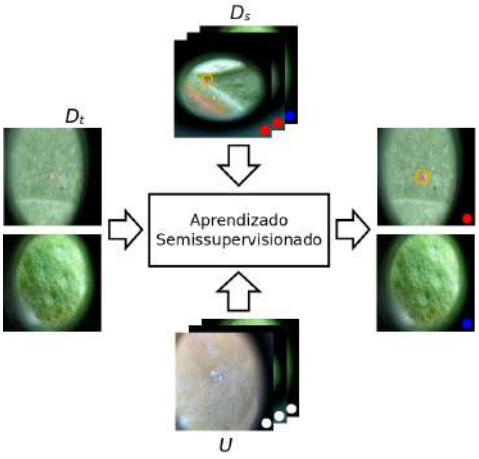
No *aprendizado semissupervisionado* (Figura 2.5b), além dos conjuntos enunciados pelo aprendizado totalmente supervisionado, há um conjunto sem rótulos, utilizado ou não, simultaneamente com o conjunto de treinamento, para melhorar a tarefa avaliada nos testes. Ou seja, dado um conjunto de treinamento D_s, como definido anteriormente, e um conjunto de dados sem rótulos $U = \{u_i, i \in \mathbb{N}\}$ [38], uma tarefa $F : X \rightarrow Y$ é uma aproximação da função f que emprega as características de D_S e U para prever rótulos em D_t. Em um método semissupervisionado, D_t não é conhecido durante o treinamento e U não necessariamente é igual a D_t e nem pode-se admitir que suas distribuições são iguais [229].

Métodos de aprendizado semissupervisionado são baseados em suposições ou conhecimentos pré-adquiridos sobre o domínio dos conjuntos de dados para auxiliar a tomada de decisão das tarefas, associando propriedades da distribuição dos elementos de X com propriedades de F. Entretanto, quando a suposição não procede, há um alto risco de os métodos terem eficácia muito baixa, não compensando seu uso. Exemplos de conhecimento a priori incluem (i) a suposição de proximidade, em que elementos próximos em um espaço de características têm alta probabilidade de pertencerem à mesma classe, (ii) suposição de clusterização, em que exemplos pertencentes a um grupo (*cluster*) devem pertencer à mesma classe e (iii) suposição de densidade, em que áreas com pouca densidade de elementos no espaço de características têm alta probabilidade de serem regiões de fronteira. Em 2020, o uso de um método semissupervisionado, denominado *self-training*, obteve resultados de estado da arte [202] na base de dados ImageNet. Na abordagem *self-training*, o treinamento inicia-se com a adequação de um classificador F com base em D_s, depois aplica-se a inferência desse classificador F nos elementos de U e adiciona-se os elementos de U, com rótulos preditos por F, aos exemplos de treinamento, gerando um novo conjunto de treinamento $D'_s = X \cup \{(u_i, F(u_i)), u_i \in U, i \in \mathbb{N}\}$ para treinar um novo classificador. Há outra abordagem, além da semissupervisionada, para a definição de problema de supervisão incompleta, chamada aprendizado ativo [163], entretanto, diferentemente do aprendizado semissupervisionado, utiliza influência humana direta no momento do treinamento.

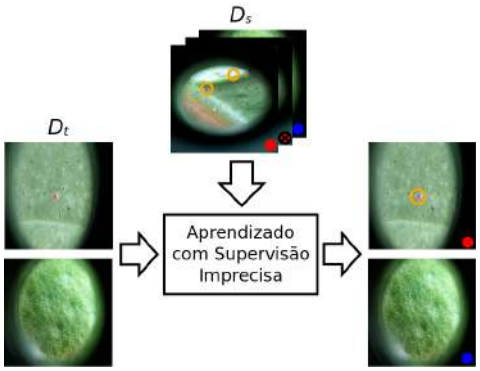
O aprendizado com supervisão imprecisa [229] (Figura 2.5c) considera rótulos erroneamente atribuídos e rótulos não totalmente corretos, ou seja, atribuição de um elemento que não existe e não atribuição de um elemento em que ele está presente. Sua definição formal é similar ao aprendizado totalmente supervisionado, adicionando-se a premissa de que o conjunto D_s



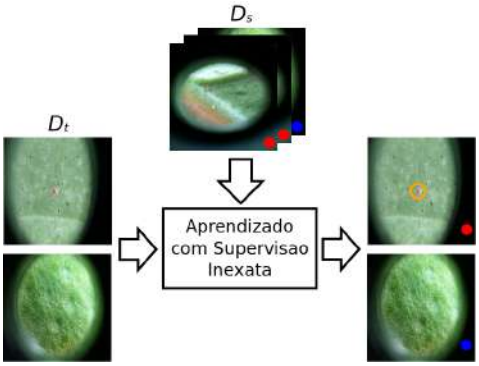
(a) Os círculos vermelhos representam indivíduos da classe positiva, azuis da classe negativa e marcações em laranja representam a localização das RI.



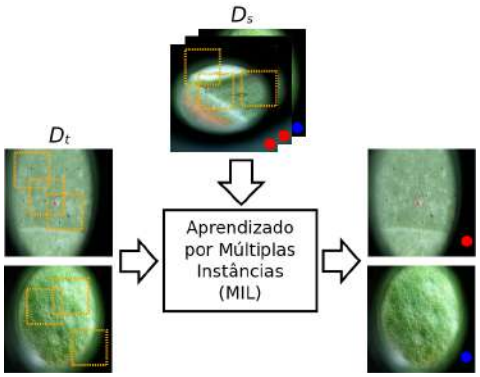
(b) Os círculos vermelhos representam indivíduos da classe positiva, azuis da classe negativa, brancos sem rótulos e marcações em laranja representam a localização das RI.



(c) Os círculos vermelhos representam indivíduos da classe positiva, azuis da classe negativa, com "X" erros de rotulação e marcações em laranja representam as localizações das RI, inclusive incorretas.



(d) Os círculos vermelhos representam imagens da classe positiva, azuis da classe negativa e marcações em laranja representam a localização das RI.



(e) Os círculos vermelhos representam *bags* da classe positiva, azuis da classe negativa e marcações em laranja representam *patches* ou instância das *bags*.

Figura 2.5: Ilustração dos diversos tipos de aprendizado para classificação. Figura inspirada no trabalho de Cheplygina et al. [38].

contém rótulos que podem ser incorretos. Assim, $D_s = \{(x_i, y_i), y_i = f(x_i), x_i \in X, y_i \in Y, i \in \mathbb{N}\} \cup \{(x_j, y_j), y_j \neq f(x_j), x_j \in X, y_j \in Y, j \in \mathbb{N}\}$ e, mesmo com essas condições, F deve prever os rótulos corretos para um conjunto D_t. Esse tipo de situação ocorre quando os elementos de rotulação são difíceis de serem visualizados, como a existência de pequenos nódulos cancerígenos em imagens médicas ou quando os rótulos são produzidos por não especialistas, causando erros em sua anotação. Geralmente, os trabalhos com tarefas referentes ao aprendizado com supervisão imprecisa se dividem em métodos para buscar a correção dos rótulos ou métodos robustos a erros de rotulação.

Em tarefas para supervisão inexata, algum nível de supervisão é dado, entretanto, essa supervisão não é suficiente para a utilização de aprendizado totalmente supervisionado, ilustrado na Figura 2.5d. O exemplo precursor dessa área é a predição do efeito do princípio ativo de drogas [54] e seu objetivo é indicar se uma forma de molécula produzirá efeito ou não. As moléculas têm muitas formas que podem ou não ativar seu efeito e a observação para cada uma de suas variantes é impraticável. Então, rotula-se as moléculas como um grupo para decidir se uma forma específica desse grupo produz algum efeito. Essas tarefas são conhecidas como tarefas de aprendizado por múltiplas instâncias *(multiple instance learning*, MIL).

Recentemente, outra classe de algoritmos com supervisão inexata surgiu a partir do estudo dos métodos de aprendizado profundo descritos na Seção 2.4.1. Os algoritmos utilizam somente rótulos de classificação para gerar mapas de saliências e inferir a localização de objetos contidos nas imagens de entrada. Essa classe de algoritmos recebe o nome do seu precursor mapeamento da ativação de classes (*class activation mapping*, CAM) [227] ou somente mapas de ativação, descritos na Seção 2.4.4.

2.4.1 Aprendizado Profundo

Aprendizado profundo (*deep learning*) [95] é uma subárea do aprendizado de máquina que emprega algoritmos para processar dados através de um aprendizado multiestágios inspirado na propagação de impulsos elétricos do cérebro humano. Inicialmente desenvolvidos para geração de algoritmos de aprendizado totalmente supervisionado, hoje o aprendizado profundo apresenta exemplos empregados em praticamente todas as divisões do aprendizado fracamente supervisionado, atuando, na maior parte das vezes, como extratores de características. As redes neurais profundas (*deep neural networks*, DNNs) [76] utilizam (muitas) camadas de neurônios, normalizações e *poolings* para processar os dados por meio de valores numéricos armazenados, chamados de parâmetros (com grande presença nos neurônios mas não restrito a eles). As informações ou características do problema são usualmente passadas através dessas camadas, com a saída da camada anterior fornecendo a entrada para a próxima camada.

Redes neurais convolucionais (*convolutional neural networks*, CNNs) [91, 171], como a LeNet [94] (primeira DNN largamente conhecida) e a AlexNet [91], foram precursoras do uso extensivo de camadas e trouxeram consigo o aumento do número de parâmetros e, consequentemente, a necessidade de maior poder computacional para treinar, testar e executar as redes. Arquiteturas, tais como as redes Inception [171] e ResNet [71], em algumas de suas versões, utilizam mais de cem camadas para construir suas redes.

As DNNs requerem muitas imagens para serem treinadas e a maior parte das pesquisas utiliza imagens da Internet, cuja coleta é relativamente fácil, entretanto, podem não ser suficientes para tarefas específicas que deveriam seguir protocolos rígidos de coleta de dados. Então, para

aplicações que são utilizadas em tarefas reais, torna-se necessário estabelecer estratégias que auxiliem o treinamento das DNNs, aumentando os dados disponíveis. Uma estratégia largamente utilizada com as redes é a aumentação de dados (*data augmentation*), que aplica transformações geométricas e radiométricas nas imagens para ampliar a quantidade dos dados [63]. Outra estratégia atualmente utilizada para treinar as redes é o reuso de redes pré-treinadas em outras bases e que fazem bem o papel de extração de características. Essas redes podem ser usadas para o retreinamento de outros métodos que serão gerados ou que já existam na literatura. O reuso do aprendizado das redes neurais recebe o nome de transferência de aprendizado (*transfer learning*) [129]. Esse reuso pode ser aplicado por meio do *fine tuning* (adaptação) do treinamento de uma ou mais camadas de um modelo, previamente treinado, em uma nova base de dados.

As DNNs, como outros métodos para geração de modelos, exibem problemas como o sobreajuste (*overfitting*). O sobreajuste ocorre quando um modelo se adapta excessivamente aos dados de treinamento, impedindo a generalização das predições e diminuindo sua efetividade. Há técnicas que diminuem a influência do sobreajuste, chamadas de técnicas de regularização. Uma delas, denominada abandono ou *dropout* [168], consiste em remover conexões entre camadas de neurônicos enquanto as DNNs estão sendo treinadas. Isso força que caminhos diferentes dentro das redes neurais sejam utilizados para derivar características para predições.

Tipicamente, uma CNN contém três tipos de camadas: camadas convolucionais, camadas de *pooling* e camadas totalmente conectadas (por exemplo, LeNet [94] e AlexNet [91]). Geralmente, a camada de classificação dessas redes é feita por um classificador totalmente conectado. Uma CNN totalmente convolucional [119] não utiliza camadas de neurônios totalmente conectadas. Isto é possível ao utilizar um classificador totalmente convolucional, ou seja, uma camada de convoluções 1×1, com o número de filtros igual ao número de classes, precedida por uma camada de *pooling* e uma função de classificação (*Sigmoid* ou *Softmax*). Dessa forma, os dois tipos de CNNs se diferenciam pelo tipo de classificador.

2.4.2 Aprendizado por Múltiplas Instâncias

As tarefas de aprendizado por múltiplas instâncias (*multiple instance learning* ou *multi-instance learning*, MIL) são definidas como aprender $F : X \rightarrow Y$ utilizando uma base de dados de treinamento $D_s = \{(X_i, y_i), f(X_i) = y_i, X_i = \{x_{ij}, j \in \mathbb{N}\} \subset X, i \in \mathbb{N}\}$, em que X_i é chamado de *bag* e x_{ij} suas *instâncias*, para prever os rótulos de um conjunto de dados D_t. No momento do treinamento, assume-se que as instâncias recebem o mesmo rótulo que suas *bags*, ou seja, $g(x_{ij}) = f(X_i) = y_i$, em que g é uma função que leva cada elemento de $\{x_{ij}, i \in \mathbb{N}, j \in \mathbb{N}\}$ no conjunto de rótulos Y, conforme ilustrado na Figura 2.5e.

O cenário para o uso de um método MIL ocorre quando há rótulos para a base original, mas eles não são suficientes para executar a tarefa requerida. Muitas vezes, rotular novamente as bases é uma atividade impraticável, ou seja, que demanda muito tempo ou o custo é alto demais. Um exemplo onde métodos derivados de MIL são muito utilizados é a segmentação de instâncias em imagens médicas [70, 205], em que as imagens possuem dimensões extremamente altas (40.000×40.000 pixels) e as áreas cancerígenas são muito confundidas com as áreas normais. Dessa forma, os rótulos de *bags* são usados para treinar instâncias que produzem a segmentação das *bags* ou para treinar geradores de máscaras de treinamento que são usados em métodos totalmente supervisionados.

As tarefas referentes ao MIL são divididas segundo dois objetivos distintos [29], inferir

sobre as *bags*, que é o mais comum, ou inferir sobre cada uma das instâncias individualmente. A inferência sobre as *bags* consiste em utilizar várias instâncias para calcular a predição final, cabendo às instâncias, dependendo dos algoritmos, o papel de auxiliar o resultado final das inferências. A classificação de instâncias é preponderantemente diferente da classificação de *bags*, pois a classificação de *bags* normalmente usa conjuntos de instâncias ao mesmo tempo nos treinamentos, enquanto que o treinamento da classificação de instâncias avalia cada instância individualmente. As funções de perda geralmente são diferentes para as duas tarefas porque na classificação de *bags* o erro de inferência de uma instância não afeta o resultado em nível de *bag*, mas um erro de inferência de instância na classificação de instâncias pode ser problemático [29].

A tarefa de inferência individual sobre as instâncias geralmente não tem significado sozinha, pois o interesse final na maior parte dos trabalhos é saber qual o resultado das *bags*. Então, apesar da tarefa de inferência sobre as instâncias existir independentemente da tarefa de inferência sobre as *bags*, a classificação individual sobre as instâncias é sempre utilizada como parte de um processo maior que resulta na classificação de *bags*.

O termo MIL padrão assume que *bags* negativas contêm apenas instâncias negativas e *bags* positivas contêm pelo menos uma instância positiva. Essas instâncias positivas são nomeadas testemunhas (*witnesses*). Na tarefa binária original de classificação para MIL, seu conjunto de rótulos é definido como $Y = \{0, 1\}$ e $f(X_i) = \begin{cases} 1 \ se \ \exists \ y_{ij}=1 \\ 0 \ se \ \nexists \ y_{ij}=1 \end{cases}$, em que y_{ij} seria o rótulo verdadeiro para x_{ij}, se existisse. Por essa definição, no momento da inferência, não é necessário analisar todas as instâncias para decidir que $F(X_i) = 1$. Usando a mesma definição de Y, pode-se relaxar a suposição padrão e trabalhar com a suposição de que a inferência seja feita com base na interação de todas as instâncias. Nesse caso, para gerar o resultado das *bags* ao final da inferência das instâncias, usa-se uma função G sobre a inferência original das instâncias F_{inf}, ou seja, $F = \underset{\mathbf{i,j}\in\mathbb{N}}{G} \circ F_{inf}(x_{ij})$ [70, 169]. Outra abordagem para garantir uma medida global para cada *bag* é aplicar um *pooling* das instâncias após um número de camadas [136] ou usar uma forma diferente de se criar o vetor de características final [72], o que altera a formulação anterior para $F = F_{inf} \circ \underset{\mathbf{i,j}\in\mathbb{N}}{G}(x_{ij})$, com G um extrator de características das instâncias e F_{inf} um método de inferência sobre a *bag* X_i [83].

2.4.3 Mapas de Saliências

Mapas de saliência são gerados por modelos de detecção de objetos salientes ou pelo uso da previsão de atenção visual. Modelos baseados em objetos salientes visam detectar apenas os objetos mais evidentes de uma cena e segmentar toda a extensão desses objetos. Modelos de previsão de atenção visual são usados para encontrar onde os humanos primeiramente fixam o olhar, ou seja, um pequeno conjunto de pontos de fixação [27], Figura 2.6. Atenção visual ou saliência visual é um problema de pesquisa fundamental em psicologia, neurobiologia, ciências cognitivas e visão computacional [192]. Tanto modelos de saliências quanto de previsão de atenção produzem mapas de saliências com valores contínuos (intervalo $[0, 1]$), em que um valor com predição mais alta indica que o pixel da imagem correspondente seja visto primeiramente.

Um mapa de saliências é definido como um modelo ou função $S : X \rightarrow Y$, em que $X \subset \mathbb{R}^{w \times h \times 3}$ e $Y \subset [0, 1]^{w \times h}$ em que $S(x)$, para $x \in X$, pode ser considerada como uma imagem em tons de cinza (Figura 2.7), com branco representando valores próximos a 1 e preto caso contrário, ou uma imagem térmica (Figura 2.8), em que o vermelho indica valores próximos

(a) Pontos de atenção (b) Mapa de atenção gerado por pontos.

Figura 2.6: Previsão de atenção. Figura presente nos trabalhos de Borji et al. [26, 27].

a 1, amarelo próximo a 0,5 e azul valores próximos a 0.

(a) Imagem original (b) Mapa de saliências

Figura 2.7: Exemplo de mapa de saliências gerado por um modelo tradicional de detecção de saliências [140]. Figura presente no trabalho de Borji et al. [27].

(a) Imagem original (b) Mapa de saliências

Figura 2.8: Exemplo de mapa de saliências colorido gerado pelo CAM. Figura presente no trabalho de Zhou et al. [227].

A detecção de objetos salientes ou a segmentação de objetos salientes é usualmente interpretada, na área de visão computacional, como um processo que inclui dois estágios: (i) detectar os objetos mais salientes e (ii) segmentar a região precisa desses objetos. Entretanto, dificilmente os métodos atuais distinguem entre os dois estágios, gerando modelos que provêm as duas tarefas ao mesmo tempo. O primeiro estágio não é limitado a apenas um objeto, entretanto, a maioria dos modelos tenta segmentar o objeto mais saliente, embora seus mapas possam salientar vários objetos em uma cena. O segundo estágio recai sobre o domínio dos problemas

clássicos de segmentação de visão computacional, com a diferença de que a precisão é determinada apenas pelo objeto mais saliente [27].

Os trabalhos nessa área são divididos de acordo com o período em que foram publicados, existindo três vertentes ou ondas de trabalhos [53]. O primeiro período, conhecido como detecção de saliência não baseada em aprendizado, utilizava premissas como o contraste local de imagens aplicado à teoria da estrutura biológica do sistema visual para encontrar objetos salientes [84]. O segundo período inspirou-se em modelos para encontrar regiões de saliências ou proto-objetos, a partir da detecção de saliências, considerado como um problema de segmentação binária [1]. O terceiro período, conhecido como métodos de detecção de saliências baseados em DNNs, não requer o uso de características extraídas manualmente, o que reduziu a dependência do conhecimento do viés de centralização, que é a ideia de como os olhos humanos se concentram no centro das imagens enquanto as rastreiam. Os métodos do primeiro e segundo períodos são algoritmos conhecidos como métodos tradicionais de detecção de saliências e geração de seus mapas. Atualmente, uma nova onda baseada em algoritmos fracamente supervisionados com base em algoritmos de supervisão inexata tem recebido crescente atenção [161, 227].

2.4.4 Mapas de Ativação

O mapeamento da ativação de classes (*class activation mapping*, CAM) [227] e o mapeamento da ativação de classes por pesos dos gradientes (*gradient-weighted class activation mapping*, Grad-CAM) [161] são dois métodos fracamente supervisionados de supervisão inexata, largamente utilizados para geração de mapas de saliências. Os dois foram criados para detectar saliências através da utilização das matrizes de características geradas por convoluções chamadas mapas de características. Eles utilizam a informação da ativação das características relativas a uma classe para transformar o conjunto de mapas de características final, provido após a última camada convolucional, em um mapa de saliências de uma classe específica. Essa classe de métodos, que compreende o CAM e o Grad-CAM conjuntamente com suas derivações, é chamada de mapas de ativação ou mapas de atenção, conforme ilustrado na Figura 2.9.

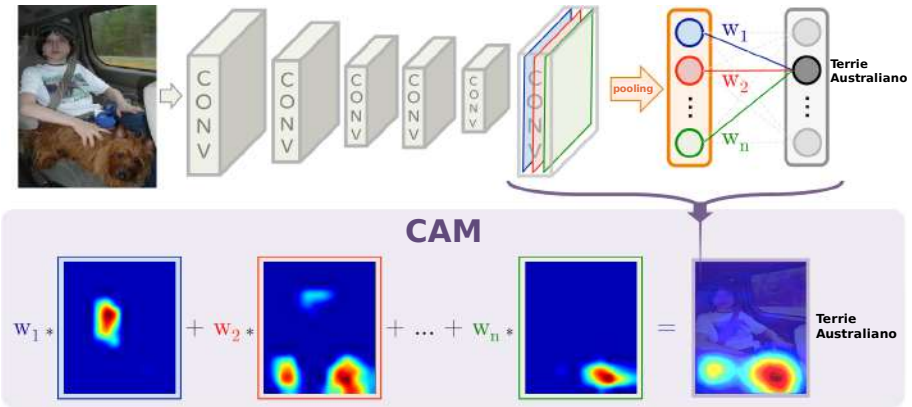


Figura 2.9: Ilustração de funcionamento do método CAM. A parte superior apresenta uma DNN convolucional de classificação e ressalta o conjunto de mapas de características da última convolução. A parte inferior apresenta a representação visual da Equação 2.1, que mostra a criação dos mapas de ativação e seus pesos. Figura modificada do trabalho de Zhou et al. [227].

O método CAM original usa modelos de CNNs cujas arquiteturas têm uma camada de *po-*

oling global após a última camada convolucional. Segundo a técnica CAM, define-se M^c_{CAM}, na Equação 2.1, como sendo o mapeamento de ativação de classes segundo a classe c, a coordenada da largura w, a coordenada da altura h, a ativação do k-ésimo mapa de características da última camada convolucional da localização espacial $f^k_{w,h}$, e os pesos do *pooling* global w^c_k relativo a classe c e ao k-ésimo mapa de características. Para transformar M^c_{CAM} em um mapa de saliências, é necessário escalar sua matriz para o tamanho da imagem original e normalizar todos os elementos da nova matriz resultante para que seus valores pertençam ao intervalo $[0, 1]$, gerando um mapa de saliências como na Figura 2.8. É necessário o aumento do resultado do método de mapa de ativação para o tamanho da imagem porque, na maior parte das arquiteturas de classificação, há um *downsampling* dos mapas de características no decorrer das camadas.

$$M^c_{CAM} = Relu\left(\sum_k w^c_k f^k_{w,h}\right). \tag{2.1}$$

O Grad-CAM generaliza a fórmula do CAM para todos os tipos de CNNs. Para isso, ele utiliza os gradientes relativos aos mapas de características descritos na Equação 2.2, em que $M^c_{Grad\text{-}CAM}$ é o mapa de ativação, Z é a dimensão espacial do mapa de características, y^c é o cálculo da pontuação para a classe c provido pela camada de neurônios após o *pooling* global e os outros elementos têm o mesmo significado declarado no CAM. O Grad-CAM é um discriminador de classe e localiza regiões relevantes em imagens, entretanto, não possui a capacidade de destacar detalhes finos como alguns algoritmos de segmentação (a segmentação das RI são inexatas), mas tem a capacidade de achar corretamente a localização de objetos.

$$M^c_{Grad\text{-}CAM} = Relu\left(\sum_k w^c_k f^k\right), \quad w^c_k = \frac{1}{Z}\sum_w\sum_h \frac{\partial y^c}{\partial f^k_{w,h}}. \tag{2.2}$$

2.4.5 Abordagens Baseadas em Atenção e Seleção de Características

Em arquiteturas de aprendizado profundo, abordagens baseadas em atenção produzem modelos baseados em atenção [12, 145]. Modelos baseados em atenção incorporam a noção de relevância de características, permitindo que o modelo focalize dinamicamente a atenção apenas em certas partes da entrada que efetivamente executam a tarefa requerida [30]. Isso significa que os modelos baseados em atenção selecionam características relevantes que ajudam a melhorar e apoiar suas decisões.

Alguns autores distinguem a seleção de características existente em métodos fracamente supervisionados das abordagens baseadas em atenção pelo motivo de métodos fracamente supervisionados incluírem estruturas extras no processo de seleção de características, enquanto isso seria implícito em modelos baseadas em atenção [56]. No entanto, como ideia base para o desenvolvimento, ambos os conceitos surgiram da atenção visual biológica [27]. Parte dos modelos baseados em atenção aplicam estratégias de seleção vindas de abordagens fracamente supervisionadas para destacar ou ocultar características [40], tornando os dois conceitos muito próximos entre si.

As abordagens baseadas em atenção normalmente usam otimização de parâmetros, em que pesos de atenção altos correspondem diretamente às RI [2]. Para esta tese, define-se que mapas de ativação baseados em atenção produzem mapas de saliências baseados nos parâmetros de

otimização treinados para este fim. A Equação 2.3 [34] mostra uma formulação usual para produzir mapas de ativação baseados em atenção em que f_k é o k-ésimo mapa de características da última camada convolucional e M_{act} é o resultado do método de mapa de ativação produzido através de parâmetros otimizados para ressaltar as características em f_k enquanto minimiza o erro final de classificação (onde $\otimes$ é a multiplicação elemento a elemento).

$$f_k^{\otimes}(x) = f_k(x) \otimes M_{act}(x). \tag{2.3}$$

2.4.6 Adaptação de Domínio não Supervisionada

Muitos métodos de aprendizado profundo fazem uma suposição comum: os dados de treinamento e teste são extraídos da mesma distribuição. Quando essa restrição é violada, um classificador treinado no conjunto de origem (no caso deste trabalho, o conjunto de treinamento) provavelmente sofrerá uma queda no desempenho quando testado no conjunto de destino (conjunto de validação/teste) devido as suas diferenças. O problema de treinar um classificador discriminativo ou outro preditor na presença de uma mudança entre as distribuições dos conjuntos de treinamento e validação/teste é conhecido como adaptação de domínio [61] e os conjuntos de treinamento e validação/teste são denominados domínio de origem e domínio de destino, respectivamente. Adaptação de domínio usando uma única fonte refere-se ao objetivo de aprender um conceito sobre os dados rotulados em um domínio de origem e que funcione em um domínio de destino. A adaptação de domínio não supervisionada aborda especificamente a situação em que há dados de origem rotulados e apenas dados de destino não rotulados disponíveis para uso durante o treinamento [195].

Devido a sua capacidade de adaptar dados rotulados para uso em uma nova tarefa, a adaptação de domínio pode reduzir a necessidade de dados rotulados no domínio de destino. Quando o domínio de destino é o mesmo e os desafios recaem sobre o que é aprendido no domínio de origem, um extrator de características pode ser treinado para influenciar o classificador a não aprender características que não sejam comuns entre o domínio de origem e domínio de destino [195]. Neste caso, o trabalho de Ganin e Lempitsky [61] propôs um extrator de características que força o classificador de domínio a ter um desempenho ruim, negando o gradiente do classificador de domínio com uma camada de reversão de gradiente [61]. Ao realizar a retropropagação do erro para atualizar os pesos do extrator de características, o processo de treinamento deve confundir o classificador de domínio e com isso diminuir a diferença entre as características utilizadas no domínio de origem e destino, que são o motivo dos modelos terem boa eficácia somente na origem. Isso força a produção de resultados mais condizentes entre o domínio e o destino. A Figura 2.10 mostra um exemplo de arquitetura nesta linha que foi proposta por Ganin e Lempitsky [61].

Seja a DNN G da Figura 2.10, que mapeia cada entrada $x \in X$ em um rótulo de classes $y \in Y \subset [0,1]^n, n \in \mathbb{N}$ e um rótulo de domínio $d \in [0,1]$. Ela pode então ser descrita como uma sequência de três outros mapeamentos. Assume-se que a entrada x seja a base para um mapeamento de um modelo G_f (um extrator de características) que leva em um vetor de características n-dimensional $f \in \mathbb{R}^d$. G_f pode incluir várias camadas totalmente conectadas e o vetor de parâmetros de todas as camadas neste mapeamento é denotado como θ_f, ou seja, $f = G_f(x; \theta_f)$. O segundo mapeamento G_y (preditor de rótulos) usa o vetor de características f como entrada para um modelo que o leva diretamente para o rótulo y, onde os parâmetros desse

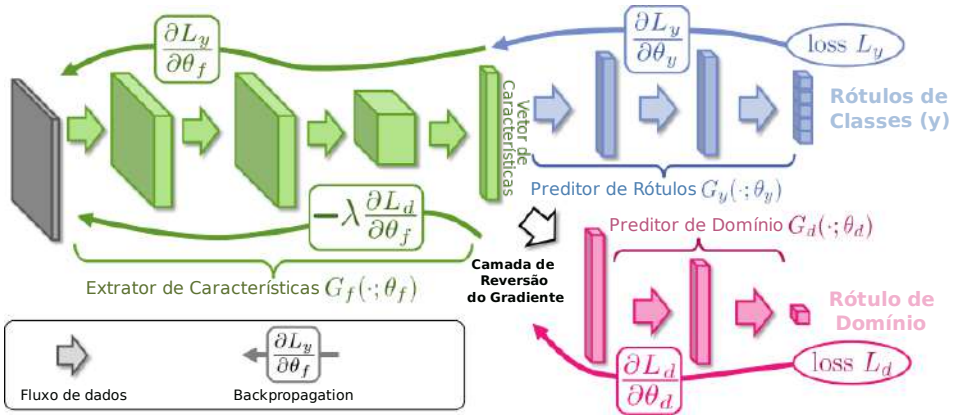


Figura 2.10: Ilustração da adaptação de domínio não supervisionada. A imagem mostra uma DNN como extratora de características em verde (G_f) e um preditor de rótulo em azul (G_y) que juntos formam uma DNN padrão (G). A adaptação de domínio não supervisionada é obtida pela adição de um classificador de domínio (G_d), em vermelho, conectado ao extrator de características através de uma camada de reversão de gradiente que multiplica o gradiente por uma certa constante negativa durante o treinamento baseado em retropropagação. Caso contrário, o treinamento prossegue de maneira padrão e minimiza: os erros segundo as previsões dos rótulos para indivíduos do domínio de origem; e a perda relativa à classificação de domínio para todas as amostras. A reversão dos gradientes garante que as características nos dois domínios sejam semelhantes (tão indistinguíveis quanto possível para o classificador), resultando em características independentes de domínio. Figura adptada do trabalho de Ganin e Lempitsky [61].

mapeamento são denotados como θ_y. Finalmente, o terceiro mapeamento G_d (classificador de domínio) usa o mesmo vetor de características f para transformá-lo no rótulo de domínio d que contém os parâmetros θ_d. Por simplicidade e para diminuir a quantidade de notação no decorrer do texto, as letras gregas referentes aos parâmetros não serão mostradas junto aos modelos (θs).

Para treinar G corretamente, deve-se minimizar a função de perdas de predição de rótulos (*Loss* L_y), na parte anotada (G_y) do conjunto de treinamento, e os parâmetros do preditor de rótulos, que são otimizados para minimizar a função de perdas correspondente ao domínio (*Loss* L_d) para as amostras de domínio de origem e destino. O extrator de características (G_f) é influenciado pelas duas funções de perdas, sendo impactado negativamente pela variação da função de domínio. Isso garante a discriminação das características f, menos dependentes de domínio, e o bom desempenho geral de predição da combinação do extrator de características e do preditor de rótulos. A Equação 2.4 expressa o erro para treinamento do método da Figura 2.10.

$$L(x_i, y_i, d_i) = L_y(G_y(G_f(x_i)), y_i) - \lambda L_d(G_d(G_f(x_i)), d_i). \tag{2.4}$$

O parâmetro λ controla a troca entre os dois objetivos, ou seja, (1) minimizar o erro relativo à predição de classes e (2) maximizar o erro de domínio, que moldam as características durante o aprendizado. Nota-se que, para o extrator de características G_f, cujo gradiente foi invertido, o objetivo é de maximização do gradiente relativo ao domínio.

Capítulo 3

Trabalhos Relacionados

Esta tese de doutorado envolve quatro frentes de evolução do conhecimento, sendo duas delas categorizadas como aprendizado fracamente supervisionado do tipo de supervisão inexata: aprendizado por múltiplas instâncias (MIL) e uso de mapas de saliências produzidos por mapas de ativação e métodos derivados, que somente foram possíveis com o advento das DNNs. A terceira frente, modelos baseados em atenção, está relacionada ao uso de técnicas para aumentar a relevância de características importantes nas tarefas requeridas. A quarta frente descreve o uso das principais aplicações de DNNs na agricultura, mais precisamente na automatização do Manejo Integrado de Pragas (MIP). Assim, este capítulo está dividido da seguinte forma. A primeira seção descreve técnicas tradicionais de MIL e sua aplicação utilizando DNNs (Seção 3.1), a segunda seção apresenta os principais métodos fracamente supervisionados baseados em DNNs para detecção de saliências (Seção 3.2), a terceira seção expõe trabalhos que produziram conceitos importantes relacionados ao mecanismo de atenção (Seção 3.3), a quarta seção discute os principais trabalhos utilizando DNNs na agricultura (Seção 3.4) e, por fim, a última seção lista todos os trabalhos que tiveram influência nas abordagens propostas nesta tese (Seção 3.5).

3.1 Aprendizado por Múltiplas Instâncias

O trabalho de Dietterich et al. [54] foi o primeiro a utilizar o MIL, enquanto abordava o problema de predição de atividade de drogas, como descrito na Subseção 2.4.2. Primeiramente, eles rotularam todas as formas diferentes de moléculas de uma mesma droga como exemplos positivos e usaram a estrutura de um aprendizado totalmente supervisionado para predizer se as drogas seriam ativadas. No entanto, o resultado foi insatisfatório e com muitos falsos positivos e negativos, pois uma molécula de um medicamento pode conter centenas de formas de baixa energia e, para ativar seu efeito, é necessário achar as formas que se ligam corretamente com moléculas de proteínas [137]. Então, eles determinaram o embasamento inicial sobre o aprendizado por múltiplas instâncias e abordaram uma molécula como uma *bag* e suas diferentes formas como instâncias não rotuladas de um problema.

Logo após a publicação de Dietterich et al. [54], os trabalhos de Long e Tan [120] e Blum e Kalai [22] desenvolveram métodos específicos para múltiplas instâncias. É uma premissa aceitável pensar que as instâncias são variáveis aleatórias independentes e identicamente distribuídas (*independent and identically distributed*, i.i.d.). Porém, Zhou e Xu [230] indicaram que

as instâncias não são e nem devem ser consideradas independentes, enquanto que as *bags* de um conjunto têm maior probabilidade de serem amostras i.i.d.. Com base nessa premissa, Zhou et al. [232] desenvolveram um algoritmo eficaz para MIL ao propor um método que admitiu uma correlação de dependência entre as instâncias. Zhou e Xu [230] também indicaram que existe uma ponte entre o aprendizado por múltiplas instâncias e o aprendizado semissupervisionado, dizendo que o MIL seria um caso especial de aprendizado semissupervisionado, ideia que foi modificada com a definição formal das tarefas para os dois tipos de técnicas, conforme descrito na Subseção 2.4.2.

Maron [126] atribuiu ao MIL a condição de *framework* para múltiplas instâncias, gerando um algoritmo clássico de MIL que funcionava sob a suposição de que as instâncias positivas seriam localizadas em uma única região no espaço de características. Eles apresentaram o método de densidade diversa (*diversity density*, DD) que calcula uma função de probabilidade ao redor de um ponto no espaço de características chamado conceito positivo (*positive concept*). O ponto relativo ao conceito positivo está o mais próximo possível das instâncias positivas de *bags* positivas e o mais longe possível de instâncias negativas. Em outras palavras, o DD oferece a probabilidade média de um ponto no espaço de características pertencer à classe positiva dada sua distância de instâncias negativas em sua vizinhança. Zhang e Goldman [219] utilizaram o algoritmo de maximização de expectativa (*expectation-maximization*, EM) para localizar o máximo da função entregue por DD, dando origem ao algoritmo EM-DD, considerado o mais tradicional em MIL.

Segundo Ouadou [137] e Zhou [229], muitos algoritmos foram desenvolvidos dentro do *framework* MIL. Em suma, quase todos os algoritmos de aprendizado supervisionado tiveram suas versões de múltiplas instâncias [10, 96, 232]. Vários autores tentaram adaptar os métodos de aprendizado totalmente supervisionado, mudando seu foco do uso em indivíduos providos por bases de dados para o uso em instâncias e *bags* [228], como Andrews et al. [8], que propuseram dois métodos para estender a definição do algoritmo máquinas de vetores de suporte (*suport vector machines*, SVM) para o MIL. Os dois métodos ficaram conhecidos como formulação do padrão de margem máxima do MIL (*maximum pattern margin formulation of MIL*, mi-SVM) e formulação da margem máxima de bags para MIL (*maximum bag margin formulation of MIL*, MI-SVM). O mi-SVM difere do SVM por não conhecer os rótulos reais das instâncias e assim tratar as instâncias como variáveis inteiramente desconhecidas. Ele tenta encontrar um hiperplano que classifique todas as instâncias negativas de *bags* negativas como negativas e pelo menos uma instância da *bag* positiva como positiva. No caso do MI-SVM, o foco é na *bag*, pois uma *bag* positiva é representada pela sua instância mais positiva, enquanto uma *bag* negativa é representada pela sua instância menos negativa. Depois que essas instâncias são identificadas para cada *bag*, as outras instâncias podem ser descartadas. A diferença entre o mi-SVM e o MI-SVM é que, enquanto no mi-SVM todas as instâncias da *bag* fazem parte do processo de otimização, no MI-SVM uma variável seletora é usada para determinar a instância que representa a *bag*.

Outro exemplo característico de MIL foi a adaptação de Wang e Zucker [189] para o método de K-vizinhos mais próximos (*K-nearest neighbors*, KNN). Eles usaram o algoritmo de Hausdorff para medir a distância entre as *bags*, mas essa aplicação levou a uma contradição conhecida como "menor vencedor". O KNN é um método não paramétrico que classifica um ponto de teste de acordo com o voto da maioria de seus vizinhos. Dependendo da forma como as *bags* são definidas, existe a possibilidade de que o rótulo correto de uma *bag* seja proporci-

onado pelo menor número de suas instâncias e, em alguns casos, a *bag* tem um maior número de vizinhos negativos (para uma *bag* ser positiva, deve existir ao menos uma instância positiva), o que contradiz a definição de vizinhança do KNN. Um pequeno grupo de *bags* positivas, que contém poucas instâncias positivas e muitas *bags* negativas ao seu redor, é considerado negativo e, assim, o menor número de *bags* positivas deveria ser o vencedor (menor vencedor).

Por causa da forma de treinamento das redes neurais, a qual utiliza o erro entre a inferência e o valor real dos rótulos para atualizar os parâmetros, Dietterich et al. [54] afirmaram que as redes da época não se adequavam ao uso do MIL, pois não havia certeza da rotulação correta das instâncias, já que elas assumiam o rótulo das *bags*. Então, para tornar as redes neurais aptas ao seu uso, Zhou e Zhang [231] propuseram uma nova função de erro $L(x_{ij}, y_i)$ que considerava o valor total relativo às *bags* X_i no treinamento do modelo F sobre as instâncias x_{ij} com rótulos de *bags* y_i. Esse erro, expresso na Equação 3.1, depende do erro relativo a todas instâncias para computar o valor de cada *bag*. O desempenho deste algoritmo, na época, foi comparável a outros métodos MIL. Zhang e Zhou [218] propuseram uma melhoria desse algoritmo que incluiu o pré-processamento dos vetores de características usando o método de análise de componentes principais (*principal component analysis*, PCA).

$$L(x_{ij}, y_i) = \frac{1}{2}(\max_j\{F(x_{ij})\} - y_i)^2. \tag{3.1}$$

Muitos autores aplicaram o MIL em seus estudos para resolver problemas de recuperação de imagens baseada em conteúdo [35, 194]. Pela definição do problema, buscava-se o conteúdo dentro das imagens independentemente de sua posição, então utilizaram imagens originais como *bags* e as partições das imagens como instâncias de características distintas para treinar os algoritmos de busca. O MIL também foi utilizado para detecção de ações ou alvos em vídeos [6, 11], pois a tarefa de rotular quadro a quadro é dispendiosa, então consideraram uma *bag* sendo um conjunto de quadros e procuraram por instâncias positivas dentro desse conjunto com objetos ou ações desejadas.

Em relação aos métodos MIL mais atuais, nos últimos sete anos, técnicas baseadas em convoluções para o aprendizado profundo são empregadas largamente em tarefas reais, porém é difícil criar arquiteturas de propósito geral, pois o uso de premissas apropriadas para cada tipo de domínio facilita o aprendizado. Papandreou et al. [138] desenvolveram uma arquitetura de propósito geral para trabalhar com MIL como uma forma de aumentação do número de imagens, compondo suas *bags* como distorções de escala de uma imagem original. Eles usaram a função de cálculo de erro de Zhou e Zhang [231] (Equação 3.1) para escolher a transformação de escala preferida para ser retropropagada como erro de *bag*. Outro exemplo das poucas arquiteturas gerais disponíveis é o trabalho de Sun et al. [169], em que uma arquitetura foi proposta, denominada de rede neural convolucional de aprendizado por múltiplas instâncias (*multiple instance learning convolutional neural network*, MILCNN), que fundiu uma DNN com o cálculo da probabilidade de *bags* para múltiplas instâncias e múltiplas categorias. Isso permitiu que eles gerassem uma nova função de cálculo de erro para redes neurais. Os modelos da MILCNN recebem o número de instâncias de uma *bag* e inferem cada instância como uma imagem separada e, no final, usam o cálculo de probabilidade das *bags*.

Ilse et al. [83] propuseram uma arquitetura geral para MIL, chamada *Attention-based Deep MIL*, que foi testada em vários *benchmarks* com tipos de *bags* e instâncias diferentes. Os autores propuseram uma forma de *pooling* de atenção entre o cálculo da pontuação de cada instância

para evidenciar as características locais dentro de uma *bag*, o que aumentou a influência de instâncias significativas para o cálculo final das probabilidades das *bags* e, através da ativação dos neurônios de atenção, conseguiram localizar com maior precisão as instâncias contendo o objeto de classificação. A *Attention-based Deep* MIL é uma das arquiteturas utilizadas nesta tese para comparação de resultados nas Subseções 6.2.3 e 6.2.4. Yan et al. [209], na mesma linha que Ilse et al. [83], criaram um *pooling* dinâmico para DNNs baseadas em redes de cápsulas [153]. He et al. [72] criaram uma rede convolucional profunda de múltiplas instâncias (*multiple instance deep convolutional network*, MIDCN) para classificação de imagens. Eles calcularam as diferenças entre vetores de características de instância e vetores de características pré-calculados, chamados protótipos, e previram as classes usando essas diferenças.

Há também trabalhos gerais de localização e segmentação utilizando MIL, como Liang et al. [107] que testaram anotações de classes, localizações parciais de objetos e *bounding boxes* para verificar a precisão de segmentações fracamente supervisionadas relativas a cada tipo de anotação. Porém, para segmentação em imagens médicas, alguns resultados foram promissores. Um exemplo é o trabalho de Xu et al. [205], que propôs uma abordagem de aprendizado fracamente supervisionada para a segmentação de imagens histopatológicas usando apenas rótulos em nível de imagens. Os autores geraram modelos que foram treinados em instâncias significativamente pequenas para gerar máscaras de segmentação utilizadas em uma fase de aprendizado totalmente supervisionado. O resultado final da segmentação com máscaras criadas automaticamente foi comparável ao resultado das mesmas arquiteturas treinadas com máscaras geradas por especialistas.

Ainda sobre a segmentação em imagens médicas, Hashimoto et al. [70] propuseram um novo método para a classificação de subtipos de câncer utilizando MIL, que conseguiu classificar automaticamente tumores por meio de um *ensemble*, junção das previsões de modelos por votação, de modelos treinados em escalas distintas e, utilizando a ativação de seu método de atenção baseado no método de Ilse et al. [83], geraram áreas de segmentação para regiões cancerígenas. Este método fez a junção entre o aprendizado com supervisão inexata de múltiplas instâncias e a geração de mapas de saliências pela ativação por classes, como no CAM.

Fora do contexto de segmentação, mas ainda no domínio da medicina, Choukroun et al. [42] introduziram um método MIL para classificação de mamografias usando uma VGGNet seguida por uma rede neural refinada e totalmente conectada, modificada para o paradigma MIL. Li et al. [101] desenvolveram uma DNN baseada em atenção para MIL, que usava um mecanismo adaptativo em DNNs para classificar instâncias significativas para imagens histopatológicas. Cheplygina et al. [38] descreveram muitos outros trabalhos utilizando MIL, que foram produzidos para a área da medicina e afirmaram que existe uma ligação entre o aumento dos estudos de métodos fracamente supervisionados com a carência de profissionais médicos qualificados para rotular as imagens.

No domínio de localização de objetos em vídeos, Cinbis et al. [44] usaram um método de busca seletiva para gerar rótulos de localização para objetos e, como extratores de características, utilizaram a arquitetura AlexNet conjuntamente com Fisher Vectors [155]. O método foi desenvolvido para usar instâncias como *bounding boxes* que, iterativamente, por meio da melhoria dos resultados, eram refinados para se acoplarem melhor aos objetos e inferir localizações mais precisas. Luo et al. [123] usaram o método de EM em vídeos conjuntamente com DNNs para detectar trechos contendo ações e objetos. Eles mostraram que modelos anteriores baseados em atenção, como o *Attention-based Deep* MIL, implicitamente violavam a suposição de

que *bags* negativas não teriam instâncias positivas, porque a classe negativa recebia pontuação através dos pesos de atenção como se fossem instâncias positivas.

Li et al. [97] propuseram um método baseado em MIL chamado aprendizado por múltiplas instâncias com fluxo duplo (*dual-stream multiple instance learning*, DSMIL) para classificação de tumores. Eles propuseram um agregador para juntar as predições de uma arquitetura treinada com dois fluxos. O primeiro fluxo faz a predição de *patches* com um quinto do tamanho das imagens originais, enquanto o segundo com *patches* de um vigésimo do tamanho das imagens originais. Eles propuseram o uso de aprendizado contrastivo auto-supervisionado para extrair caraterísticas representativas para o aprendizado MIL e um mecanismo de fusão baseado em pirâmides para melhorar a extração de características e a localização.

Recentemente, nos últimos dois anos, alguns trabalhos baseados em múltiplas instâncias abordaram temas atuais, como a avaliação automatizada da gravidade da COVID-19 em imagens 3D de tomografias computadorizadas. He et al. [73] produziram um *framework* de aprendizado realizando a segmentação do lobo pulmonar e a classificação de várias instâncias. Xue et al. [208] propuseram um módulo MIL para combinar efetivamente as representações de características relativas às imagens e um módulo de aprendizado contrastivo para unir as avaliações do módulo MIL com dados do histórico hospitalar dos pacientes. Na área de reconhecimento de *deep-fake*, Li et al. [105] introduziram a predição de um tipo de ataque de modificação de rostos em que nem todos são falsos ou manipulados, inclusive, propuseram uma nova base de dados baseada nessa premissa. Eles abordaram o problema como MIL, tratando as faces como instâncias e os vídeos como *bags*.

Yu et al. [214] propuseram um novo sistema de predição utilizando múltiplas instâncias, algoritmos genéticos e sistemas *fuzzy* para predição de objetivos de tropas em combate. Eles usaram dados dos diferentes batalhões de tropas como instâncias e a junção como uma *bag*. A lógica *fuzzy* permitiu juntar regras de especialistas com pesos calculados por algoritmos genéticos nos modelos desenvolvidos. Gao et al. [62] utilizaram múltiplos modelos MIL como professores em uma estratégia professor-estudante para localização de objetos. Modelos de localização por múltiplas instâncias experimentam dificuldade em achar soluções ótimas globais. A junção de vários modelos ajuda a uma localização única mais acurada e precisa para direcionar as redes estudantes. Li et al. [98] trabalharam com MIL para classificação de cenas acústicas, eles utilizaram vetores de características, que foram extraídos com redes neurais temporais para espectrogramas, como instâncias para predizer o lugar onde as cenas sonoras foram colhidas (parques, aeroportos, campo).

Na agronomia, Petti e Li [142] treinaram modelos MIL em imagens produzidas por veículos aéreos não tripulados (VANTs) para reduzir a necessidade de dados manualmente anotados na estimativa de produção de algodão. Eles geraram um método, que também envolveu o aprendizado ativo [163], para contagem de flores de algodão. O aprendizado ativo minimizou ainda mais a necessidade de intervenção humana no processo de anotação dos rótulos das imagens. Lu et al. [121] propuseram o uso de MIL na classificação de sintomas de doenças e geraram o primeiro método fracamente supervisionado para identificar ameaças na cultura de trigo [88]. Eles propuseram o sistema de diagnóstico de doenças do milho baseado em múltiplas instâncias (*multiple instance learning based wheat disease diagnosis system*, DMIL-WDDS). Esse sistema usa uma DNN para classificar sintomas de doenças e gerar mapas de características de diferentes regiões afetadas para que essas regiões sejam agregados em um classificador MIL. Com base nos mapas de características, eles também propuseram um localizador de sintomas.

Finalmente, Yu et al. [213] usaram mapas de ativação para gerar exemplos de treinamentos para um algoritmo MIL de segmentação de pixels em imagens de raízes. Pixels com rótulos positivos e negativos marcados nos mapas de ativação foram extraídos como amostras e unidos em *bags*. Bellocchio et al. [18] e Adke et al. [3] utilizaram MIL para classificação e contagem de frutos e borbulhas de algodão respectivamente. Os dois trabalhos utilizaram contagem através de regressão linear fracamente supervisionada calculada a partir de rótulos para identificar a presença de frutos ou borbulhas de algodão.

A aplicação de métodos MIL na classificação de pragas e vetores de doenças é escassa. Possivelmente, as publicações referentes a esta tese são as iniciais para o uso de múltiplas instâncias e localização fracamente supervisionada. A Tabela 3.1 lista todos os trabalhos sobre múltiplas instâncias desta seção.

Tabela 3.1: Trabalhos relacionados ao Aprendizado por Múltiplas Instâncias.

Trabalhos	Informações Relevantes	Ano
Dietterich et al. [54]	Trabalho inicial sobre MIL e DNNs não seriam propícias ao MIL.	1997
Long e Tan [120]	Método específico para retângulos paralelos ao eixo usando múltiplas instâncias.	1998
Blum e Kalai [22]	Método para redução de ruídos no MIL.	1998
Maron [126]	Atribuição do MIL como *framework*.	1998
Wang e Zucker [189]	Adaptação do K-vizinhos mais próximos para MIL. Menor vencedor.	2000
Zhang e Goldman [219]	Algoritmo EM-DD.	2002
Zhou e Zhang [231]	Erro sobre o valor total relativo às *bags* em DNNs para MIL.	2002
Andrews et al. [8]	Criação de algoritmos derivados do SVM: MI-SVM e mi-SVM.	2003
Zhang e Zhou [218]	PCA conjuntamente com algoritmo [231].	2004
Chen et al. [35]	Recuperação de imagens baseadas em conteúdo.	2006
Zhou [228]	Possibilidade de adaptar métodos totalmente supervisionados para MIL.	2006
Zhou e Xu [230]	Instâncias não são i.i.d. e nem devem ser consideradas como entidades não relacionadas.	2007
Ali e Shah [6]	Detecção de ações ou alvos em vídeos.	2008
Babenko et al. [10]	Algoritmo de *boosting* para MIL.	2008
Babenko et al. [11]	Detecção de ações ou alvos em vídeos.	2009
Zhou et al. [232]	Algoritmo de grafos para MIL.	2009
Leistner et al. [96]	Algoritmo de florestas aleatórias para MIL.	2010
Papandreou et al. [138]	Convolução para fazer as redes neurais invariantes a escalas.	2015
Cinbis et al. [44]	Método de busca seletiva para geração de rótulos como *bounding boxes*.	2016
Wei e Zhou [194]	Recuperação de imagens baseadas em conteúdo.	2016
Sun et al. [169]	Novo cálculo de erros para instâncias de uma *bag*.	2016
Choukroun et al. [42]	Método MIL para classificação de mamografias.	2017
Ouadou [137]	Detecção de veículos.	2017
Lu et al. [121]	Proposição do DMIL-WDDS, MIL para classificação de doenças do trigo.	2017
Zhou [229]	Revisão de vários conceitos relacionados.	2018
Ilse et al. [83]	*Deep* MIL com *pooling* de atenção. Violação da condição de múltiplas instâncias.	2018
Yan et al. [209]	*Pooling* dinâmico para DNNs com cápsulas.	2018
Cheplygina et al. [38]	Correlação entre o aumento de métodos fracamente supervisionados e carência de profissionais médicos.	2019
He et al. [72]	Protótipos para extrair características das instâncias.	2019
Li et al. [101]	DNN com atenção para imagens histopatológicas.	2019
Xu et al. [205]	CAMEL para a segmentação de imagens histopatológicas.	2019
Bellocchio et al. [18]	MIL para classificação e contagem de frutos.	2019
Hashimoto et al. [70]	*Ensemble* para escolha da escala propícia.	2020
Liang et al. [107]	Precisão da segmentação relativa à diferentes tipos de anotações.	2020
Luo et al. [123]	Método de EM em vídeos conjuntamente com DNNs.	2020
Li et al. [105]	Reconhecimento de faces falsas em *deepfake* usando MIL.	2020
Yu et al. [213]	Mapas de ativação e MIL para segmentação de pixels	2020
Li et al. [97]	DSMIL para classificação e localização fracamente supervisionada de câncer.	2021
He et al. [73]	*Framework* MIL para avaliação da gravidade da COVID-19.	2021
Xue et al. [208]	Um novo módulo MIL outro contrastivo para avaliação da gravidade da COVID-19.	2021
Petti e Li [142]	Método MIL para contagem de flores de algodão.	2022
Yu et al. [214]	MIL e fuzzy com pesos calculados por algoritmos genéticos.	2022
Gao et al. [62]	Múltiplos modelos MIL como professores em uma estratégia professor-estudante.	2022
Li et al. [98]	MIL para classificação de cenas acústicas.	2022
Adke et al. [3]	MIL para classificação e contagem borbulhas de algodão.	2022

3.2 Mapas de Saliências Fracamente Supervisionados Baseados no Aprendizado Profundo

Os trabalhos relacionados com essa área surgiram recentemente como tentativa de explicar o comportamento das DNNs, pois são frequentemente vistas como caixas-opacas [151]. Alguns trabalhos que utilizam modelos baseadas em atenção, como Teh et al. [176], mesmo não citando explicitamente o uso de mapas de ativação, usam os parâmetros de atenção aprendidos relativamente aos mapas de características. Então, trabalhos que utilizam atenção para salientar áreas onde há a possibilidade de se achar as regiões de interesse (RI) em mapas de características, utilizam o princípio da detecção de saliências. Porém, o trabalho de Zhou et al. [227] deu origem ao mapeamento da ativação de classes (CAM), conjuntamente com o termo mapas de ativação, enquanto o trabalho de Selvaraju et al. [161] deu origem ao Grad-CAM (Seção 2.4.4). Ambos utilizaram as saliências explicitamente para localização e segmentação.

Zhou et al. [227] e Selvaraju et al. [161] foram influenciados por mecanismos anteriores aos mapas de ativação, como o trabalho de Oquab et al. [136]. Dada uma imagem, eles utilizaram um extrator de características, com entrada menor que o tamanho da área total, como uma janela deslizante para percorrer essa imagem. Depois, geraram uma composição dos mapas de características considerando a mesma ordem espacial relativa a janela deslizante e aplicaram convoluções nessa composição, criando um novo conjunto de mapas representando a imagem original inteira. Eles utilizaram a composição dos mapas de características para achar saliências onde os objetos teriam a maior probabilidade. Pinheiro e Collobert [143] também, com um trabalho prévio ao conhecimento dos mapas de ativação, utilizaram os mapas de características como base para segmentação fracamente supervisionada, fazendo com que cada classe tivesse seu próprio mapa de características e utilizaram conhecimentos prévios de características do contexto para diminuir problemas com pixels falsamente segmentados (falsos positivos).

Após o CAM e o Grad-CAM, vários trabalhos tentaram estender seu entendimento e melhorá-lo, como Kwak et al. [92] que criaram o mapeamento de ativação de classes para o agrupamento de superpixels (*superpixel-pooled* CAM, SP-CAM), uma forma de estender o CAM para sua arquitetura de DNN que usa superpixels para segmentar imagens. Zhu et al. [233] também geraram uma versão para o CAM chamada mapa de ativação de instâncias (*instance activation mapping*, IAM) que cobre a extensão total dos objetos testados, enquanto o CAM mostrava apenas localização e as partes mais discriminativas de objetos. Ahn e Kwak [4] usaram o CAM como base para calcular a afinidade entre áreas das imagens e gerar rótulos para DNNs de segmentação. Nguyen et al. [135] adaptaram uma versão do CAM para detectar a localização de ações temporais em vídeos, ou seja, detectar o mapeamento de ativação de classes pelo tempo (*temporal* CAM, T-CAM). Chen et al. [32], visando à detecção automática de defeitos em superfícies com texturas diversificadas, criaram o mapeamento de ativação de classes de atenção espacial (*spatial attention* CAM, SA-CAM) proposto para melhorar a adaptabilidade da segmentação e gerar um mapa de calor mais preciso. Wang et al. [188] propuseram o mapeamento de ativação de classes por pontuações (Score-CAM), que gera seu mapa de ativação final obtendo pesos para cada mapa por meio das pontuações dadas nas inferências de classificação, o resultado final é obtido por uma combinação linear dos pesos dados na classificação e dos mapas de ativação.

Sun et al. [170] propuseram uma estratégia de inferência condicional por classes pra gerar

mapas de saliências mais precisos. Eles mostraram que os mapas de ativação marcam fracamente áreas fora das regiões com características mais discriminativas, porém, quando essas regiões são removidas sem retreinamento, os mapas de ativação mostram outras partes das RI. Criando uma estratégia de corte e união, eles conseguiram gerar mapas de saliências com melhores resultados. Zhang et al. [220] propuseram uma nova estratégia para usar rótulos dos componentes de classes (partes específicas de um objeto) para substituir os rótulos de nível de imagem em redes de convolução para grafos. Eles mostraram que regiões componentes possuem pequena variação de características intraclasse e não se sobrepõem interclasse, o que leva a uma melhor geração de resultados dos mapas de ativação através de sua combinação.

Durand et al. [56] propuseram um método fracamente supervisionado para classificação, localização e segmentação de objetos multiclasses sem restrição de domínio chamado aprendizado fracamente supervisionado de redes neurais convolucionais profundas (*Weakly supervIsed Learning of Deep Convolutional neurAl neTworks*, WILDCAT). Para isso, eles utilizaram uma DNN como extratora de características e, ao invés de fazer um *pooling* dos mapas de características para usar uma camada de classificação, geraram mapas de características por classes e fizeram um *pooling* para cada grupo de mapas, que foram ligados ao resultado da predição de cada classe. Ou seja, os mapas de características pertencentes ao grupo de uma classe só são atualizados pelo erro de classificação relativo a essa classe. Isso possibilitou a geração de mapas de saliências específicos para cada categoria e, com isso, a segmentação e a localização multiclasses com maior precisão. O WILDCAT é uma das arquiteturas utilizadas nesta tese para comparação de resultados nas Subseções 6.2.3 e 6.2.4.

Em relação à segmentação na área médica, Gondal et al. [64] usaram o CAM para aprender uma representação que permitisse localizar características discriminativas em imagens da retina e atingiram uma boa precisão para a tarefa de classificação. Na maioria das lesões de retinopatia diabética, as lesões são de tamanho extremamente pequenos nas imagens. Porém, apesar de mapas de ativação terem desempenho adequado na detecção de regiões unitárias e pequenas, seus resultados são ampliados para o tamanho original da imagem, tornando-se mapas de saliências, e tendem a mostrar saliências que extrapolam os limites das regiões requeridas, impossibilitando seu uso em uma segmentação refinada. Feng et al. [58] criaram um método muito parecido com o CAM para segmentar nódulos de câncer que, em sua primeira etapa, insere uma imagem em uma DNN, previamente treinada com rótulos de classificação, e gera mapas de saliências usando os mapas de características e pesos aprendidos durante o treinamento. Depois, usaram as mesmas imagens da etapa anterior para mesclá-las com seus mapas de saliências, como em uma conexão residual, criando uma segmentação refinada para nódulos cancerígenos.

Adiga et al. [2] trabalharam com detecção de câncer para desenvolver um método baseado na métrica de aprendizado de divisão e conquista (*divide and conquer metric learning*, DCML) [154], que simplifica a tarefa de aprendizado dividindo o espaço das *manifolds* em vários subespaços para criar máscaras de segmentação. Izadyyazdanabadi et al. [85] desenvolveram uma arquitetura, também para segmentação de câncer, que utilizou o CAM conjuntamente com classificadores intermediários em todos os blocos da arquitetura proposta, que melhorou seus resultados.

Alguns trabalhos sobre mapas de ativação utilizaram múltiplas instâncias, como é o caso de Choe et al. [40]. Eles estudaram os métodos fracamente supervisionados para localização e afirmaram que, ao considerar o número de indivíduos de treinamento e as diferentes

quantidades de supervisão, os métodos referentes aos mapas de ativação não progrediram significativamente desde o CAM. Além disso, eles mostraram a correlação entre o CAM e os métodos MIL enunciando que dado um conjunto de mapas de características produzidos por uma DNN, $F^k(X)$, $X \in \mathbb{R}^{W \times H \times c}$, $F^k(X) \in \mathbb{R}^{w \times h}$ com W e w as larguras, H e h as alturas, c os canais e k o número de mapas de características, o CAM pode ser visto como um problema de múltiplas instâncias em que cada $F^k_{i,j}$ é uma pontuação para o *patch* $x_{i,j} \subset X$ onde $x_{i,j} = X[i : i + \frac{W}{w}, j : j + \frac{H}{h}, :]$ do conjunto de mapa de características. Então, o *pooling* global efetua $\frac{1}{w \times h} \sum_{i,j} F^k_{i,j}$ unindo a pontuação de área dos $i \times j$ instâncias de cada mapa de características. Eles também propuseram um novo protocolo experimental que usa uma quantidade fixa de supervisão para que os métodos encontrem seus hiperparâmetros. Além disso, sugeriram considerar um paradigma que mistura algoritmos fracamente supervisionados e uma pequena quantidade de supervisão como em algoritmos de aprendizado com poucos exemplos (*few-shot learning*), mudando o paradigma do aprendizado fracamente supervisionado. Por fim, declararam a necessidade de considerar outras opções para resolver os problemas de algoritmos derivados do CAM.

Outros trabalhos utilizaram mapas de ativação com objetivo de gerar *patches*, como é o caso de Shen et al. [165], que propuseram uma DNN chamada classificador de múltiplas instâncias com reconhecimento global (*globally-aware multiple instance classifier*, GMIC) capaz de classificar câncer de mama utilizando mapas de saliências para construção de múltiplos *patches* em um projeto arquitetural de ponta a ponta. Eles utilizaram informações locais (*patches*) e globais (imagens originais) para construir um classificador que seleciona pequenas regiões cancerígenas. Shen et al. [166] estenderam o conceito do GMIC utilizando uma arquitetura de menor capacidade, mas com maior eficiência de memória para a imagem original, e uma arquitetura com maior capacidade para derivar características das predições dos *patches*. Métodos anteriores, criados para classificação de câncer, normalmente necessitavam de duas etapas para classificação. A primeira, a segmentação das áreas cancerígenas, era apenas um meio ou pré-processamento para a segunda, a classificação. A GMIC fez o mesmo processo sem a necessidade de rótulos de segmentação. A ideia é similar aos métodos propostos nesta tese, porém, por ser treinada como um classificador ponta a ponta, a GMIC pode não ser adequada para a classificação na CPB (Subseção 6.2.7). Liu et al. [112] propuseram uma arquitetura fracamente supervisionada que segue a mesma linha da GMIC, cujo fluxo é realizado em etapas, entretanto, ainda une as duas arquiteturas na etapa final, chamada mapas de ativações locais-globais (*global-local activation maps*, GLAM), também no contexto de detecção de câncer de mama. Primeiramente, eles treinaram a arquitetura global, depois a congelaram e treinaram a arquitetura local com os cortes produzidos pela arquitetura global e, na última etapa, refinaram o treinamento das duas arquiteturas ao mesmo tempo.

Wu et al. [199] fundiram os mapas de saliências provenientes dos mapas de ativação das camadas iniciais de redes neurais com mapas de saliências provenientes de mapas de ativação criados a partir de camadas finais. Isso permitiu obter localizações de objetos muito pequenos em imagens de sensoriamento remoto. Para diminuir a perda de objetos minúsculos ou densamente distribuídos, introduziram, em sua rede proposta, um módulo de ativação por divergência, que melhora a resposta dos mapas de ativação para as camadas iniciais, e um módulo de similaridade, que melhora a distribuição das predições nos mapas de ativação das camadas iniciais e diminui o ruído nos mapas finais. Hacıefendioğlu et al. [69] utilizaram mapas de ativa-

ção para classificar e localizar o acúmulo de gelo em hélices de turbinas de geração de energia eólica. Eles compararam os resultados de suas localizações fracamente supervisionadas com os resultados de segmentações feitas com uma rede U-Net [150].

Na classificação de pragas, Wang et al. [193], Luo et al. [122] e Chen et al. [37] propuseram o uso de mapas de ativação para diminuir a influência do plano de fundo no treinamento de classificadores de pragas. Yang et al. [210] utilizaram os mapas de ativação para recortar as RI e criar novas imagens para ajudar no treinamento de DNNs utilizando bases muito desbalanceadas, o que é comum em conjuntos de dados contendo pragas. Cap et al. [28] propuseram uma rede adversária generativa (*generative adversarial network*, GAN) utilizando mapas de ativação baseados em atenção para reproduzir sintomas apenas em áreas específicas de folhas, isto é, usaram os mapas de ativação para que os sintomas não fossem criados no plano de fundo. Kim et al. [88] utilizaram os mapas de ativação para localização de sintomas de mofo (míldio) da cebola. Chen et al. [33] utilizaram uma abordagem contendo atenção espacial para criar mapas de ativação com base na Equação 2.3. Liu et al. [117] aplicaram os mapas de ativação para entender o grau de atenção dado às partes mais discriminativas das imagens em diferentes modelos, bem como fizeram uma interseção e união dos resultados do CAM e de um método de segmentação derivado do GrabCut [152] para quantificar o grau de atenção que os modelos davam as pragas alvos.

Diferentemente dos métodos MIL, os mapas de ativação têm mais trabalhos relacionados à classificação de sintomas de doenças e pragas. Entretanto, precisam ser mais explorados, pois podem ajudar significativamente na localização sem a necessidade de rótulos para esse fim. Para sumarizar esta seção, a Tabela 3.2 lista todos os trabalhos citados sobre mapas de ativação.

3.3 Modelos de Redes Neurais Baseados em Atenção

As abordagens baseadas em atenção existem desde a década de 1980, porém, em 2014, essas abordagens tornaram-se um conceito fundamental em redes neurais e o estado da arte em muitas tarefas [46]. Os modelos baseados em atenção para DNNs surgiram com o processamento de linguagem natural (*natural language processing*, NLP). Bahdanau et al. [12] propuseram uma arquitetura para busca com redes neurais recorrentes (*recurrent neural network search*, RNNSearch), que melhorou os modelos de codificação e decodificação (*encoder-decoder*). A RNNSearch não precisava codificar uma sentença de entrada inteira em um único vetor de comprimento fixo, ela codificava a sentença de entrada em uma sequência de vetores, escolhendo um subconjunto desses vetores de forma adaptativa, aplicando e introduzindo a atenção, enquanto gerava a tradução. Logo após, a atenção em modelos de redes neurais se espalhou para outros campos do aprendizado de máquina [46].

Ainda no processamento de linguagem natural, Seo et al. [162] propuseram a rede de fluxo de atenção bi-direcional (*bi-directional attention flow*, BIDAF) para responder perguntas sobre o contexto de um parágrafo. Eles apresentaram um processo hierárquico de vários estágios que representava o contexto em diferentes níveis de granularidade e usava o mecanismo de fluxo de atenção bidirecional para obter uma representação de contexto das perguntas, isso sem nenhuma representação antecipada. Outros trabalhos ainda desenvolveram novos métodos de atenção em tarefas referentes ao processamento de texto, Yang et al. [211] classificaram documentos, Xiong et al. [204] treinaram modelos para responder questões, See et al. [160] sumarizaram textos.

Tabela 3.2: Trabalhos relacionados aos Mapas de Ativação.

Trabalhos	Informações Relevantes	Ano
Oquab et al. [136]	Janelas deslizantes para gerar a composição dos mapas de características finais.	2015
Pinheiro e Collobert [143]	Mapa de características por classes.	2015
Teh et al. [176]	Rede baseada em atenção para calcular a localização dos objetos.	2016
Zhou et al. [227]	Trabalho inicial sobre os mapas de ativação e produção do CAM.	2016
Durand et al. [56]	WILDCAT com mapa de características específicas por classes.	2017
Feng et al. [58]	Conexão residual entre mapas de características e imagens.	2017
Gondal et al. [64]	Método fracamente supervisionado para localização de lesões de retinopatia diabética.	2017
Kwak et al. [92]	SP-CAM, a melhoria do CAM com superpixels.	2017
Selvaraju et al. [161]	Grad-CAM, a extensão do CAM usando gradientes.	2017
Ahn e Kwak [4]	Cálculo da afinidade entre áreas das imagens e geração de rótulos de segmentação.	2018
Izadyyazdanabadi et al. [85]	Todos os blocos usando CAM para segmentação de câncer.	2018
Nguyen et al. [135]	Melhoria do CAM para o aspecto temporal T-CAM.	2018
Zhu et al. [233]	IAM e melhoria do CAM para segmentação da instância inteira.	2019
Rony et al. [151]	Revisão de localização de objetos fracamente supervisionados para a medicina.	2019
Shen et al. [165]	Produção do GMIC, arquitetura ponta a ponta usando características globais e locais, que escolhe *patches* através de mapas de ativação.	2019
Adiga et al. [2]	Uso do aprendizado de divisão e conquista para detecção de câncer.	2020
Chen et al. [32]	Melhoria do CAM para o aspecto temporal SA-CAM.	2020
Choe et al. [40]	Correlação entre o CAM e MIL, mapas de ativação não progrediram significativamente desde o CAM e uso de quantidade fixa de supervisão para melhorar os mapas de ativação.	2020
Wang et al. [188]	Melhoria do CAM referente a pontuações de classificação Score-CAM.	2020
Wang et al. [193]	Mapas de ativação para diminuir a influência do plano de fundo no treinamento de imagens com pragas.	2020
Cap et al. [28]	Uso de mapas de ativação em GANs.	2020
Kim et al. [88]	Uso de mapas de ativação na localização de mofo da cebola.	2020
Shen et al. [166]	Evolução do GMIC para melhorar os resultados e melhorar os tempos de inferência.	2021
Liu et al. [112]	Evolução do GMIC chamada GLAM.	2021
Luo et al. [122]	Mapas de ativação para diminuir a influência do plano de fundo no treinamento de imagens com pragas.	2021
Chen et al. [33]	Mapa de ativação baseado em atenção para classificação de pragas.	2021
Yang et al. [210]	Ajuda para recortar as imagens no treinamento de bases totalmente desbalanceadas.	2021
Chen et al. [37]	Mapas de ativação para diminuir a influência do plano de fundo nas predições.	2021
Liu et al. [117]	Mapa de ativação como medida para verificar quais áreas os modelos mais usavam para a classificação.	2022
Sun et al. [170]	Mostraram que os mapas de ativação respondem melhor a inferencia em partes das imagens e sua junção cria saliências mais precisas.	2022
Zhang et al. [220]	Rótulos dos componentes de classes para substituir os rótulos de nível de imagem.	2022
Wu et al. [199]	Fundiram os mapas de saliências provenientes de mapas de ativação das camadas iniciais e finais para diminuir a perda de objetos em sensoriamento remoto.	2022
Hacıefendioğlu et al. [69]	Classificação e localização do acúmulo de gelo em hélices de turbinas eólicas.	2022

Mnih et al. [130] foram os primeiros a aplicar atenção em trabalhos voltados à classificação de imagens e produziram o modelo de atenção recorrente (*recurrent attention model*, RAM), que foi capaz de extrair e processar informações de imagens ou vídeos selecionando uma sequência de regiões de alta resolução adaptativamente. Os modelos RAM podem ser treinados usando métodos de aprendizado por reforço para entender regras específicas de tarefas, enquanto seus recursos computacionais podem ser controlados. Jaderberg et al. [86] desenvolveram a *spatial transformer network* (STN), que continha um novo módulo feito para permitir explicitamente a manipulação espacial de dados na rede. Esse módulo tem a possibilidade de ser inserido em arquiteturas convolucionais existentes e consegue atribuir a capacidade de transformar ativamente mapas de características, sem supervisão de treinamento ou mudanças no processo de otimização. RAM e STN foram arquiteturas pioneiras com base na atenção visual humana [46]. Raffel e Ellis [145] propuseram um modelo simplificado de atenção que fosse aplicável a redes neurais totalmente conectadas e demonstraram que o modelo resultante poderia resolver os problemas sintéticos de memória de longo prazo existentes na época.

Diferentemente das DNNs com módulos ou camadas de atenção, há arquiteturas totalmente atencionais, como os transformadores neurais (*transformers*) [182], e as redes de atenção base-

adas em grafos (*graph attention networks*, GAT) [183]. As *transformers* foram propostas como redes baseadas apenas no mecanismos de atenção ou, mais precisamente, na autoatenção. As GATs são redes que contêm camadas autoatencionais para resolver as deficiências de métodos anteriores baseados em convoluções sobre grafos ou suas aproximações. Outros trabalhos baseados em *transformers* foram desenvolvidos, inclusive o trabalho de Child et al. [39], que trouxe a proposta de esparsidade (*sparse transformers*) para aplicar os transformadores às imagens e diminuir o processamento.

Mehta et al. [127] buscaram diminuir a quantidade de processamento das *transformers* e propuseram a DeLight (*deep and light-weight transformer*). A DeLight oferece desempenho semelhante ou superior aos modelos baseados em *transformers*, alocando parâmetros de forma mais eficiente.

Dosovitskiy et al. [55] propuseram os transformadores visuais (*visual transformers*, ViT). Eles mostraram que ViTs não precisam depender de CNNs, o que era comum em trabalhos anteriores de NT e imagens, ou seja, um transformador puro aplicado diretamente a sequências de *patches* de imagens pode ter um desempenho muito bom em tarefas de classificação de imagens. Um modelo ViT atinge excelentes resultados se é pré-treinado em grandes quantidades de dados e, subsequentemente, adaptado a bases pequena ou média. Bazi et al. [16] propuseram um método de classificação de cenas em sensoriamento remoto baseado em ViT, o que propiciou a identificação de lugares com tamanhos muito pequenos relativamente ás imagens de satélites, como campos de aviação, praias, estacionamentos de carros e áreas de esportes.

A atenção também foi aplicada em tarefas que misturam múltiplas fontes, como textos e imagens, ou textos e vídeos. Xu et al. [206] introduziram um modelo baseado em atenção que aprendia automaticamente a descrever o conteúdo de imagens. Eles mostraram como os modelos eram capazes de aprender a corrigir onde focavam nas RI ao mesmo tempo que geravam as palavras correspondentes na sequência de saída. O trabalho de Xu et al. [206] foi um dos precursores e, posteriormente, muitos outros também aplicaram o conceito de atenção para tarefas conjuntas, como Tan et al. [175] para reconhecimento de emoções, Anderson et al. [7] e Bin et al. [21] para sumarização de imagens e vídeos e Zhang et al. [224] para respostas de questões visuais.

Praveen e Menon [144] e Li et al. [99] trabalharam com aplicação de atenção em representações temporais de imagens multiespectrais de sensoriamento remoto. Praveen e Menon [144] classificaram coberturas de solo com um novo mecanismo de atenção espectral bidirecional, obtido de uma rede recorrente bidirecional de memória curta e loga (*bidirectional long short-term memory*, *bidirectional* LSTM [226]), que é computacionalmente eficiente e capaz de diversificação adaptativa de informação espectral. A rede consegue selecionar bandas espectrais que contêm as informações mais importantes. Li et al. [99] uniram imagens hiperespectrais e imagens que detalham as elevações de terrenos como entradas de dois ramos de uma rede. Eles aplicaram um método de atenção nas múltiplas camadas espectrais para escolher as informações multiespectrais mais importantes e um método de atenção espacial nas imagens com informações de elevação de terreno. Então, uniram as características dos dois ramos para classificar a cobertura de terrenos em cidades.

Considerando as camadas em DNNs, a arquitetura chamada EfficientNet-B0 [173], a rede base mais utilizada no decorrer desta tese, é composta por blocos convolucionais para dispositivos móveis de gargalo invertido (*inverted bottlenecks*) [78] e camadas de atenção por compressão e excitamento (*squeeze-and-excitation*, SE) [79]. Os blocos de SE foram os primeiros

a adotar o modelo de atenção como camadas de DNNs em 2017, entretanto, Hu et al. [79] tiveram seu trabalho publicado apenas em 2020. As camadas de compressão e excitação calculam os parâmetros relativos à atenção para cada mapa de características resultante de um bloco convolucional de forma adaptativa. Assim, essas camadas modelam explicitamente as interdependências entre os canais (os mapas de características mais importantes recebem maior pontuação). Hu et al. [79] mostraram que as camadas de atenção podem ser empilhadas para formar novas arquiteturas.

Woo et al. [196] exploraram a atenção espacial por camadas proposta por Chen et al. [34]. Para calcular a atenção para cada uma delas, eles aplicaram um *pooling* na dimensão de profundidade do mapa de características e, para calcular a atenção espacial, aplicaram operações de *average-pooling* e *max-pooling* ao longo do eixo das camadas e as concatenaram para gerar um descritor de características. A atenção por camada foca no "que" é significativo dada uma imagem de entrada, enquanto a atenção espacial foca "onde" é uma parte informativa. Chen et al. [36] propuseram um bloco de atenção dupla, o qual agrega e propaga características globais informativas de todo o espaço, no caso de imagens, ou espaço-temporal, no caso de vídeos, permitindo que camadas de convoluções subsequentes acessem características de todo o espaço de forma eficiente. O componente foi projetado com um mecanismo de dupla atenção, em que a primeira parte reúne características de um espaço em um conjunto compacto e a segunda parte seleciona e distribui de forma adaptativa a atenção.

Wang et al. [185] construíram uma rede de atenção residual (*residual attention networks*, RAN) empilhando módulos de atenção. Os pesos de atenção da rede são marcados de forma adaptativa dependendo do número de camadas residuais. Em cada módulo de atenção distinto, estruturas fazem a rede focar em diferentes aspectos das características. Zhang et al. [225] projetaram blocos de atenção locais e não locais para extrair características que capturassem as dependências de longo alcance entre os pixels e reforçar a atenção nas partes mais desafiadoras de imagens. Seu projeto objetivou a restauração de alta qualidade em imagens. Yao e Wu [212], com um conceito similar ao de Chen et al. [34] mas com implementação completamente diferente, propuseram um algoritmo de classificação de imagens combinando atenção por canais (cada mapa de características após uma convolução) e atenção espacial (Equação 2.3). As duas formas de atenção foram combinadas sequencialmente em submódulos depois de cada bloco de convoluções na rede neural proposta.

Recentemente, em 2022, Guo et al. [68] propuseram um novo módulo de autoatenção para permitir o uso de correlações autoadaptáveis e de longo alcance, evitando os problemas que ViTs e CNNs carregam, como o uso de imagens como vetores, a aplicação de cálculos com complexidade quadrática, a exclusão da adaptabilidade por canais e a utilização de filtros espaciais que avaliam apenas localmente. Eles apresentaram uma nova rede, baseada em um módulo chamado de atenção de núcleo grande (*large kernel attention*, LKA), chamada rede de atenção visual (*visual attention network*, VAN), que melhorou o estado da arte para classificação, localização e segmentação de objetos. Majid et al. [125] melhoraram arquiteturas tradicionais de aprendizado de máquina ao usar atenção espacial para classificação da existência de fogo em imagens. Além disso, utilizaram o Grad-CAM como método para mostrar a localização dos focos de incêndios nas imagens do trabalho.

Nas tarefas relacionadas a pragas e sintomas de doenças, Liu et al. [113] propuseram um módulo de atenção seguindo a mesma ideia de atenção espacial de Woo et al. [196]. Em vez de usarem pesos para gerar mapas de saliências provenientes de mapas de ativação diretamente,

eles empregaram camadas em sequência para produzir os mapas de saliências. Primeiro, eles aplicaram uma convolução 1×1 para compactar as camadas de mapas de características em apenas um e depois usaram um filtro 7×7 procedido por uma convolução transposta e uma multiplicação elemento a elemento (Equação 2.3). O método destacou insetos em equipamentos de aquisição de imagens de pragas (armadilhas) para fazer a detecção e classificação multiclasse. Wang et al. [186] exploraram o mecanismo de atenção por camada de Hu et al. [79] em cada bloco de convolução de projeção e bloco residual para abordar a detecção e contagem de pragas em campo. Eles introduziram o conjunto de dados *In-Field Pest in Food Crop* (IPFC) (não disponível), que contém 17.192 imagens de pragas de campo.

Zeng e Li [216] propuseram uma rede neural convolucional de autoatenção (*self-attention convolutional neural network* , SACNN) para melhorar a extração de características de manchas de doenças em diferentes culturas e classificá-las. A necessidade de atenção é justificada, pois as RI relativas aos sintomas são muito pequenas. A SACNN é formada por uma rede principal, que extrai as características globais da imagem, e uma rede de autoatenção, que obtém as características locais da área da lesão, ou seja, segue o mesmo principio de características globais (imagem inteira) e características de granularidade mais finas (apenas características das regiões com sintomas). Pelo mesmo motivo, precisão limitada para lidar com pragas com escalas muito pequenas, Wang et al. [190] introduziram um mecanismo de atenção em redes residuais para obter características mais ricas, especialmente características detalhadas de pequenas RI. Então, aplicaram esse conceito em redes de proposição de regiões (*region proposal network*, RPN) [148] para melhorar a qualidade das regiões propostas.

Na classificação de pragas e sintomas na área de agricultura, a atenção foi validada como uma forma de melhorar a extração de características locais de pequenas regiões, inclusive esta tese corrobora com estes resultados. O uso de mapas de ativação baseados em atenção é um modo de melhorar as classificações de RI pequenas e minúsculas. Todos os trabalhos citados nesta seção foram listados na Tabela 3.3.

3.4 Automatização do Manejo Integrado de Pragas Utilizando o Aprendizado Profundo

Recentemente, muitas aplicações têm utilizado aprendizado profundo para trabalhar com imagens em diversas áreas [23, 129, 158, 184]. Entretanto, trabalhos mais antigos sobre automatização do manejo integrado de pragas foram desenvolvidos em redes neurais totalmente conectadas ou algoritmos clássicos de aprendizado de máquina, como Deng et al. [52] usando o SVM, Sankaran et al. [157] com KNN, Larios et al. [93] usando florestas aleatórias (*random forests*), Hernández-Rabadán et al. [75] utilizando classificadores bayesianos, entre outros [200, 201]. Quando as CNNs começaram a se difundir, os primeiros objetivos de pesquisas foram o desenvolvimento e melhoria de técnicas de coleta de dados e a criação de bases de dados mais adequadas, pois o treinamento de algoritmos de aprendizado profundo exige uma grande quantidade de dados representativos da área.

Barbedo [13] e Barbedo et al. [15] descreveram as dificuldades para criar bases de dados e treinar métodos de aprendizado de máquina que efetivamente fossem aplicáveis em campo. Eles classificaram os problemas referentes ao uso das bases de dados em extrínsecos e intrínsecos ao reconhecimento em imagens. Os problemas extrínsecos são problemas que influenciam os

Tabela 3.3: Trabalhos relacionados aos Modelos Baseados em Atenção.

Trabalhos	Informações Relevantes	Ano
Mnih et al. [130]	Os primeiros a aplicar atenção em trabalhos voltados à classificação de imagens, produzindo o RAM.	2014
Bahdanau et al. [12]	RNNSearch foi a primeira rede a aplicar atenção em texto.	2015
Jaderberg et al. [86]	STN foi a rede com módulo feito para permitir explicitamente a manipulação espacial de dados na rede.	2015
Xu et al. [206]	Descrição do conteúdo de imagens.	2015
Raffel e Ellis [145]	Atenção em redes totalmente conectadas.	2016
Seo et al. [162]	BIDAF para responder perguntas sobre o contexto de um parágrafo.	2016
Yang et al. [211]	Classificação de documentos.	2016
Xiong et al. [204]	Treinamento de modelos para responder questões.	2016
Wang et al. [185]	RAN produziu um bloco de atenção para a ResNet.	2017
Vaswani et al. [182]	NT uma arquitetura totalmente baseada em autoatenção. (*transformers*).	2017
Veličković et al. [183]	GAT uma arquitetura de grafos totalmente baseada em autoatenção.	2017
Chen et al. [34]	SCA-CNN atenção por camadas e espacial.	2017
Anderson et al. [7]	Sumarização de imagens.	2018
Bin et al. [21]	Sumarização de vídeos.	2018
Zhang et al. [224]	Respostas a questões visuais.	2018
Woo et al. [196]	Atenção espacial por camadas, foco em o "que" e "onde".	2018
Chen et al. [36]	Bloco de atenção dupla.	2018
Tan et al. [175]	Reconhecimento de emoções.	2019
Tan e Le [173]	Proposição das *EfficientNets*.	2019
Zhang et al. [225]	Blocos de atenção locais e não locais.	2019
Liu et al. [113]	Módulo de atenção baseado em Woo et al. [196].	2019
Yao e Wu [212]	Uso de atenção por canais e espacial ao mesmo tempo.	2019
Child et al. [39]	*Transformers* esparsos para lidar com imagens.	2019
Hu et al. [79]	Blocos SE de compressão e excitamento.	2020
Wang et al. [186]	Exploraram o mecanismo de atenção por camada de Hu et al. [79].	2020
Zeng e Li [216]	SACNN rede convolucional combinada com uma de autoatenção.	2020
Dosovitskiy et al. [55]	ViT como arquitetura totalmente baseada em atenção.	2021
Bazi et al. [16]	Uso de ViT para sensoriamento remoto.	2021
Mehta et al. [127]	DeLight aloca parâmetros de forma mais eficiente.	2021
Wang et al. [190]	Mecanismo de atenção em redes residuais para RPN.	2021
Correia e Colombini [46]	Analisaram cerca de 650 artigos.	2022
Praveen e Menon [144]	Classificaram coberturas de solo com um novo mecanismo de atenção espectral bidirecional.	2022
Li et al. [99]	Uniram imagens hiperespectrais e imagens que detalham as elevações de terrenos.	2022
Guo et al. [68]	Atenção de núcleo grande (LKA) e a rede de atenção visual (VAN).	2022
Majid et al. [125]	Classificação da existência de fogo com uso de atenção espacial.	2022

resultados, mas não participam diretamente do reconhecimento, enquanto os problemas intrínsecos são diretamente relacionados ao reconhecimento e mostram a existência de infestações que, mesmo sendo diferentes, têm sintomas visualmente parecidos, têm diferentes sintomas em estações climáticas e fases fenológicas diferentes, apresentam sintomas diferentes em folhas, caules e frutos, e, quando em conjunto, ocasionam sintomas totalmente diferentes dos originais.

Nachtigall et al. [132] e Tan et al. [174] foram os primeiros a investigar o uso de arquiteturas de aprendizado profundo no domínio da classificação de doenças. Tan et al. [174] treinaram uma CNN para identificar lesões na casca de maçãs e depois a usaram, sem retreinamento, para identificação de lesões em melões. Cruz et al. [47] usaram transferência de aprendizado e injeção de contexto para condicionar o aprendizado das arquiteturas e conseguiram influenciar suas camadas a aprender conceitos relativos ao reconhecimento de doenças. Com essas estratégias, eles classificaram o estresse causado por falta de nutrientes e por agentes causadores de doenças em oliveiras. Muitos outros trabalhos avaliaram CNNs conhecidas em suas bases, com pouca ou nenhuma modificação [14, 60, 111], não trazendo muita evolução para a área computacional.

Hughes e Salathé [81] criaram uma base de dados de imagens, chamada PlantVillage, que consiste em 55.000 imagens (capturadas em laboratórios) de sintomas de doenças em folhas. Mohanty et al. [131] treinaram seus modelos na PlantVillage e mostraram que é possível uti-

lizar as técnicas de aprendizado profundo no reconhecimento de doenças em imagens RGB e tons de cinza. Wang et al. [187] usaram imagens de maçãs com e sem podridão (*Physalospora malorum*) da base PlantVillage para treinar CNNs variadas da literatura e classificar as imagens em "sem doenças", "grau de doença inicial", "grau de doença médio" e "grau final". A base de dados PlantVillage foi a primeira grande base de dados pública na área de classificação de doenças (sintomas), porém, mais tarde, Ferentinos [59] introduziu uma nova versão da PlantVillage com 87.848 imagens, que não foi disponibilizada ao público.

Em relação à classificação de doenças via dispositivos móveis e embarcados, Petrellis [141] propôs um aplicativo para classificar sintomas em vinhedos. Ele usou técnicas de processamento visual que mesclavam o uso de histograma de cores e segmentação via um limiar global para reconhecer, isolar e diagnosticar as lesões com base em seu número, tamanho e cores. Ferentinos [59], vislumbrando também dispositivos móveis, propôs seu uso em campo mas não apresentou experimentos. Bhandari et al. [20] trabalharam com veículos aéreos não tripulados (VANTs) e veículos terrestres não tripulados (VTNTs) para reconhecer doenças em alfaces usando câmeras multiespectrais, hiperespectrais e RGB. Eles apresentaram um método para localização de alfaces, baseado em sua reflectância, que também reconhecia condições necessárias para aplicação automática de insumos. Xing e Lee [203] aplicaram diferentes redes neurais no reconhecimento de dez diferentes pragas das tangerinas e avaliaram o tempo de inferência de um modelo da Inception-ResNet-V3 [172] (436,8 MB de tamanho), melhor classificador de seu trabalho, em uma GPU (GTX 1080Ti, 12GB) e uma CPU (Snapdragon 835, 6GB) para dispositivos móveis. Eles concluíram que é difícil armazenar o modelo em um dispositivo com recursos limitados e o tempo de execução nas diferentes plataformas é muito diferente (GPU = 14 ms/imagem e CPU = 117 ms/imagem). Esses fatores impactam o uso de modelos grandes de DNNs em dispositivos móveis e embarcados.

Antes de 2018, poucos trabalhos exploraram CNNs para classificação de pragas diretamente, como os trabalhos de Liu et al. [118] e Fuentes et al. [60]. Normalmente, os sintomas provocados pelas pragas eram usados para detectar sua presença, como na classificação de doenças. Liu et al. [118] usaram mapas de saliência construídos por histogramas para identificar a variação de cores entre pragas e o fundo das imagens e, assim, conseguiram detectar a localização de insetos para criar uma banco de dados com 5.136 imagens. Fuentes et al. [60] usaram sintomas de pragas e imagens de moscas brancas. Alfarisy et al. [5] coletaram da Internet 4.511 imagens de pragas do arroz e as classificaram. Xing e Lee [203] criaram uma base de dados contendo 10 tipos de insetos das tangerinas, incluindo o psilídeo (*Diaphorina Citri*, o vetor do *Greening*) e avaliaram várias arquiteturas em suas imagens.

Uma das maiores bases de dados para classificação de pragas e insetos foi introduzida por Wu et al. [197]. A base IP102 consiste em 102 classes e 75.222 imagens. Os autores aplicaram diferentes CNNs (AlexNet, GoogleNet, VGGNet e ResNet) para relatar seus resultados. Diversos trabalhos avaliaram seus resultados na IP102 (Subseção 6.1.3). Ren et al. [147] e Liu et al. [115] melhoraram o desempenho da classificação na IP102 modificando os blocos residuais da ResNet, construindo duas novas arquiteturas chamadas rede residual de reuso de características (*feature reuse residual network*, FR-ResNet [147]) e rede residual profunda de fusão de múltiplos ramos (*deep multi-branch fusion residual network*, DMF-ResNet [115]). Xu e Wang [207] usaram o conjunto de dados da IP102 para demonstrar o uso da XCloud, uma plataforma em nuvem proposta para facilitar o uso de inteligência artificial.

Nanni et al. [133] também utilizaram a base IP102 e propuseram um classificador baseado

na fusão entre métodos de criação de mapas de saliências e CNNs. Eles empregaram os mapas de saliências para destacar os pixels mais relevantes das imagens de pragas durante o treinamento de múltiplos modelos que ao final foram unidos em um *ensemble*. Eles utilizaram três métodos diferentes de detecção de saliências e mais as imagens originais provindas da base de dados criada por Deng et al. [52], que contém 563 imagens de pragas divididas em 10 espécies diferentes, e aplicaram o mesmo método na IP102. Como evolução desse trabalho, Nanni et al. [134] propuseram um novo *ensemble* de conjuntos de redes neurais baseadas em diferentes topologias, que foram treinadas com diferentes variantes do otimizador Adam [89], e o avaliaram na IP102. Eles propuseram dois novos otimizadores que introduziram um fator de escala na taxa de aprendizado do Adam.

Ayan et al. [9] utilizaram três modelos de DNNs para produzir um *ensemble* de redes chamado soma das máximas probabilidades (*sum of maximum probabilities ensemble*, SMPEnsemble), que foi avaliado na IP102 e na D0 (uma pequena base de pragas proposta por Xie et al. [201]). Os modelos da Inception-v3 [171], Xception [41] e MobileNet-v2 [156] foram unidos por meio de uma estratégia de soma dos máximos das probabilidades, o que aumentou o desempenho de classificação. Depois disso, esses modelos foram unidos utilizando uma estratégia de votação ponderada em um *ensemble* com pesos produzidos por algoritmos genéticos (*genetic algorithm ensemble*, GAEnsemble). Os pesos dessa votação foram determinados por um algoritmo genético que considerou a taxa de sucesso e a estabilidade preditiva dos três modelos.

Ung et al. [178] desenvolveram um *ensemble* de diferentes tipos de DNNs, inclusive, dentre as arquiteturas, escolheram uma que utiliza redes de pirâmides de mapas de características (*feature piramid networs*, FPN) [109], uma arquitetura recorrente baseada em atenção (*recurrent neural networks*, RAN) [185] e uma arquitetura adaptada por eles, a rede de atenção multirramos e multiescalas (*multi-branch and multi-scale attention networks*, MMAL-Net [217]), que utiliza atenção e características com granularidade mais fina, para classificar pragas em duas bases de dados públicas, a D0 e IP102. Os resultados mostraram que, ao se combinar os modelos dessas arquiteturas, eles conseguiram chegar ao melhor resultado de acurácia para D0 e IP102.

Thenmozhi e Reddy [177] classificaram pragas em três bases de dados publicamente disponíveis, a NBAIR, do *The National Bureau of Agricultural Insect Resources* da Índia, Xie1 e Xie2, que contêm 40, 24 e 40 classes de insetos, respectivamente, entretanto, não citaram o número total de imagens da base. Li et al. [100] construíram uma base de dados pública com 10 espécies de insetos de grãos estocados contendo 3.757 imagens e 159.238 anotações de *bounding boxes*. Chen et al. [31] usaram o mecanismo de busca de imagens do Google para coletar 700 imagens de quatro pragas, incluindo ácaros da família dos purpúreos (*Tetranychidae*). Eles usaram CNNs para classificar as imagens capturadas por sensores em campo. Zhang et al. [222] modificaram uma rede de cápsulas para adicionar uma forma conhecida de atenção e classificar oito pragas comuns das culturas, como brocas do milho, mariposas, joaninhas, pulgões, cigarrinhas e alguns tipos de lagartas.

He et al. [74] mostraram um sistema de classificação e localização de pragas para dispositivos móveis utilizando um modelo baseado no detector de disparo único (*single shot detector*, SSD) e compararam com outros modelos para dispositivos móveis, visando tempo de inferência e acurácia. Eles adotaram esse modelo em sua plataforma móvel para permitir que agricultores pudessem usar o programa que diagnostica pragas de culturas oleaginosas em tempo real e fornece sugestões sobre o seu controle. Além disso, projetaram uma base de dados de imagens com

12 espécies típicas de pragas chinesas contendo 3.022 imagens e 5.016 anotação de *bounding boxes* para objetos. Chudzik et al. [43], também visando dispositivos móveis, apresentaram um estudo para detecção de gafanhotos em tempo real e uma nova base de dados pública contendo 3.578 imagens e 4.406 objetos anotados chamada base de detecção de gafanhotos (*GrassHopper deteCtIon Dataset*, GHCID). Li e Yang [106] propuseram um método de reconhecimento de pragas de algodão para aprendizado com poucos exemplos (*few shot learning*), que precisa de apenas poucas imagens brutas para o treinamento, e, como caso de estudos, utilizaram um dispositivo embarcado produzido com uma placa programável para validar seus modelos treinados na base NBAIR e, em outra base, que foi denominada base de Li (*Li's dataset*).

Alguns trabalhos já começaram a produzir resultados para tarefas ligadas a pragas muito pequenas, como Li et al. [102], que buscaram detectar e localizar pulgões em imagens coletadas em campo. Eles defenderam que os métodos de detecção de pragas baseados em redes neurais não seriam satisfatórios para lidar com pulgões, pois as redes existentes não estariam aptas a lidar com objetos constituídos de pequenas regiões e com um número denso desses objetos em um mesmo local. Lins et al. [110] também trabalharam com pulgões e apresentaram uma ferramenta para automatizar sua contagem e classificação usando métodos de processamento de imagem, visão computacional e DNNs para classificar características locais. Pei et al. [139] argumentaram que métodos para detectar pulgões baseados em DNNs seriam insatisfatórios porque os pulgões seriam pequenos e, em geral, espacialmente distribuídos. Eles utilizaram descritores locais binários para encontrar regiões que possivelmente teriam pulgões e produzir localizações aplicando redes convolucionais para essas regiões. Ma et al. [124] aplicaram equipamentos relacionados à internet das coisas (*Internet of Things*, IoT) da agricultura para obter um grande número de imagens de pragas pequenas, localizadas através de um modelo de DNNs para extração de características de granularidade fina.

Li et al. [104] trabalharam com a localização de pequenas moscas-brancas e tripes em armadilhas adesivas dentro de estufas. Eles construíram um modelo de localização de pragas baseado na RPN [148] com um módulo específico para identificação de RI muito pequenas. Wu e Xu [198] propuseram um método fracamente supervisionado para segmentar folhas e frutos e classificar imagens de sintomas de doenças com base nas regiões localizadas. Bereciartua-Pérez et al. [19] utilizaram DNNs e imagens coletadas de celulares para contar moscas-brancas através da geração de mapas de densidades, que informam a localização e distribuição dos objetos, o que ajudou na regressão. As moscas ocupavam de 20 a 30 pixels em imagens com tamanho 4000×6000 pixels. Li et al. [103] usaram os mapas de densidade para contar pulgões.

Wang et al. [191] produziram uma grande base de dados, chamada *AgriPest*, para localização de pequenas pragas na natureza capturando 49.707 imagens de quatro culturas diferentes com 14 espécies de pragas. Liu et al. [116] criaram outra grande base de dados, chamada base de doenças de plantas com 271 classes (*plant disease dataset 271*, PDD271), mas contendo sintomas de doenças, com 220.592 imagens com 271 classes. Ambas bases de dados não estão públicas.

Outros trabalhos recentes sobre pragas e sintomas de doenças foram apresentados nas seções anteriores e propuseram o uso de MIL (Seção 3.1), mapas de ativação (Seção 3.4) e modelos baseados em atenção (Seção 3.2). Quando a pesquisa referente a esta tese iniciou, o desenvolvimento de DNNs especificamente para a área de classificação, localização e segmentação de pragas e doenças era escasso, com trabalhos iniciais utilizando redes pré-estabelecidas da literatura. Nos últimos três anos, os trabalhos se diversificaram, inclusive com propostas de redes

que se beneficiaram do aprendizado fracamente supervisionado. A aplicação de mecanismos de atenção, mapas de ativação e redes que usam características mais refinadas (*patches* ou características derivadas destes), como as propostas nesta tese, cresceram significativamente neste período. Isso mostra que os processos e métodos apresentados no Capítulo 5 estão alinhados com as pesquisas sobre pragas e sintomas de doenças. A Tabela 3.4 lista todos os trabalhos citados nesta seção.

Tabela 3.4: Trabalhos relacionados à Agronomia.

Trabalhos	Informações Relevantes	Ano
Larios et al. [93]	Florestas aleatórias para detectar larvas de moscas aquáticas.	2010
Sankaran et al. [157]	KNN e espectroscopia para Huanglongbing (HLB).	2011
Hernández-Rabadán et al. [75]	Classificadores Bayesianos para segmentar sintomas de doenças.	2014
Hughes e Salathé [81]	Criação da base de dados PlantVillage.	2015
Xie et al. [200]	Histogramas com aprendizado por múltiplos núcleos (MKL).	2015
Barbedo [13]	Descrição das dificuldades para criar bases de dados.	2016
Barbedo et al. [15]	Geração de uma base de sintomas de doenças.	2016
Liu et al. [118]	Mapas de saliência construídos por histogramas para gerar uma base de dados.	2016
Mohanty et al. [131]	Uso da PlantVillage. Resultados muito altos 99,35% de acurácia.	2016
Tan et al. [174]	CNNs para lesões na casca de maçãs e melões.	2016
Nachtigall et al. [132]	Identificação de sintomas de desordens em maçãs.	2016
Bhandari et al. [20]	Veículos não tripulados para sintomas de doenças em alfaces.	2017
Cruz et al. [47]	Transferência de aprendizado e injeção de contexto para sintomas de doenças.	2017
Fuentes et al. [60]	Classificação de sintomas de doenças e pragas em tomates.	2017
Petrellis [141]	Aplicativo móvel para classificar sintomas em vinhedos.	2017
Wang et al. [187]	Uso da imagens de maçãs da PlantVillage.	2017
Liu et al. [111]	Identificação de sintomas de doenças em maçãs.	2018
Alfarisy et al. [5]	Classificação e coleta de imagens da Internet de pragas do arroz.	2018
Barbedo [14]	Avaliação de bases e transferência de aprendizado para sintomas de doenças.	2018
Deng et al. [52]	SVM para classificar insetos pragas em sua base de dados.	2018
Ferentinos [59]	Nova versão da PlantVillage.	2018
Xing e Lee [203]	Criação de base de dados com insetos das tangerinas.	2018
Xie et al. [201]	Criação de base de dados D0.	2018
Li et al. [100]	Base de dados pública com espécies de insetos de grãos estocados.	2019
He et al. [74]	Sistema de classificação e localização de pragas para dispositivos móveis.	2019
Ren et al. [147]	Modificação dos blocos residuais da ResNet para classificação na IP102.	2019
Xu e Wang [207]	Uso da IP102 na plataforma online. Sem informações sobre a metodologia.	2019
Wu e Xu [198]	Método fracamente supervisionado para classificação de sintomas.	2019
Thenmozhi e Reddy [177]	Classificação de pragas na NBAIR.	2019
Wu et al. [197]	Construção da base de dados IP102 e propostas de métodos de avaliação.	2019
Li et al. [102]	Construção de uma base com pulgões e classificação.	2019
Chen et al. [31]	Treinamento com imagens da Internet e classificação com sensores em campo.	2020
Chudzik et al. [43]	GHCID e dispositivos móveis para classificação de gafanhotos.	2020
Li e Yang [106]	Reconhecimento de pragas de algodão para aprendizado com poucos exemplos.	2020
Liu et al. [115]	Modificação dos blocos residuais da ResNet para uso na IP102.	2020
Nanni et al. [133]	Uso da IP102 e *ensemble* de classificadores aplicados a mapas de saliências.	2020
Ayan et al. [9]	Uso da IP102 e *ensemble* de classificadores com pesos aprendidos por algoritmos genéticos.	2020
Lins et al. [110]	Contagem de pulgões usando processamento de imagens, visão computacional e aprendizado de máquina.	2020
Pei et al. [139]	Localização de pulgões usando ORB e redes convolucionais.	2020
Ung et al. [178]	Uso da IP102 e *ensemble* de classificadores com arquiteturas de piramide de características, arquiteturas baseadas em atenção e características com granularidade mais finas.	2021
Li et al. [104]	Modificação da RPN para localização de moscas-brancas e tripes em armadilhas adesivas.	2021
Wang et al. [191]	Construção da base de dados chamada AgriPest.	2021
Liu et al. [116]	Construção da base de dados PDD271.	2021
Ma et al. [124]	Classificação de imagens coletadas com imagens de equipamentos para IoT aplicadas à agronomia.	2021
Nanni et al. [134]	Uso da IP102 e *ensemble* de classificadores usando diferentes tipos de otimizadores.	2022
Zhang et al. [222]	Modificaram uma rede de cápsulas para adicionar uma forma conhecida de atenção.	2022
Bereciartua-Pérez et al. [19]	Classificação da existência de fogo com uso de atenção espacial.	2022
Li et al. [103]	Contagem de pulgões.	2022

3.5 Resumo Comparativo

Os trabalhos relacionados usaram algoritmos e métodos que estão inseridos no mesmo contexto desta tese de doutorado. Entretanto, a maior parte deles contém requisitos, necessidades e objetivos diferentes da metodologia apresentada nas Seções 4 e 5. A Tabela 3.5 compara todos os trabalhos que influenciaram diretamente no desenvolvimento deste doutorado, como o trabalho de Zhou e Xu [230] e sua indicação de que as instâncias não são independentes e identicamente distribuídas (i.i.d.). As propostas desta tese de doutorado também usam uma correlação entre as instâncias, porém, uma correlação de ordem que é considerada somente na avaliação das *bags* e não no treinamento das arquiteturas.

Os trabalhos de Hashimoto et al. [70] e Choe et al. [40] usaram mapas de ativação e MIL. Hashimoto et al. [70] geraram mapas de saliências para segmentação de câncer em uma arquitetura MIL, mas a metodologia aqui proposta os emprega ativamente em sequência para conseguir as instâncias adequadas ao treinamento de uma CNN. O trabalho de Choe et al. [40] descreveu os mapas de ativação como uma forma de aprendizado do tipo de múltiplas instâncias. No entanto, uma instância como descrita no MIL é um conjunto independente e limitado em relação às *bags*. Os mapas de ativação usam as características providas pelas convoluções, ou seja, usam características de todas as instâncias ao mesmo tempo.

Como apontado por Dietterich et al. [54], as redes neurais totalmente conectadas não são adequadas para o uso em métodos MIL, então, a maior parte delas foi modificada, pois o treinamento com instâncias é diferente do treinamento típico de uma DNN. Então, esta tese de doutorado propõe, para classificação de imagens binárias com RI muito pequenas, utilizar DNNs conjuntamente com métodos MIL sem a necessidade de adaptação da arquitetura por causa das instâncias (por exemplo, cálculo diferenciado do erro e *pooling* de pontuações dos resultados de instâncias), como é usual na maior parte dos trabalhos. As propostas usando MIL e mapas de ativação desta tese utilizam o MIL como suporte ao treinamento totalmente supervisionado, ou seja, a arquitetura é usada em sua versão original, sendo treinada com a forma típica do aprendizado totalmente supervisionado, mas com instâncias. Propõe-se também, para a tarefa multirrótulos, a aplicação de um técnica chamada adaptação de domínio não supervisionada do trabalho de Ganin e Lempitsky [61]. Essa técnica é utilizada para diminuir os problemas de alinhamento de distribuição entre os subconjuntos (treinamento e validação) da base de dados proposta nesta tese (Seção 4.1). Esse tipo de metodologia não foi aprofundada nos trabalhos relacionados porque não foram propostas técnicas para a área. O trabalho de Ganin e Lempitsky [61] foi aplicado pelo fato de mostrar um bom resultado para a classificação multirrótulos.

Dois trabalhos que desenvolveram arquiteturas para o aprendizado fracamente supervisionado foram comparados com os métodos propostos, *Attention-based Deep* MIL [83] e WILDCAT [56]. Essas arquiteturas foram utilizadas ou modificadas para agirem como parâmetros de comparação para as propostas desta tese nas Subseções 6.2.3 e 6.2.4.

A base de dados de Wu et al. [197] foi utilizada como uma segunda base de dados para avaliar as propostas desta tese de acordo com a literatura. Os trabalhos de Chen et al. [34] e Woo et al. [196] influenciaram a técnica de atenção proposta por este trabalho na Subseção 5.2.2. Zhou et al. [227], conjuntamente com Selvaraju et al. [161], embasaram a metodologia inicial sobre mapas de ativação e a importância das RI, inclusive, o trabalho de Zhou et al. [227] foi utilizado para comparar as localizações fracamente supervisionadas. Para gerar os rótulos de localização chamados *bounding boxes* presentes nas Subseções 6.2.5 e 6.2.6, o método de

criação de localizações de Lu et al. [121] foi utilizado como base.

Tabela 3.5: Trabalhos mais importantes para esta tese de doutorado.

Trabalhos	Informações Relevantes	Ano
Dietterich et al. [54]	Desenvolveram o trabalho inicial sobre MIL e que influenciou nas futuras modificações em DNNs.	1997
Zhou e Xu [230]	Afirmaram que instâncias não são independentes e identicamente distribuídas.	2007
Ganin e Lempitsky [61]	Adaptação de domínio não supervisionada.	2015
Zhou et al. [227]	Propuseram o CAM e o uso de mapas de ativação.	2016
Durand et al. [56]	Desenvolveram a arquitetura WILDCAT.	2017
Selvaraju et al. [161]	Criaram o Grad-CAM.	2017
Chen et al. [34]	Propuseram o uso de atenção espacial conjuntamente com atenção por canais.	2017
Lu et al. [121]	Propuseram a localização fracamente supervisionada para sintomas de doenças.	2017
Woo et al. [196]	Propuseram o uso de convoluções 1×1 para geração dos mapa de ativação baseados em atenção.	2018
Ilse et al. [83]	Desenvolveram a *Attention-based Deep* MIL.	2018
Wu et al. [197]	Produziram a IP102.	2019
Choe et al. [40]	Afirmaram que existe uma correlação entre o CAM e o MIL.	2020
Hashimoto et al. [70]	Propuseram um *ensemble* para escolher a escala mais propícia para as instâncias.	2020

Capítulo 4

Bases de Dados

Este capítulo apresenta uma das contribuições deste trabalho, a *Citrus Pest Benchmark* (CPB) (Seção 4.1), e outras bases da literatura que contêm pragas variadas (Seção 4.2). Nas subseções que apresentam a CPB, são descritas a forma como os dados foram coletados em campo (Subseção 4.1.1), as necessidades e os requisitos levantados com especialistas (Subseção 4.1.2) e os ruídos removidos da base CPB (Seção 4.1.3), que culminaram na base de dados *Noiseless Citrus Pest Benchmark* (NCPB).

4.1 *Citrus Pest Benchmark*

A base de dados de referência para pragas cítricas (*Citrus Pest Benchmark*[1], CPB) foi criada como uma opção aos conjuntos de dados da literatura e contempla desafios relacionados às necessidades reais da coleta de dados em campo. A CPB contém imagens divididas em sete classes (seis espécies de ácaros e uma classe negativa) que foram coletadas via um dispositivo móvel com uma lente de aumento acoplada (Figura 4.1). Para isso, foi utilizado um equipamento Samsung Galaxy A5 com câmera de 13 MP e uma lupa com ampliação óptica de 60×, equipada com iluminação LED branca e LED ultravioleta.

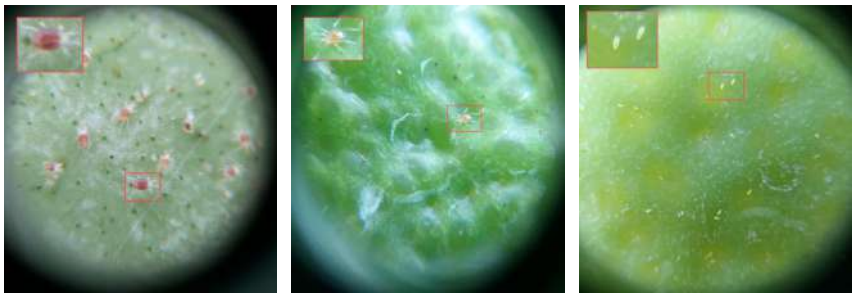
(a) (b) (c) (d)

Figura 4.1: Lupa para dispositivos móveis. (a) Lupa com luz de LED e ultravioleta. (b) Lupa acoplada a um dispositivo móvel. (c) Imagem real de um inseto. (d) Imagem aumentada do inseto. Figura reproduzida de Bollis et al. [24].

As áreas ocupadas pelas espécies de ácaros são muito pequenas em relação ao tamanho total das imagens, conforme ilustrado na Figura 4.2, que é um dos desafios para aplicação de métodos clássicos da literatura. Outro desafio da base é a presença de ruído em uma parte significativa de

[1] https://github.com/edsonbollis/Citrus-Pest-Benchmark

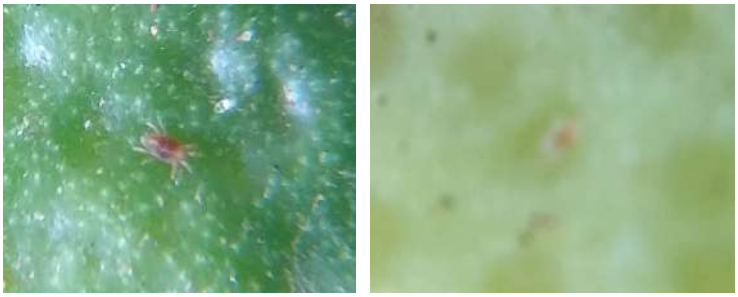
suas imagens, devido à superfície de vidro biconvexa presente na lupa e as situações climáticas externas a coleta (Figura 4.3).



(a) Ácaro purpúreo (b) Ácaro fitoseídeo (c) Ácaro da ferrugem

(d) Ácaro da leprose (e) Ácaro branco (f) Ácaro rajado

Figura 4.2: Ácaros capturados através de ampliação óptica de 60×. Os ácaros são destacados no lado superior esquerdo das imagens. Figura reproduzida de Bollis et al. [24].



(a) Imagem nítida (b) Imagem desfocada

Figura 4.3: Exemplos de ácaros da leprose da base *Citrus Pest Benchmark*. Figura reproduzida de Bollis et al. [24].

A CPB contém 10.816 imagens categorizadas em: (i) 1.902 imagens com ácaros purpúreos (*Panonychus citri*, *Eutetranychus banksi*, *Tetranychus mexicanus*), a maior de todas espécies e que produzem um sintoma amarelado nas folhas e nos frutos (Figura 4.2a); (ii) 1.426 imagens com ácaros fitoseídeos (*Euseius citrifolius*, *Iphiseiodes zuluagai*), o ácaro predador que ajuda a controlar outros ácaros (Figura 4.2b); (iii) 1.386 imagens com ácaros da ferrugem (*Phyllocoptruta oleivora*), responsáveis pelo sintoma de enferrujamento e perdas significativas das culturas de citros (Figura 4.2c); (iv) 1.750 imagens com ácaros da leprose (*Brevipalpus phoenicis*), um vetor do vírus da leprose (Figura 4.2d); (v) 806 imagens com ácaros brancos (*Polyphagotarsonemus latus*), responsáveis por causar manchas brancas nos frutos (Figura 4.2e); (vi) 696 imagens com ácaros rajados, que não trazem perdas significativas de culturas, no entanto, são regularmente vistos em campo (Figura 4.2f); e (vii) 3.455 imagens da classe negativa, que não contêm nenhum dos tipos de ácaros presentes nas outras classes.

As imagens foram divididas em três conjuntos (treinamento, validação e teste), contendo aproximadamente 60%, 20% e 20% dos ácaros de cada classe, totalizando 6.380, 2.239 e 2.197 imagens, respectivamente. Algumas das classes são muito similares entre si para olhos não treinados. Além disso, as diferenças de luminosidade e ampliação tornam a base de dados desafiadora. O problema multirrótulos torna as tarefas mais interessantes, visto que 5% das imagens (599) contêm até três classes simultaneamente.

A base foi construída com imagens de tamanho 1.200×1.200 pixels para as classes negativa e positiva, mais precisamente, 7.361 imagens de ácaros e 3.455 imagens negativas. As tarefas de classificação binária e multirrótulos foram utilizadas nos experimentos desta tese.

4.1.1 Coleta de Dados

Devido ao aumento da lupa e ao tamanho e velocidade dos ácaros (indivíduos da coleta) percorrendo folhas e frutos, a coleta de imagens e procura de ácaros em campo é difícil. Pode-se comparar esse tipo de coleta com a aquisição de imagens aéreas de carros e caminhões em movimento tomadas de veículos aéreos não tripulados, em que a tarefa final é identificar a marca do veículo. Para os ácaros, todas as características como cor, tamanho das patas, formato do corpo, tamanho do corpo e fases evolutivas dos indivíduos (durante as fases eles podem mudar de cor, de tamanho e forma) são relevantes para a identificação [67].

Além dos desafios elencados, depois da coleta em campo, a aferição das imagens dos ácaros pelas inspetoras é um problema recorrente, pois muitas imagens são descartadas porque inspetoras com anos de experiência se confundem ao classificar dois ácaros distintos em fases diferentes de crescimento. Isso mostra a dificuldade das inspetoras ao fazer sua análise em campo e a necessidade de técnicas automatizadas que auxiliem esse trabalho.

Para gerar a CPB, várias visitas foram necessárias à Fazenda São José, localizada na cidade de Rio Claro, no Estado de São Paulo. As visitas ocorreram no período de março de 2018 a janeiro de 2019, contemplando todas as estações do ano, pois o aparecimento de algumas espécies de ácaros é sazonal. As coletas ocorreram guiadas pelas inspetoras que aplicam o Manejo Integrado de Pragas (MIP) (Figura 1.1), onde inspeções programadas foram realizadas nas áreas das unidades de produção, também chamadas de talhão. Normalmente, uma unidade de produção contém até 1.000 árvores cítricas, divididas em grupos organizados em ruas (árvores alinhadas) que inspetoras percorrem escolhendo amostras, não próximas às bordas, para analisar frutos, folhas e caule. Depois de coletar os dados de uma árvore em uma rua, elas passam para a próxima árvore trinta plantas à frente e continuam o processo até o final da rua. Ao final da rua, elas continuam a inspeção na terceira rua a partir da rua finalizada (três por trinta). Os ácaros na CPB foram obtidos utilizando as amostras examinadas pelas inspetoras.

A coleta procedeu da seguinte forma: (i) o especialista deste trabalho acompanhava as inspetoras enquanto elas percorriam as ruas colhendo amostras e fazendo suas análises; (ii) as inspetoras apontavam as classes dos ácaros incidentes nas amostras e sua localização; o especialista coletava as imagens enquanto armazenava os tipos de pragas incidentes; (iii) após a coleta, as inspetoras revisavam as anotações feitas; (iv) em caso de dúvida, a imagem era descartada.

4.1.2 Escolha das Pragas

Durante os primeiros passos deste projeto, foi necessário identificar os requisitos e necessidades básicas dos agricultores e inspetores de pragas em campo, verificando quais pragas seriam as mais importantes no processo diário do MIP e como elas seriam identificadas. Então, agrônomos e técnicos agrícolas da Fazenda São José, segundo a incidência e anotações das coletas na lavouras, analisaram a intensidade de aparecimento das pragas mais comuns e elencaram as cinco mais interessantes para se classificar automaticamente e ajudar as inspetoras em suas jornadas diárias. Foi durante esse período que se verificou a necessidade diária de coleta de dados sobre ácaros na lavoura e que é necessário a utilização de lupas para sua identificação.

Então, a ideia de formulação da CPB surgiu da interação entre os pesquisadores deste projeto e gerenciadores reais da coleta de dados sobre pragas da citricultura. A base, mesmo sendo coletada usando lupas acopladas a dispositivos móveis, trouxe os requisitos de tamanho dos ácaros e problemas de luminosidade e foco que são constantes na aplicação manual do MIP. Além disso, ao efetuar o processo de coleta com as inspetoras, verificou-se a existência de um tipo de ácaro que aparecia constantemente nas análises, mas não é considerada uma praga por não causar danos extensos (ácaro rajado, ilustrado na Figura 4.2f), totalizando seis tipos de ácaros na CPB. Pretende-se ainda, após a finalização do doutorado, utilizar a CPB como parte do treinamento das novas inspetoras que irão a campo que, mesmo com novas ferramentas para o reconhecimento dos ácaros, necessitarão do conhecimento referente às suas características.

4.1.3 *Noiseless Citrus Pest Benchmark*

Para avaliar o impacto da presença de ruído no treinamento de algoritmos utilizando a base de dados CPB, as imagens ruidosas foram removidas dos conjuntos de treinamento e validação. O critério utilizado para remover as imagens ruidosas foi baseado na visibilidade dos ácaros nas imagens. Em outras palavras, os exemplares coletados foram analisados um a um e, dentre eles, foram escolhidas imagens sem o auxílio ou uso de técnicas de processamento. Essa parte da CPB foi denominada de CPB sem ruído (*Noiseless Citrus Pest Benchmark*, NCPB).

A abordagem manual foi seguida porque técnicas computacionais existentes, como a diagonal do Laplaciano e a razão de coeficientes wavelets, falharam em separar imagens ruidosas de exemplares sem ruído. Quando a aplicação desses métodos automáticos foi avaliada, muitas imagens ainda continham ácaros indistinguíveis no conjunto possivelmente sem ruídos. Ao mesmo tempo, uma grande parte de ácaros facilmente detectáveis visualmente foram removidos deste mesmo grupo. O problema com essas técnicas é que elas consideram a imagem inteira de entrada, se um borramento cobrir uma parte significativa de uma imagem e os ácaros ainda estiverem visíveis, os métodos geralmente solicitarão para remover estas imagens. Por outro lado, se o ruído for apenas sobre o ácaro, os métodos indicarão que a imagem deve ser mantida. Após esses resultados negativos, cada imagem foi revisada manualmente, sendo removida se os contornos corporais dos ácaros não são reconhecíveis.

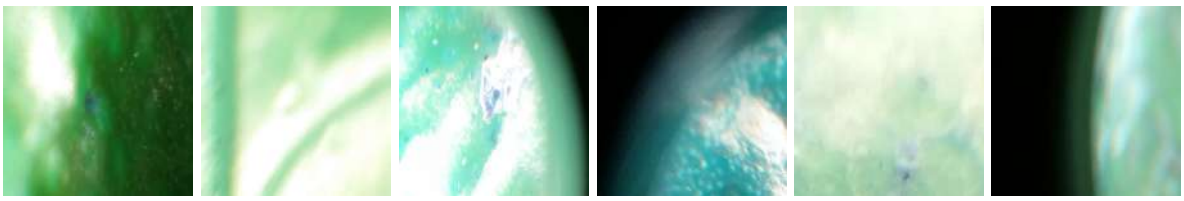
O conjunto de treinamento da NCPB diminuiu para 3.243 imagens positivas e 1.532 imagens negativas, totalizando 4.775. O conjunto de validação da NCPB atingiu 1.142 imagens positivas e 524 negativas, totalizando 1.666. A NCPB contém aproximadamente 75% de cada conjunto de treinamento/validação da CPB. A Tabela 4.1 exibe uma comparação do número de imagens da base CPB e da NCPB. Assume-se o conjunto de testes da NCPB como o conjunto de testes

da CPB, porque o objetivo é simular uma situação de campo e, para isso, o conjunto de testes deve ser fiel às imagens coletadas no campo. Não é realista avaliar os métodos propostos em um contexto sem qualquer ruído. Portanto, o conjunto de testes não foi modificado.A Figura 4.4 faz uma comparação qualitativa dos problemas de luminosidade e borramento presentes na base de dados CPB e que foram removidos na NCPB.

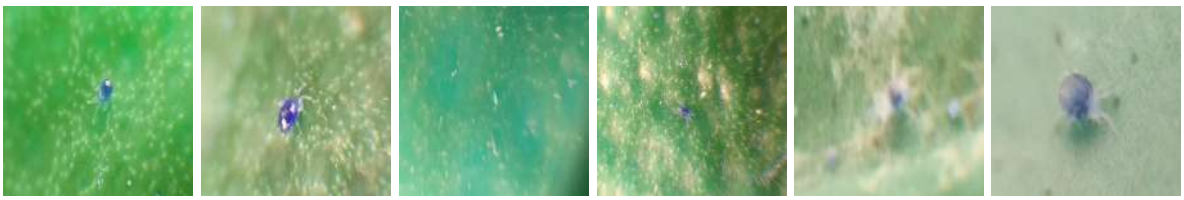
Tabela 4.1: Descrição dos conjuntos de dados da CPB.

Base de Dados	Treinamento	Validação	Teste	Total
CPB [24]	6.380	2.239	2.197	10.816
NCPB [25]	4.775	1.666	–	6.441

(a) Borramento



(b) Luminosidade



(c) Sem ruídos

Figura 4.4: *Patches* de ácaros da CPB: cada captura usa ampliação de 60x, cada divisão é 1/9 do tamanho da imagem original. As cores dos ácaros nas imagens originais estão próximas da faixa vermelha, mas essa faixa se altera após seu processamento.Essa mudança nas cores foi adotada para diferenciar as imagens originais da CPB e as imagens que foram geradas com algum tipo de processamento (resultados do trabalho). Assim, imagens RGB indicam originalidade e BGR indicam algum tipo de processamento. Este conjunto de imagens é proveniente de resultados para esta tese. (a) Imagens borradas. (b) Imagens afetadas por luminosidade. (c) Imagens consideradas sem ruídos.

A CPB é a contribuição esperada referente à C1: *"um novo conjunto de dados de referência (benchmark) para o problema do reconhecimento de pragas dos citros, onde pequenas regiões de interesse contendo diferentes tipos de ácaros estão presentes nas imagens"* e é resultado da busca pelo objetivo O1: *"criar uma base de dados com diferentes tipos de ácaros da citricultura paulista"*. A criação da NCPB como subconjunto da CPB é parte dos esforços para responder à questão de pesquisa Q4: *"os ruídos como luminosidade e borramento, presentes na captura de imagens em campo, afetam o treinamento dos modelos de redes neurais profundas?"*.

4.2 Comparação entre Bases de Dados

Tabela 4.2: Bases de dados existentes em trabalhos da literatura. Os trabalhos aqui descritos utilizaram primeiramente as bases ou são os desenvolvedores delas. O campo "quantidade" indica o número de imagens que foi relatada no texto do trabalho, "nome" o nome dado para a base, "tipo" descreve se a base é pública ou proprietária e "ano" mostra o ano em que a base foi utilizada nos trabalhos ou criada. "N.A." indica ausência de informações. A linha destacada corresponde à base proposta neste trabalho.

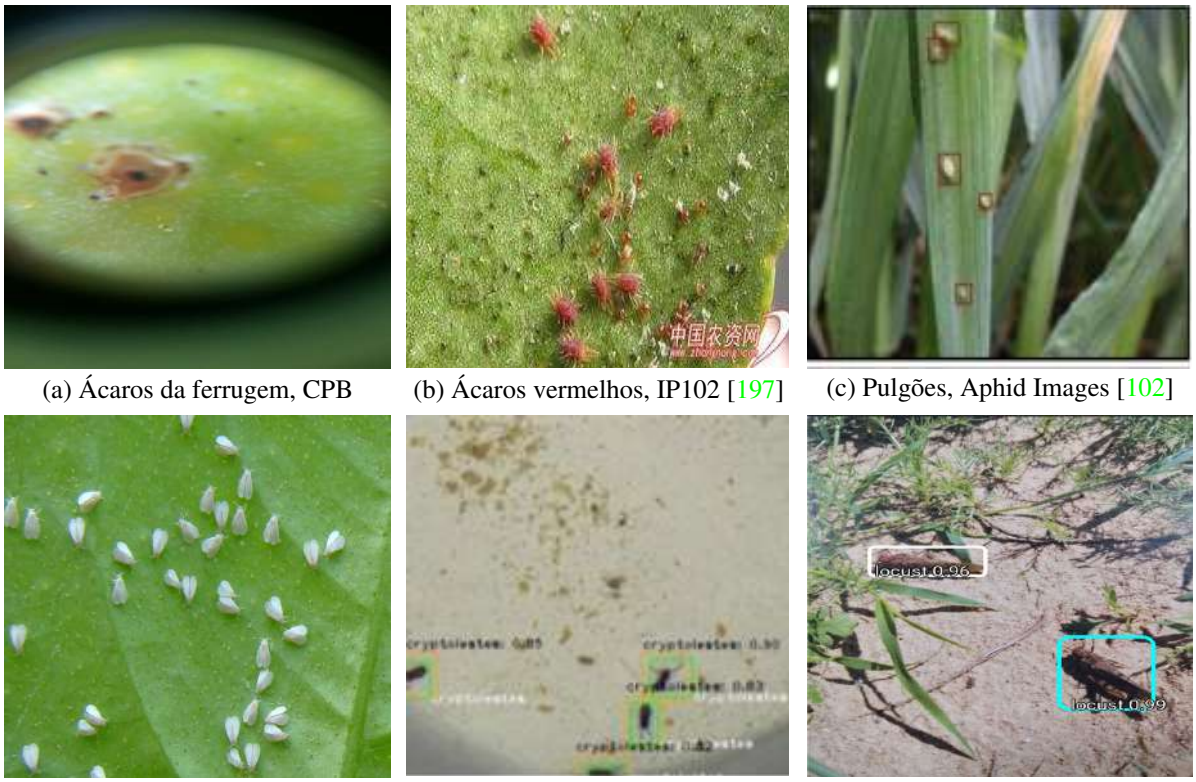
	Bases de Dados			
Trabalhos	Quantidade	Nome	Tipo	Ano
Xie et al. [200]	312	D1, D2	N.A.	2015
Liu et al. [118]	5.136	Pests ID	N.A.	2016
Xie et al. [201]	312	D0, D3	N.A.	2018
Alfarisy et al. [5]	4.511	Paddy Pest Images	N.A.	2018
Deng et al. [52]	563		pública	2018
Xing e Lee [203]	5.247	Pest Tangerine	N.A.	2018
He et al. [74]	3.022	Oilseed Rape Pest Imaging	pública	2019
Li et al. [100]	3.757	RGBInsect	pública	2019
Li et al. [102]	2.200	Aphid Images	N.A.	2019
Liu et al. [113]	88.670	PestNet	N.A.	2019
Thenmozhi e Reddy [177]	N.A.	NBAIR	pública	2019
Wu et al. [197]	75.222	IP102	pública	2019
Lins et al. [110]	120	Rhopalosiphum padi	pública	2020
Chen et al. [31]	700		N.A.	2020
Chudzik et al. [43]	3.578	GHCID	pública	2020
Bollis et al. [24]	**10.816**	**CPB**	**pública**	**2020**
Pei et al. [139]	361		N.A.	2020
Li et al. [104]	1.941		pública	2021
Wang et al. [191]	49.707	AgriPest	proprietária	2021
Yang et al. [210]	18.391	RPDID	proprietária	2021
Liu et al. [117]	67.953	IP67	N.A.	2022

Trabalhos que geraram ou foram os primeiros a utilizar as bases de dados, descritos na Tabela 4.2, produziram propostas para várias culturas, inclusive para citricultura. Porém, alguns trabalhos não disponibilizaram publicamente as bases ou não mencionaram se elas estariam disponíveis para uso (na Tabela 4.2 estas situações foram marcadas como "N.A."). Atualmente, é mais comum publicar os dados. Bases mais antigas do que a base construída nesta tese têm menos de 10 mil imagens, talvez pela dificuldade de coleta ou falta de mão de obra especializada para anotação. Mais recentemente, um número maior de imagens tem sido usada nos trabalhos que objetivaram divulgar suas bases, provavelmente, para que redes neurais sejam melhor treinadas. As bases públicas para pesquisa e com grande número de imagens têm sido muito utilizadas por trabalhos posteriores, como a IP102 [197].

A base de dados IP102 foi escolhida como uma segunda base de dados para os experimentos desta tese. Até onde temos conhecimento, ela é a única base de dados grande, pública e

anotada que também contém ácaros entre suas pragas. Como mostra a tabela, ela é uma base contendo muitos exemplos, mas a maior parte deles mostra insetos ou ácaros salientes, ou seja, pragas que cobrem uma área maior proporcionalmente às imagens. A IP102 foi atualizada para sua versão 1.1, pois foram encontrados erros de anotações, duplicidades e imagens totalmente indistinguíveis em sua versão 1.0. Nesta tese, foram utilizadas as duas versões para reportar experimentos, então, foram designados os nomes IP102 1.0 e IP102 1.1. Para a tarefa de localização, é utilizada a IP102 1.0, pois as anotações referentes à localização na IP102 1.1 não estão disponíveis. Outros trabalhos usaram a IP102 (Subseção 6.1.3), mas não informaram quais versões foram utilizadas.

Na Figura 4.5, pode-se verificar uma comparação entre imagens de bases de pragas existentes e imagens da base CPB. A diferença entre as escalas e as dimensões das regiões de interesse nas imagens são notáveis. Enquanto na imagem da Figura 4.5a quase não se consegue enxergar os ácaros, eles são bem evidentes na imagem da Figura 4.5b. Nas Figuras 4.5c a 4.5f as pragas são salientes a olho nu. Esses são apenas alguns exemplos, deve-se ressaltar que a proporção do tamanho das pragas depende do tamanho dos insetos que foram escolhidos para as imagens e da distância em que as imagens foram adquiridas. Bases mais recentes, como as de Pei et al. [139], Wang et al. [191] e Li et al. [104], trouxeram pragas com corpos proporcionalmente menores, isto é, regiões de interesse de tamanhos mais parecidas com a base CPB. Mas, elas trazem pragas como pulgões e outros insetos macroscópicos, as quais não são abrangidas pelo escopo desta tese.



(a) Ácaros da ferrugem, CPB (b) Ácaros vermelhos, IP102 [197] (c) Pulgões, Aphid Images [102]

(d) Moscas brancas, NBAIR [177] (e) Insetos dos grãos, RGBInsect [100] (f) Gafanhotos, GHCID [43]

Figura 4.5: Comparação entre bases de dados de pragas existentes na literatura.

Capítulo 5

Metodologia Proposta

Este capítulo descreve os processos, métodos e algoritmos utilizados e desenvolvidos nesta tese. O primeiro processo apresentado é o método fracamente supervisionado de aprendizado por múltiplas instâncias guiado por mapas de saliências ou *MIL-Guided*[1] (Seção 5.1). O segundo processo é conhecido como método fracamente supervisionado de aprendizado por múltiplas instâncias baseado em atenção e guiado por mapas de saliências ou *Attention-based MIL-Guided*[2] (Seção 5.2). A terceira seção apresenta a adaptação de domínio não supervisionada e remoção do rótulo classe negativa para classificação (Seção 5.3). Ao final, são apresentadas as métricas utilizadas neste trabalho (Seção 5.4). Nas subseções, são apresentados o algoritmo de seleção de múltiplos *patches* baseado em mapas de saliências ou *Patch-SaliMap* (Subseção 5.1.5) e o método de avaliação ponderada (Subseção 5.1.6), os quais fazem parte do *MIL-Guided* e *Attention-based MIL-Guided*, e os mapas de ativação com dois pesos ou *Two-WAM* (Subseção 5.2.2), um requisito arquitetural para o *Attention-based MIL-Guided*.

5.1 *MIL-Guided*

Esta seção apresenta uma abordagem para um processo fracamente supervisionado aplicado a imagens com regiões de interesse proporcionalmente muito pequenas. O método fracamente supervisionado de aprendizado por múltiplas instâncias guiado por mapas de saliências (*weakly supervised multiple instance learning guided by saliency maps*, *MIL-Guided*) é descrito. Esta abordagem foi publicada no *Agriculture-Vision Workshop* da *Conference on Computer Vision and Pattern Recognition* (CVPR 2020) [24]. O *MIL-Guided* é um processo de aprendizado por múltiplas instâncias de classificação binária para simplificar o problema de classificação de ácaros.

A Subseção 5.1.5 detalha o método guiado por mapas de saliência chamado algoritmo de seleção de múltiplos *patches* baseado em mapas de saliências (*multi-patch selection strategy based on saliency maps*, *Patch-SaliMap*), uma estratégia de seleção de vários *patches* baseada nos maiores valores de saliências. A Subseção 5.1.6 define o método proposto para avaliar as imagens originais considerando seus *patches*. As etapas para o *MIL-Guided* são exibidas na Figura 5.1.

[1] https://github.com/edsonbollis/Weakly-Supervised-Learning-Citrus-Pest-Benchmark

[2] https://github.com/edsonbollis/Attention-based-MIL-Guided

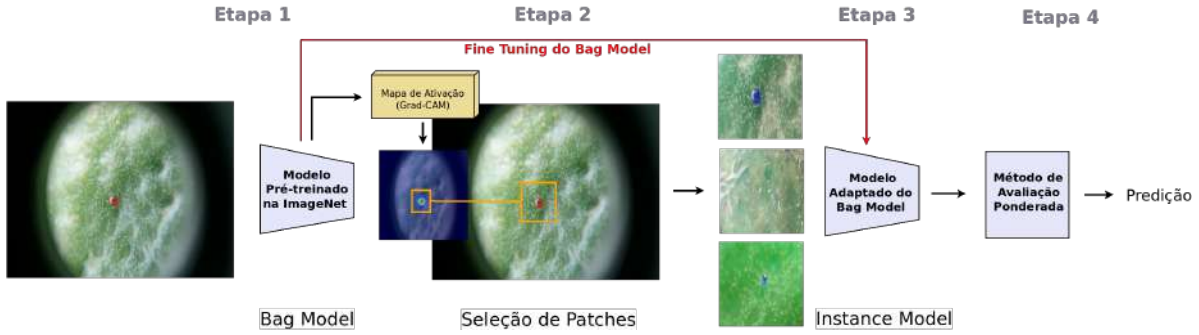


Figura 5.1: O processo consiste em quatro etapas. Na Etapa 1, uma DNN (pré-treinada na ImageNet) é treinada na CPB. Na Etapa 2, os mapas de saliências guiam a geração automática de *patches* das imagens. Na Etapa 3, os modelos são ajustados (*fine tuning* na tarefa de destino) de acordo com a abordagem MIL. Na Etapa 4, um algoritmo de avaliação ponderada para prever a classe das imagens é aplicado. Figura modificada de Bollis et al. [24].

O método proposto consiste em quatro etapas. A Etapa 1 refere-se à construção do *Bag Model* com modelo pré-treinado na ImageNet (Subseção 5.1.1), a Etapa 2 trata da geração das instâncias (Subseção 5.1.2), a Etapa 3 refere-se à construção do *Instance Model* por meio do *fine tuning* do *Bag Model* (Subseção 5.1.3) e a Etapa 4 trata do uso do método de avaliação ponderada (Subseção 5.1.4).

5.1.1 Etapa 1: Construção do *Bag Model* com Modelo Pré-treinado

Como descrito na Subseção 2.4.2, o MIL é um aprendizado fracamente supervisionando em que os dados de treinamento têm rótulos e são chamados de *bags* com $D_s = \{(X_i, y_i), y_i = f(X_i), i = 1, ..., n, X_i \in X\}$, e cada *bag* contém vários *patches* chamados *instâncias* com $\overline{X_i} = \{x_{ij}, j = 1, ..., k\} \subset X_i$, em que x_{ij} é parte de X_i, n é o número de imagens, k é o número de instâncias, e j é a numeração da instância em $\overline{X_i}$. Nesse contexto, na Etapa 1, o modelo da rede neural profunda (DNN), treinado em um conjunto de *bags* rotuladas, é chamado *Bag Model*.

O *Bag Model*, Etapa 1 da Figura 5.1, é o melhor modelo obtido com imagens originais e classifica a existência de RI com um mínimo de confiança possível, pois ele é utilizado na Etapa 2 como base para o algoritmo de "localização" dos ácaros e seleção de *patches*. RIs de tamanhos muito pequenas em imagens de treinamento, redimensionadas para tamanhos menores, não produzem características suficientemente robustas para predição e, provavelmente, são desprezadas ou desaparecem durante o *downsampling* das características. Por isso, nesta etapa, é necessário o uso de imagens grandes para treinar as DNNs. Por mais que as RIs ainda sejam pequenas em relação às imagens grandes, o *Bag Model* precisa distinguir minimamente se elas estão presentes. Então, para auxiliar o treinamento do *Bag Model*, utiliza-se um modelo pré-treinado na base ImageNet, que segundo Kornblith et al. [90], é um modelo para reconhecimento de características variadas, as quais dificilmente são aprendidas em bases com menos variabilidade nas classes e imagens.

5.1.2 Etapa 2: Geração das Instâncias

Na Etapa 2, são escolhidos *patches* de tamanho $d \times d$, com $d \in \mathbb{N}^+$ e $d \, mod \, 2 = 0$ (em que a operação $a \ mod \ b$ representa o resto da divisão do valor a pelo valor b), das *bags* através da seleção de subtensores das imagens originais, como é detalhado na Subseção 5.1.5. São utilizados os mapas de saliências das *Bag Models* para identificar áreas nas imagens onde há alta probabilidade de RIs serem localizadas. Em outras palavras, o algoritmo é aplicado em cada $X_i \in X$ para gerar $\{x_{ij}, j = 1, ..., k\}$ *patches* de X_i, com $k = 5$ e criar uma nova base de dados de instâncias $\overline{X} = \{x_{ij}, i = 1, ..., n, j = 1, ..., k\}$ sem rótulos para o MIL.

Na Etapa 2 da Figura 5.1, vê-se que o *Patch-SaliMap* usa o *Bag Model* para gerar os mapas de saliências através dos mapas de ativação do Grad-CAM [161]. Os mapas de saliências ressaltam as localizações que são utilizadas para escolha dos *patches* nas imagens originais, os quais são levados para a base de dados de instâncias. Então, a base de dados de instâncias, que é utilizada na Etapa 3, tem um número muito maior de imagens, mais precisamente $k \times n$ imagens, com a mesma resolução da base original, entretanto, com altura (d) e largura (d) menores que as originais. Nesta situação de escolha, alguns *patches* em imagens originalmente classificadas como positivas podem não conter ácaros.

5.1.3 Etapa 3: Construção do *Instance Model* Utilizando o *Bag Model*

Na Etapa 3, os rótulos relativos as *bags* são transferidos para suas instâncias (no MIL, a rotulação manual é feita apenas para as *bags*). Ou seja, se $y_i = f(X_i)$ for o rótulo de $X_i \in X$, então $f(x_{ij}) = y_i, \ x_{ij} \in \overline{X}$. Isso permite gerar a nova base com pseudorrótulos $\overline{D_s} = \{(x_{ij}, y_i), f(x_{ij}) = y_i, i = 1, ..., n, j = 1, ..., k, x_{ij} \in \overline{X}\}$. Em seguida, o *Bag Model* é retreinado utilizando as instâncias presentes em $\overline{X}$, explorando um esquema de transferência de aprendizado para o MIL entre as imagens originais e suas instâncias. Nessa fase do processo, são utilizadas cinco instâncias de cada *bag* com rótulos negativos e duas instâncias de *bags* positivas para diminuir o desbalanceamento dos dados e para diminuir a probabilidade de não existirem ácaros nas imagens com rótulos positivos. O melhor modelo desta fase recebe o nome de *Instance Model*, como pode ser visto na Etapa 3 da Figura 5.1.

Destaca-se que é possível usar o mesmo modelo sem alterações para imagens com tamanhos diferentes, pois existe um *pooling* global após a última camada convolucional para cada DNN. O *pooling* transforma um mapa de características de dimensões $w \times h \times c$ em um mapa de características de tamanho $1 \times 1 \times c$. Portanto, pode-se reutilizar o *Bag Model* para treinar os novos modelos, independentemente do tamanho das imagens. Entretanto, esse *pooling* não é obrigatório para o processo e, se uma arquitetura não o utiliza, pode-se usar um modelo pré-treinado na ImageNet para esta etapa.

5.1.4 Etapa 4: Uso do Método de Avaliação Ponderada

Na Etapa 4 da Figura 5.1, todos os modelos treinados em $\overline{D_s}$ também são avaliados em suas próprias imagens para produzir o resultado final para as *bags* de treinamento de D_s, conforme o Método de Avaliação Ponderada, descrito na Seção 5.1.6. O melhor modelo avaliado em $\overline{D_s}$ é chamado de *Instance Model* e fornece as probabilidades finais para cada instância de $\overline{D_t}$ e, depois de aplicado o Método de Avaliação Ponderada, para cada *bag* de D_t.

Esta etapa é somente um modo de avaliar várias instâncias que representam uma imagem original. O *Instance Model* é um modelo produzido por uma arquitetura sem modificações e que é utilizado independentemente do método de avaliação ponderada. É possível usar o *Instance Model* como um modelo simples para prever a classificação de vários *patches* de uma imagem e verificar se há ácaros em cada um separadamente. No caso da CPB, pode-se criar uma *grid* com k imagens de tamanho 1200×1200 em *patches* com tamanho $d \times d$, em que $k = \frac{1200 \times 1200}{d \times d}$ e $1200 \ mod \ d = 0$. Em seguida, cada um dos k *patches* gerados sem auxílio do *Patch-SaliMap* é analisado pelo *Instance Model* para inferir o resultado para a *bag* (imagem original).

5.1.5 Algoritmo de Seleção de Múltiplos *Patches* Baseado em Mapas de Saliências

O objetivo deste algoritmo é prover detalhes finos das imagens originais para a base de dados de instâncias, porque a maior parte dos ácaros dos citros não é facilmente visível a olho nu. Baseando-se nos mapas de saliências, ele seleciona *patches* significativos em imagens e, por esse motivo, foi chamado de Algoritmo de Seleção de Múltiplos *Patches* Baseado em Mapas de Saliências (*Patch-SaliMap*). O Algoritmo 1 descreve formalmente a estratégia. Há outros algoritmos para a seleção de múltiplas regiões de interesse, por exemplo, o *Mean Shift* [45]. Entretanto, como uma decisão de pesquisa, o foco foi mantido nos mapas de ativação e não no algoritmo de seleção das instâncias. O *Patch-SaliMap* é uma forma simples de escolha de possíveis regiões de interesse e produção de instâncias.

Seja $X_i \in X \subset \mathbb{R}^{h \times w \times 3}$ um tensor representando uma imagem, onde $h, w \in \mathbb{N}^+$ são a altura e largura de X_i. Seja $S : \mathbb{R}^{h \times w \times 3} \rightarrow \mathbb{R}^{h \times w}$ uma função que representa um mapa de saliências, em que $S(X_i)$ é o mapa de saliências de X_i. O *Patch-SaliMap* recebe como entradas $X_i, S(X_i), k, d$ e produz tensores $x_{ij} \in \overline{X} \subset \mathbb{R}^{d \times d \times 3}, j = \{1, ..., k\}$, em que $k \in \mathbb{N}^+$ é o número total de instâncias, $j \in \mathbb{N}^+$ é o índice das instâncias, e $d = 2 \times m, m \in \mathbb{N}^+$ é o tamanho de um lado de um *patch* quadrado.

O *Patch-SaliMap* usa o conhecimento prévio de que ácaros são pequenos o suficiente para serem capturados em imagens com tamanho igual à área coberta por um *patch* e produz k *patches* de tamanho $d \times d$ pixels. Como consequência, a probabilidade de se produzir instâncias com ácaros nos primeiros *patches* é maior do que a probabilidade dos *patches* finais, criando uma relação de ordem entre as instâncias.

Usar os valores máximos das células da matriz do mapa de saliências é uma boa opção para a inferência. No entanto, quando o *Instance Model* é treinado, as regiões de máximos locais para instâncias negativas geralmente trazem características fáceis de se aprender, o que pode tornar o algoritmo tendencioso em encontrar apenas essas características. Portanto, na geração de $\overline{X}$ foram produzidos cortes aleatórios x_{ij} em imagens $X_i \in X$ onde haviam rótulos negativos. Isso pode ter ajudado o *Instance Model* a aprender características mais diversificadas de imagens com rótulos negativos.

5.1.6 Método de Avaliação Ponderada

Para prever a classe de cada *bag* de D_s e D_t, propõe-se o uso do método de avaliação ponderada, que usa pesos estáticos para calcular a média ponderada entre as probabilidades das instâncias de uma mesma *bag*. Então, o método calcula a probabilidade final $P(.)$ a partir de cada proba-

Algoritmo 1 *Patch-SaliMap*

Entrada: $X_i, S(X_i), k, d$
Saída: x_{ij}

function PATCH_SALIMAP:
  $m = d/2$
  for $j := 1 : k$ **do**
    $a, b :=$ *calcula os índices com valores máximos de* $S(X_i)$
    if $a \pm m, b \pm m$ *está fora das bordas de* X_i **then**
      $a, b :=$ *corrige os índices a,b usando d*
    # calcula um novo *patch* ao redor de a e b
    new patch $:= X_i[a - m : a + m, b - m : b + m, :]$
    min $:=$ *calcula o valor mínimo de* $S(X_i)$
    # esconde a área recortada
    $S(X_i)[a - m : a + m, b - m : b + m, :] :=$ *min*
    $x_{ij}[j] :=$ *new patch*
return x_{ij}

bilidade $p(.)$ provida pela inferência do *Instance Model*. Assim, dado $X_i \in X, i = \{1, ..., n\}$ e $x_{ij} \in \overline{X}, j = \{1, ..., k\}$, para cada *bag* a probabilidade $P(.)$ é expressa na Equação 5.1.

$$P(X_i) = \frac{\sum_{j=1}^{k}(k - j + 1)p(x_{ij})}{\sum_{j=1}^{k}(k - j + 1)}. \tag{5.1}$$

O método de avaliação ponderada atribui um peso k maior à primeira instância x_{i1}, que intuitivamente é a escolha da saliência com maior probabilidade de conter ácaros, pois é o local em que a matriz de saliências carrega o maior valor. Isso acontece porque o *Patch-SaliMap* faz uma busca ordenada decrescente pelos índices relativos aos valores da matriz do mapa de saliência e, portanto, sempre o primeiro *patch* escolhido é o local com maior probabilidade de conter ácaros. Os valores subsequentemente escolhidos pelo *Patch-SaliMap* são menores do que o primeiro, portanto, o algoritmo atribui custos decrescentes aos próximos *patches* até o último *patch* escolhido receber peso igual ao valor 1.

O MIL-Guided é um dos métodos fracamente supervisionados da contribuição referente à C2: *"métodos fracamente supervisionados para classificação e localização de pragas"* e o *Patch-SaliMap* é o método referente à contribuição C4: *"uma estratégia eficaz de seleção de múltiplas regiões de interesse baseada em mapas de saliências para localizar automaticamente pragas dos citros"*.

5.2 *Attention-based MIL-Guided*

O método fracamente supervisionado de aprendizado por múltiplas instâncias baseado em atenção e guiado por mapas de saliências (*attention-based weakly supervised multiple instance learning guided by saliency maps, Attention-based MIL-Guided*) estende o *MIL-Guided* explorando mapas de ativação baseados em atenção. Esta abordagem foi publicada no periódico *Computers*

and Electronics in Agriculture (COMPAG 2022) [25].

A Subseção 5.2.1 mostra as modificações trazidas pelo *Attention-based MIL-Guided* e a Subseção 5.2.2 detalha a formulação matemática dos mapas de ativação com dois pesos (*Two-Weighted Activation Mapping*, *Two-WAM*). As principais motivações para a extensão do *MIL-Guided* são: (i) tornar seu uso mais adequado para dispositivos móveis, ou seja, gerar um modelo mais leve e não dependente de gradientes (o Grad-CAM usa os gradientes para gerar os mapas de ativação); (ii) aumentar a capacidade de encontrar regiões pequenas passiveis de conter ácaros, uma desvantagem do método Grad-CAM empregado no *MIL-Guided*; e (iii) obter uma melhor delimitação das regiões dos ácaros.

Por esses motivos, o *Attention-based MIL-Guided* utiliza um módulo ou camada de mapas de ativação baseados em atenção. Esse elemento arquitetural interno tem a capacidade de destacar corretamente as RIs, diminuir a influência de características relacionadas ao plano de fundo e inferir um número mais significativo de regiões que melhor se adaptem aos objetos de interesse.

5.2.1 Modificações e Evolução

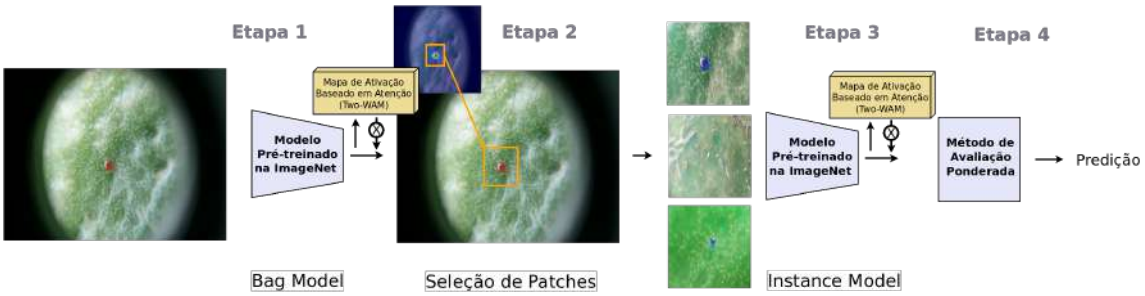
Esta abordagem baseia-se na atenção e antecipa a criação dos resultados dos mapas de ativação. Para isso, traz a inferência dos mapas de saliências para o primeiro passo do processo. Ao contrário do fluxo do *MIL-Guided*, o *Bag Model* do *Attention-based MIL-Guided* produz localizações referentes aos mapas de ativação como parte de sua saída e, consequentemente, mapas de saliências, que alimentam o *Patch-SaliMap* (Figura 5.2c). Anteriormente, o *MIL-Guided* utilizava o *Bag Model* como entrada do algoritmo para o Grad-CAM, o qual gerava os mapas de saliências (Figura 5.2b). A única diferença entre os *Bag Models* e *Instance Models* do *Attention-based MIL-Guided* e *MIL-Guided* é a adição da camada de mapas de ativação baseados em atenção. Além disso, o *Instance Model* do *Attention-based MIL-Guided* usa um modelo pré-treinado na ImageNet ao invés do *Bag Model* (Subseção 6.2.2). O método de Avaliação Ponderada é aplicado exatamente como no *MIL-Guided*.

O *MIL-Guided* é independe de arquitetura. Diferentemente, o *Attention-based MIL-Guided* requer uma arquitetura que gere mapas de saliências como saída. O *Two-WAM*, apresentado na próxima subseção, é um método de mapas de ativação que aprende a usar RI para influenciar o processo de treinamento e, ao mesmo tempo, melhorar suas localizações por meio dos pesos de atenção aprendidos.

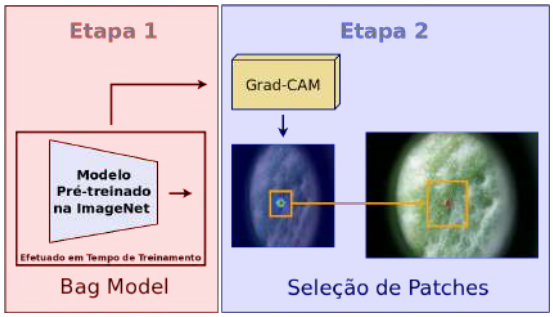
5.2.2 Mapas de Ativação com Dois Pesos

O *Two-WAM* é definido como uma transformação $T : \mathbb{R}^{w \times h \times k} \rightarrow \mathbb{R}^{w \times h}$, em que w, h e k denotam o número de linhas, colunas e mapas de características, respectivamente. Em outras palavras, o *Two-WAM* é um método que transforma um tensor com k mapas de características em apenas um, usando dois pesos otimizados para cada mapa. O processo é semelhante a uma convolução 1×1, exceto que (i) é usada uma única multiplicação escalar entre cada peso de atenção e uma matriz de características ao invés de usar uma convolução com *kernel* deslizante e, no caso do *Two-WAM*, (ii) é modelada uma combinação dos canais resultantes como uma função polinomial contendo coeficientes (pesos) lineares e exponenciais.

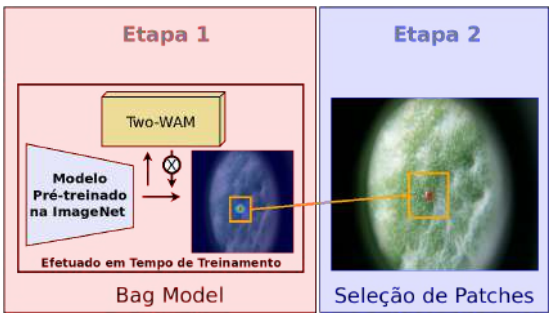
A transformação para a camada de mapas de ativação baseada em atenção (*Two-WAM*) é



(a) *Attention-based MIL-Guided*



(b) *MIL-Guided* Etapas 1 e 2



(c) *Attention-based MIL-Guided* Etapas 1 e 2

Figura 5.2: (a) O *Attention-based MIL-Guided* consiste em quatro etapas. Na Etapa 1, uma DNN (pré-treinada na ImageNet) é treinada com uma abordagem baseada em atenção, ou seja, um modelo que produz classificações e que também fornece localizações fracamente supervisionadas. Na Etapa 2, vários *patches* são automaticamente gerados usando os mapas de saliências produzidor por mapas de ativação baseados em atenção do *Bag Model*. Na Etapa 3, uma DNN (pré-treinada na ImageNet) é retreinada de acordo com uma tarefa MIL de inferir sobre as instâncias (como um classificador totalmente supervisionado). Na Etapa 4, o método de avaliação ponderada é aplicado para prever a classe da imagem original. (b) Versão *MIL-Guided* das Etapas 1 e 2. (c) Versão do *Attention-based MIL-Guided* das Etapas 1 e 2. O símbolo $\otimes$ denota a multiplicação elemento a elemento. Figura traduzida de Bollis et al. [25].

formalmente definida para considerar que os mapas de características sejam tensores contendo valores de ponto flutuante variando no espaço n-dimensional real. Por essa razão, define-se uma transformação entre k mapas de características como na Equação 5.2, a qual pode ser reescrita na forma da Equação 2.1, em que α_k e β_k são pesos que a transformação aprende em tempo de treinamento e c é um valor constante. O resultado final T_{act} é o mapa de ativação.

O T_{act} destaca os mapas de características originais f_k para $i = 1, \ldots, n$ através da operação matemática descrita na Equação 5.3 (similar à Equação 2.3), em que $\otimes$ denota uma multiplicação elemento a elemento.

$$T_{act}(x) = \frac{\sum_k \alpha_k \cdot f_k(x) \cdot c^{\beta_k}}{\sum_k \alpha_k \cdot c^{\beta_k}}. \tag{5.2}$$

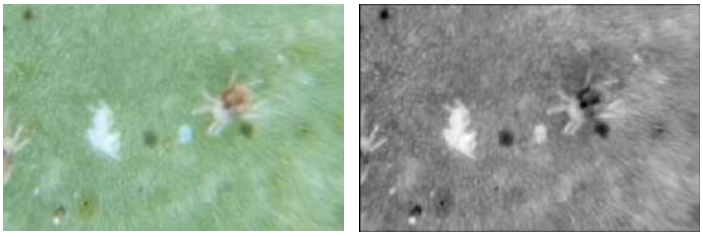
$$f_k^{\otimes}(x) = f_k(x) \otimes T_{act}(x). \tag{5.3}$$

A transformação *Two-WAM* pode ser entendida como uma fusão de inteiros para uma imagem RGB que emprega três coeficientes lineares iguais a 1, três coeficientes exponenciais iguais a 0, 1 e 2, e o valor constante igual a 256. Seu resultado gera uma nova imagem ilustrada na Figura 5.3, em que três canais são codificados em um único canal (Equação 5.4).

Na Equação 5.4, o divisor carrega o maior valor que pode ser assumido por ele para garantir

que $Im(T) \subset [0,1]^{w \times h}$.

$$T([R,G,B]) = \frac{R \cdot 256^0 + G \cdot 256^1 + B \cdot 256^2}{255 \cdot (1 + 256 + 256^2)}. \tag{5.4}$$



(a) $[R,G,B]$ (b) $T([R,G,B])$

Figura 5.3: Transformação de uma imagem RGB. (a) imagem codificada em RGB, $[R,G,B] \in \{0,...,255\}^{w \times h \times 3}$. (b) Equação 5.4 aplicada em (a) e representada em tons de cinza, $T([R,G,B]) \in [0,1]^{w \times h}$.

O *Two-WAM* visa representar dois ou mais mapas de características em apenas um, como faz a transformação de inteiros. Se for empregado $c = 10$, a transformação aprenderá as casas decimais. Por exemplo, ignorando o denominador, se forem tomados dois valores de ponto flutuante como nos mapas de características ($k = 2$) e forem usados $f_1 = 0,25$ e $f_2 = 0,01$ e $\alpha_1 = 1$, $\alpha_2 = 1$, $\beta_1 = 2$ e $\beta_2 = 0$, a Equação 5.2 produzirá o valor $T_{act} = 0,25 \cdot 10^2 + 0,01 \cdot 10^1 = 25,01$, que mostra 2 valores reais em apenas 1. É deixado ao algoritmo do gradiente descendente estocástico [87, 149] calcular a melhor maneira de transformar k valores de ponto flutuante em apenas 1 usando $c = 10$.

O *Attention-based MIL-Guided* é um dos métodos fracamente supervisionados da contribuição referente à C2: *"métodos fracamente supervisionados para classificação e localização de pragas"* e o *Two-WAM* é a proposta para a C3: *"uma nova formulação matemática que utiliza dois pesos de treinamento para criação de mapas de ativação baseados em atenção"*.

5.3 Adaptação de Domínio não Supervisionada e Remoção do Rótulo da Classe Negativa

O método proposto nesta seção modifica os modelos do *Attention-based MIL-Guided* para efetuar seu treinamento através de uma técnica de adaptação de domínio não supervisionada e do não uso do rótulo da classe negativa da base de dados CPB. Diferentemente do que foi empregado nas versões anteriores, para o treinamento multirrótulos, um rótulo de domínio e um rótulo de classe positiva no lugar do rótulo da classe negativa foram utilizados para a técnica de adaptação proposta. A Figura 5.4 ilustra a versão modificada dos modelos do *Attention-based MIL-Guided* para melhor generalização em conjuntos de imagens que não fazem parte do conjunto pré-estabelecido de treinamento.

A adaptação de domínio não supervisionada se faz necessária por causa da forte taxa de acertos do *Bag Model* no conjunto de treinamento, mas com um retorno muito baixo no conjunto de validação. Como o conjunto de treinamento é dividido em dois subconjuntos e usado como treinamento e pseudovalidação (Seção 6.1.1), enquanto os modelos são treinados, o sobreajuste

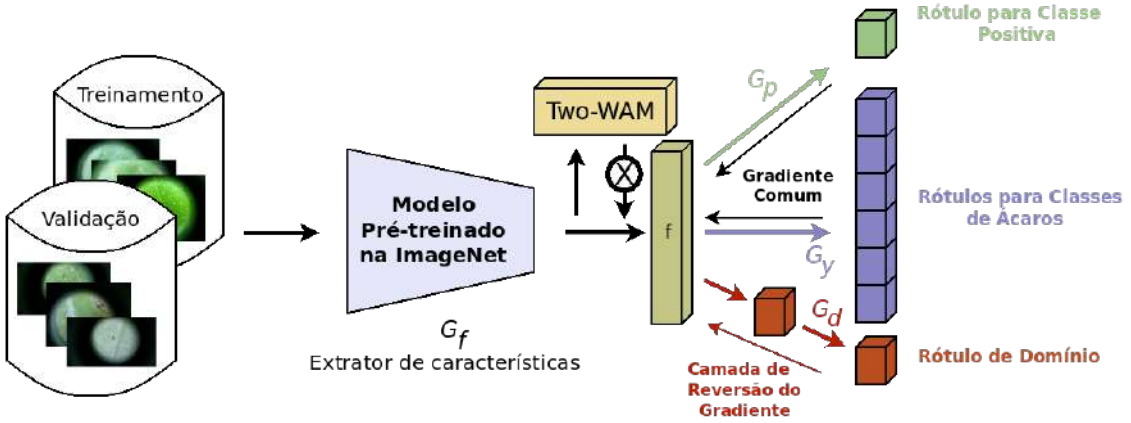


Figura 5.4: Os conjuntos de treinamento e validação são utilizados ao mesmo tempo no treinamento dos modelos de classificação multirrótulos. O *Bag Model* do *Attention-based MIL-Guided* é adaptado para conter um classificador diferenciado. Após a geração do vetor de características, um classificador o utiliza para as predições multirrótulos. Esse classificador é aplicado em paralelo com uma sequência de uma camada de reversão de gradiente procedida por um classificador que aponta quando uma imagem pertence ao conjunto de validação ou ao conjunto de treinamento. No classificador, ao invés de usar a classe negativa, utiliza-se o rótulo para a classe positiva. Quando a imagem pertence ao conjunto de validação, o erro segundo os rótulos das imagens é zerado. Figura inspirada no trabalho de Ganin e Lempitsky [61].

dos modelos é pequeno. Entretanto, em todos os modelos observados, sempre existiu uma queda considerável em sua eficácia quando avaliados no conjunto de validação "real" da base CPB. Isso evidencia uma diferença na distribuição das caraterísticas dos dois conjuntos e permite a proposição da aplicação da adaptação de domínio não supervisionada.

No caso da não utilização da classe negativa, a necessidade é mostrada experimentalmente na Seção 6.3.3. Para o problema multirrótulos com as 7 classes da base CPB, as classes existentes são representadas por um vetor onde cada um dos valores internos corresponde a existência da classe específica na imagem $[x_1, x_2, x_3, x_4, x_5, x_6, x_7], x_i \in \{0, 1\}, i = 1, ..., 7$ com x_7 o rótulo referente à classe negativa e $x_7 = 0$ se existe $x_i = 1, i = 1, .., 6$ (não existência de um dos 6 ácaros apresentados na Seção 4.1). Porém, os modelos podem ser treinados por uma representação de forma a conter somente 6 rótulos. Nesse caso, o rótulo da classe negativa é representado pela não existência de uma das classes de ácaros [0,0,0,0,0,0] e esse vetor tem um índice a menos. Para a rede, esta é uma diferença fundamental, pois o uso do rótulo da classe negativa premia a busca por características do plano de fundo. Sem o seu uso, busca-se somente as características para classificar o tipo de ácaro.

Existe a possibilidade de se trabalhar com o rótulo oposto ao da classe negativa em seu lugar, ou seja, se $x_7 = 0$, o novo rótulo terá valor igual a 1 e, se $x_7 = 1$, o novo rótulo terá valor 0. Com isso, premia-se a busca pela existência de ácaros. Na classificação binária, o oposto da classe negativa é chamado de classe positiva, então esse nome é adotado nesta tese para identificar o rótulo que corresponde à existência de um ácaro em uma imagem, mesmo que outras classes de ácaros estejam presentes, $[x_1, x_2, x_3, x_4, x_5, x_6, x_7], x_i \in \{0, 1\}, i = 1, ..., 7$ com x_7 o rótulo referente à classe positiva e $x_7 = 1$ se existe $x_i = 1, i = 1, .., 6$. Consequentemente, no treinamento de um modelo, pode-se usar oito classes $[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8], x_i \in \{0, 1\}, i = 1, ..., 8$ com x_7 o rótulo referente à classe negativa e x_8 o rótulo da classe positiva, $x_7 = 0$ se existe $x_i = 1, i = 1, .., 6$ e $x_8 = 1$ se $x_7 = 0$.

Na Seção 6.3.3, são apresentados experimentos mostrando que as características da classe negativa prejudicam a classificação multirrótulos e provocam o decaimento da eficácia dos modelos treinados na CPB.

Seja $x_i \in X \subset \mathbb{R}^{h \times w \times 3}$ um tensor representando uma imagem, em que $H, W \in \mathbb{N}^+$ são a altura e largura de x_i, propõe-se o cálculo de uma função ou modelo $G : \mathbb{R}^{H \times W \times 3} \rightarrow Y \subset \{0, 1\}^{c+1+1}$, que representa o *Bag Model* multirrótulos, tal que $c \in \mathbb{N}$ é o número de classes positivas do problema $X = T \cup V$, em que T e V são, respectivamente, os conjuntos de treinamento e validação de uma base. Se o modelo G for dividido conforme o uso de $G_f : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^n$, $G_y : \mathbb{R}^n \rightarrow \mathbb{R}^c$, $G_p : \mathbb{R}^n \rightarrow \mathbb{R}$ e $G_d : \mathbb{R}^n \rightarrow \mathbb{R}$, sendo $G = G_p \circ G_f(x_i) \cup G_y \circ G_f(x_i) \cup G_d \circ G_f(x_i)$, propõe-se o modelo gerado ao minimizar a função de perdas da Equação 5.5 nesta tese.

$$L(x_i, y_i, d_i) = \delta(d_i) \cdot \big(L_p(G_p(G_f(x_i)), \delta(y_i)) + L_y(G_y(G_f(x_i)), y_i)\big) - \lambda L_d(G_d(G_f(x_i)), d_i). \tag{5.5}$$

$$\delta(z) = \begin{cases} 1 \; se \; \exists \; z_j = 1 \\ 0 \; se \; \nexists \; z_j = 1 \end{cases}$$

É importante ressaltar que a Equação 5.5 é uma adaptação da equação do cálculo do erro para G_f, G_y, G_d na Equação 2.4 [61]. Nesta adaptação, é proposta a utilização explicita da função δ para zerar o erro quando a imagem pertence ao conjunto de validação, em que $z = [z_1, ..., z_j, ..., z_n], n \in \mathbb{N}$. O parâmetro λ é calculado iterativamente saindo de zero e aumenta até o valor 1, como descrito por Ganin e Lempitsky [61].

A adaptação de domínio não supervisionada e remoção do rótulo da classe negativa são abordagens para melhora C2: *"métodos fracamente supervisionados para classificação e localização de pragas"*.

5.4 Medidas de Avaliação

Os resultados dos métodos e modificações propostas foram avaliados na base de imagens CPB [24] e na base IP102 [197]. A acurácia normalizada e medida F1, que são medidas de avaliação da IP102 e da CPB, são utilizadas para medir a eficácia de classificação para esta tese. A intersecção sobre a união (*intersection over union*, IoU) e precisão média (*average precision*, AP) são utilizadas como medidas para a eficácia de localização fracamente supervisionada para a IP102. Para medir a eficiência das redes, o número de parâmetros é usado. É importante ressaltar que esta tese aborda um problema de múltiplas classes não balanceadas e, por isso, é necessário usar a acurácia normalizada.

5.4.1 Acurácia Normalizada

A acurácia (*Acurácia*) é uma medida que identifica a quantidade de exemplos corretamente classificados em relação ao total de exemplos. *Acurácia* é dada pelo número total de imagens classificadas corretamente em relação ao número de imagens avaliadas, Equação 5.6, em que VP são os verdadeiros positivos, VN são os verdadeiros negativos, FP são os falsos positivos e

FN são os falsos negativos.

$$\textit{Acurácia} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FN} + \text{FP}}. \tag{5.6}$$

A acurácia normalizada (*Acur*) reporta a taxa de sucesso considerando o número de exemplos para cada classe. A Equação 5.7 expressa matematicamente sua fórmula, em que TVP é a taxa de verdadeiros positivos e TVN é a taxa de verdadeiros negativos.

$$\textit{Acur} = \frac{\text{TVP} + \text{TVN}}{2}. \tag{5.7}$$

5.4.2 Medida F1

A medida F1, Equação 5.10, reporta a média harmônica entre a revocação e a precisão. A revocação, Equação 5.8, é a percentagem de exemplos positivos classificados como positivos (nenhum exemplo positivo é deixado de fora) e a precisão, Equação 5.9, é a percentagem de exemplos classificados como positivos que são realmente positivos (nenhum exemplo negativo é incluído).

$$\textit{Revocação} = \frac{\text{VP}}{\text{VP} + \text{FN}}, \tag{5.8}$$

$$\textit{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}}. \tag{5.9}$$

$$\text{F1} = 2 \times \frac{\textit{Precisão} \times \textit{Revocação}}{\textit{Precisão} + \textit{Revocação}}, \tag{5.10}$$

5.4.3 Número de Parâmetros

O número de parâmetros (*Param*) de uma arquitetura é quantificado pela soma de todos os pesos de neurônios existentes em uma de suas instâncias. Ele representa a memória disponível para o aprendizado dos conceitos relacionados a uma tarefa. Foi uma das primeiras medidas de avaliação, juntamente com o tamanho dos modelos (em *bytes*), para verificar se um modelo é adequado para uso em dispositivos móveis e embarcados. Sua aplicação pode ser vista no trabalho de Iandola et al. [82], que gerou uma das primeiras arquiteturas voltadas para esse tipo de dispositivos (SqueezeNet). Seja m a quantidade de neurônios de uma arquitetura e w_i o número de parâmetros do i-ésimo neurônio. A Equação 5.11 mostra o cálculo do número de parâmetros total de uma de suas instâncias.

$$\textit{Param} = \sum_{i=1}^{m} w_i. \tag{5.11}$$

5.4.4 Intersecção Sobre a União

A intersecção sobre a união (*intersection over union*, IoU) é uma medida que verifica o quanto duas regiões estão sobrepostas. Esta medida é usada para estimar o quão próximas e similares são as localizações inferidas e anotadas. Dada uma área demarcada A e uma segunda área demarcada B, a intersecção sobre a união retorna um resultado entre 0 e 1. A Equação 5.12

mostra essa relação de sobreposição entre duas áreas.

$$\text{IoU} = \frac{A \cap B}{A \cup B}. \tag{5.12}$$

5.4.5 Precisão Média

A precisão média (*average precision*, AP) é a medida que calcula a área abaixo da curva de precisão/revocação para localização ou classificação [57]. O cálculo da curva de precisão/revocação é realizado por meio de um método de ranqueamento. A média da precisão média (mAP) é definida como a média do retorno da AP para todas as classes. A AP é aproximada pelo método de interpolação em um conjunto de onze pontos igualmente espaçados, como mostra a Equação 5.13:

$$\text{AP} = \int_0^1 p(r)dr \simeq \frac{1}{11} \sum_{r \in \{0;\ 0{,}1;\ 0{,}2;\ldots;\ 1\}} p(r). \tag{5.13}$$

em que a precisão p, para cada nível de revocação r, é calculada por $p(r) = \max_{r' \in [a,b]} p(r')$.

Capítulo 6

Experimentos e Resultados

Neste capítulo, as seções foram divididas conforme a apresentação dos métodos do Capítulo 5. A Seção 6.1 descreve os experimentos relacionados ao *MIL-Guided*, a Seção 6.2 contém os experimentos relativos ao *Attention-based MIL-Guided* e a Seção 6.3 mostra os experimentos ligados à adaptação de domínio não supervisionada e a remoção do rótulo da classe negativa no treinamento dos modelos. A Tabela 6.1 sumariza os experimentos em relação ao seu conteúdo, base utilizada e tipo de classificação.

Tabela 6.1: Descrição dos experimentos.

Subseções	Conteúdo	Base	Tarefa
Subseção 6.1.2	Avaliação preliminar de arquiteturas	IP102 1.0	Classificação Multiclasses
Subseção 6.1.3	Comparação com resultados da literatura	IP102 1.0 e 1.1	Classificação Multiclasses
Subseção 6.1.4	Experimentos iniciais	CPB	Classificação Binária
Subseção 6.2.2	Avaliação da influência do ruído em modelos	CPB e NCPB	Classificação Binária
Subseção 6.2.3	Comparação com métodos da literatura	CPB	Classificação Binária
Subseção 6.2.4	Comparação com métodos da literatura	IP102 1.1	Classificação Multiclasses
Subseção 6.2.5	Comparação com métodos da literatura	IP102 1.0	Localização Multiclasses
Subseção 6.2.6	Experimento qualitativo	CPB	Localização Binária
Subseção 6.2.7	Avaliação da unificação de etapas	CPB	Classificação Binária
Subseção 6.3.2	Avaliação das propostas no novo contexto	CPB	Classificação Multirrótulos
Subseção 6.3.3	Avaliação do uso do rótulo da classe negativa	CPB	Classificação Multirrótulos
Subseção 6.3.4	Avaliação da adaptação de domínio	CPB	Classificação Multirrótulos

Os experimentos envolveram duas bases de dados. Na base IP102 [197] (versões 1.0 e 1.1), diversas arquiteturas foram avaliadas em tarefas de classificação (Subseções 6.1.2, 6.1.3 e 6.2.4) e localização fracamente supervisionada de imagens de pragas (Subseção 6.2.5). Na base de dados *Citrus Pest Benchmark* (CPB) [24], o problema de classificação de ácaros foi primeiramente avaliado para classificação binária (Subseções 6.1.4, 6.2.2, 6.2.3 e 6.2.7), que prevê a existência ou não de ácaros (classe positiva e negativa) e, posteriormente, avaliado para classificação multirrótulos (Subseções 6.3.2, 6.3.3 e 6.3.4): ácaros purpúreos, ácaros fitoseídeos, ácaros da ferrugem, ácaros da leprose, ácaros brancos, ácaros rajados e classe negativa. A localização fracamente supervisionada de ácaros da CPB foi avaliada qualitativamente neste capítulo (Subseção 6.2.6), pois a CPB não contém rótulos de localização.

As seções estão estruturadas em subseções contendo uma subseção de configuração dos experimentos no início, subseções com experimentos realizados e, ao final, uma subseção de discussões para cada seção.

6.1 Experimentos para o *MIL-Guided*

Esta seção descreve os experimentos relativos à criação do método fracamente supervisionado de aprendizado por múltiplas instâncias baseado em mapas de saliências ou *MIL-Guided*. A Subseção 6.1.1 apresenta as configurações utilizadas nos experimentos desta seção. Em seguida, são mostradas as comparações entre diferentes arquiteturas de aprendizado profundo (DNNs) na base de dados IP102 1.0, o que estabeleceu a melhor entre elas para utilização nos demais experimentos (Subseção 6.1.2). Então, os resultados dos modelos da melhor arquitetura foram comparados com os demais resultados disponíveis na literatura (Subseção 6.1.3). Também foram avaliados experimentos preliminares relativos ao *MIL-Guided* e à base CPB (Subseção 6.1.4). Por fim, as discussões relativas a esta seção são apresentadas (Subseção 6.1.5).

Os experimentos presentes nesta seção foram publicados e apresentados no *Agriculture-Vision Workshop* da *Conference on Computer Vision and Pattern Recognition* (CVPR 2020) [24].

6.1.1 Configuração dos Experimentos

Os experimentos foram avaliados em seis DNNs, a saber: Inception-v4 [172], ResNet-50 [71] e ResNet-18 [71], NASNet-A [234], MobileNet-v2 [156] e EfficientNet-B0 [173]. Essas redes foram escolhidas porque cobrem diferentes estratégias convolucionais de extração de características e número de pesos apresentados nas DNNs atuais.

Cada DNN foi treinada com a descida do gradiente estocástico (SGD), o otimizador AdaDelta [215], com valor máximo de *batch* igual a 128, taxa de aprendizado igual a 0,1, valor do decaimento igual a 0,0005 e uso da função de custo *cross-validation entropy* sobre a saída da função softmax. Para todas as DNNs, foram usados modelos pré-treinados na ImageNet [91] e o treinamento foi ajustado (aplicado o *fine tuning*) em todas as camadas das redes. As imagens foram normalizadas, subtraindo-se a média e dividindo-se pelo desvio padrão, com base na ImageNet. Para a IP102, todas as imagens foram redimensionadas para 224×224 pixels.

A aumentação automática dos dados (*data augmentation*) foi aplicada em tempo de treinamento, onde os experimentos usaram escala de 0,6 a $1{,}4\times$, rotação de 0 a 360 graus com valores múltiplos de 15, reflexão vertical e horizontal e translação de 0 a 4 pixels na vertical e na horizontal.

Para reduzir os efeitos de *overfitting* na base de dados IP102, usou-se *dropout* [168] entre cada módulo da arquitetura EfficientNet-B0 (20%) e após cada convolução do tipo *depth-wise* (30%). Para a base CPB, também usou-se *dropout* entre cada um dos módulos da EfficientNet-B0 (20%), após cada convolução *depth-wise* (40%) e antes da camada final (30%). Também foram utilizadas técnicas de regularização ℓ_1 e ℓ_2 com penalidade de 30%.

Para gerar os mapas de saliências, usou-se o método de Mapeamento da Ativação de Classes por Pesos do Gradiente (Grad-CAM) [161].

Os modelos foram treinados em unidades gráficas de processamento (*graphics processing units*, GPUs) dos tipos NVIDIA RTX 5000 e RTX 2080 Ti. Os experimentos foram realizados usando Keras, executado juntamente com TensorFlow. O código auxiliar foi desenvolvido usando as bibliotecas NumPy, Pandas e Scikit-Learn. Para o Grad-CAM[1] e todas as DNNs

[1] https://github.com/jacobgil/keras-grad-cam

(exceto a EfficientNet[2]), os experimentos foram realizados usando a implementação do Keras.

Para cada experimento, cinco conjuntos de treinamentos diferentes foram gerados para reduzir os efeitos de aleatoriedade. Uma semente diferente em cada experimento foi usada para dividir o conjunto de treinamento em duas partes, a primeira, usada efetivamente no treinamento e, a segunda, chamada de pseudovalidação, usada para escolher o melhor modelo enquanto o treinamento é efetuado. O conjunto de validação foi usado como pseudoteste para escolher os melhores modelos e não sobreajustar os hiperparâmetros. Os resultados nos conjuntos de testes foram avaliados somente antes da escrita dos artigos publicados ou desta tese. O uso do balanceamento do *batch* por classe foi aplicado em todos os experimentos.

6.1.2 Arquiteturas de Aprendizado Profundo Avaliadas na IP102 1.0

A base de dados IP102 1.0 [197] foi utilizada para avaliar diferentes DNNs na tarefa de classificação de pragas e insetos. O desempenho da comparação entre as arquiteturas foi analisado usando a acurácia. Os resultados para IP102 1.0 são reportados na Tabela 6.2.

Tabela 6.2: Resultado de diferentes arquiteturas DNNs em relação à acurácia e ao desvio padrão da classificação (em %) no conjunto de *validação* da IP102 1.0. Os destaques correspondem aos melhores resultados da tabela. "Param. (M)" significa o número de parâmetros em milhões.

DNNs	Acur. (%)	Param. (M)
Inception-v4	48,2 $\pm$2,6	41,2
ResNet-50	54,2 $\pm$0,4	23,6
ResNet-18	50,4 $\pm$0,5	11,2
NASNet-A	53,4 $\pm$2,9	4,4
EfficientNet-B0	**59,8 $\pm$0,4**	4,1
MobileNet-v2	53,0 $\pm$0,7	**2,3**

Não surpreendentemente, a arquitetura EfficientNet (o estado da arte entre as DNN convolucionais da época) atingiu a melhor eficácia na classificação (59,8% de acurácia). Entretanto, foi usada a menor versão da EfficientNet, chamada B0. Isso pode indicar que o número relatado não representa o limite da eficácia da classificação alcançável pelas EfficientNets.

Em relação ao número de pesos (considerando um cenário voltado aos dispositivos móveis), a arquitetura MobileNet-v2, a menor DNN entre os experimentos, obteve 53,0% de acurácia, uma diferença de 6,8 pontos percentuais quando se compara sua eficácia à da EfficientNet-B0.

6.1.3 Comparação com a Literatura para a IP102

A Tabela 6.3 mostra a comparação dos melhores resultados relatados no conjunto de testes da base IP102. É importante ressaltar que, até meados de 2020, os resultados da IP102 estavam em torno de 60% de acurácia, porém, a partir desse ano, os resultados ficaram mais próximos de 70%. Isso provavelmente ocorreu porque a versão IP102 1.1 foi lançada. Então, pode-se conjecturar que os resultados até meados de 2020 foram avaliados para a IP102 1.0 e depois para a IP102 1.1, entretanto, os trabalhos da literatura não mencionam informações a respeito

[2] https://github.com/qubvel/efficientnet/blob/master/efficientnet

do assunto. Os resultados do *MIL-Guided* (*Bag Model*) corroboram com essa hipótese, pois nenhuma alteração ocorreu na arquitetura ou na forma de treinamento dos modelos, mas as avaliações na IP102 1.0 e IP102 1.1 diferem em 9 pontos percentuais.

O resultado Inicial (ResNet-50) [197] é o melhor resultado reportado pelos autores da IP102. Eles também relataram estatísticas que demonstram que a base de dados é fortemente desbalanceada em comparação com outras bases.

Tabela 6.3: Desempenho da classificação de diferentes DNNs no conjunto de *teste* da base de imagens IP102. "Param. (M)" significa o número de parâmetros em milhões para cada DNN. "–" significa que o valor não foi encontrado no trabalho original e os destaques correspondem aos melhores resultados. Os resultados marcados com * foram calculados a partir dos valores fornecidos nos trabalhos.

DNNs	Acur. (%)	F1 (%)	Param. (M)	Ano
Inicial (ResNet-50) [197]	49,0	40,1	23,6	2019
FR-ResNet [147]	55,2	54,8	30,8	2019
XCloud (DenseNet-121) [207]	61,1	–	7,1	2019
DMF-ResNet [115]	59,2	58,4	29,7	2020
SaliencyEnsemble [133]	61,9	59,2	71,0*	2020
MIL-Guided (*Bag Model* IP102 1.0) [24]	60,7	59,6	**4,1**	2020
SMPEnsemble [9]	66,2	64,4	49,9*	2020
GAEnsemble [9]	67,1	65,8	49,9*	2020
CRN [210]	70,4	–	**4,1**	2021
MMAL-Net [178, 217]	72,2	64,6	–	2021
Ensemble MMAL-Net [178]	74,1	67,7	–	2021
OptimizerEnsemble [134]	**74,1**	**73,0**	592,8*	2022
MIL-Guided (*Bag Model* IP102 1.1) [25]	69,5	69,0	**4,1**	2022

A abordagem FR-ResNet [147] alterou blocos residuais adicionando convoluções e reutilizando características dentro de um mesmo bloco. Eles levantaram a hipótese de que a reutilização de características em convoluções intermediárias de um bloco melhora o desempenho final de uma rede. Para serem justos e não aumentarem muito o número de parâmetros de cada bloco testado, compararam diferentes tipos de convoluções internas aos blocos utilizando o mesmo número de parâmetros em todos os experimentos.

A abordagem DMF-ResNet [115] usa como base o trabalho da FR-ResNet [147] e propõe um bloco baseado no mecanismo de atenção, que foi incorporado ao bloco residual anteriormente desenvolvido para fundir várias ramificações enquanto recalibra a reutilização de características.

A abordagem XCloud (DenseNet-121) [207] não trouxe nenhuma informação sobre como os autores atingiram a acurácia relatada, nem quantas vezes treinaram a rede e se o valor relatado seguiu o protocolo da base. Isso porque não era o objetivo final do trabalho, que propôs mostrar a eficácia de uma infraestrutura de treinamento de redes na nuvem. Além disso, outras métricas não foram relatadas em seu estudo, por exemplo, a medida F1.

A abordagem SaliencyEnsemble [133] usou vários métodos de detecção de saliências para gerar nove bases de dados contendo mapas de saliências derivados das imagens da IP102, em

que cada detector gerou uma base de dados distinta. Os autores treinaram uma DenseNet-121 para cada uma dessas bases e uma para a base original, fazendo um *ensemble* de dez redes.

As abordagens SMPEnsemble e GAEnsemble foram propostas no mesmo trabalho [9]. A SMPEnsemble é um *ensemble* que une os três melhores modelos treinados de um grupo de arquiteturas através da soma das maiores probabilidades produzidas por esses modelos: Inception-v3 [171], Xception [41] e MobileNet-v2 [156]. A GAEnsemble usa algoritmos genéticos para escolher os pesos de uma votação ponderada para os mesmos modelos usados na SMPEnsemble.

A abordagem CRN [210] utilizou a EfficientNet-B0 como extratora de características e incluiu um módulo de rebalanceamento convolucional, um módulo de aumentação de imagens e um módulo de fusão de características. Os autores não reportaram resultados para a medida F1.

A abordagem MMAL-Net [178, 217] é uma arquitetura previamente existente que utiliza atenção para conseguir características de granularidade mais fina. A abordagem Ensemble MMAL-Net [178] utiliza a arquitetura MMAL-Net conjuntamente com uma FPN [109] e uma RAN [185] em um *ensemble* de redes.

A abordagem OptimizerEnsemble [134] une diversas redes neurais em outro *ensemble*, cujos modelos foram treinados com diferentes otimizadores, incluindo novos otimizadores propostos pelos autores. Foram utilizadas uma DenseNet-201 [80], 15 ResNet-50 [71], 15 GoogleNet [171], 15 ShuffleNet [223], 11 MobileNet-v2 [156] e 15 EfficcientNet-B0 [173] no *ensemble*.

É importante notar que os melhores resultados produzidos atualmente para a IP102 são valores relacionados a *ensembles* de modelos. Isso aumenta linearmente a quantidade de parâmetros dependendo da quantidade de modelos utilizados. Além disso, o retorno do aprendizado não é proporcional ao tempo de treinamento dos modelos que estão sendo treinados. Os modelos do *MIL-Guided* para o *Bag Model* treinados por esta tese conseguiram se manter competitivos com outras abordagens, mesmo com muito menos parâmetros.

Esta subseção foi concebida para ajudar a efetivar o objetivo O4: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"*. A arquitetura EfficientNet-B0 mostrou um melhor resultado para lidar com bases desbalanceadas e, como tal, torna-se parte da contribuição C5: *"resultados competitivos em relação aos métodos disponíveis na literatura para duas bases de dados distintas'*.

6.1.4 Avaliações Preliminares para a Classificação Binária da CPB

Para esta seção e as seguintes, o método *MIL-Guided* foi avaliado utilizando a EfficientNet-B0, devido a sua eficácia na IP102. Os resultados desta seção foram divididos em três partes:

- **Tradicional**: Como a EfficientNet-B0 requer imagens de entrada com tamanho 224×224 pixels, todas as imagens da base CPB foram redimensionadas, distorcendo-se a resolução para testar a configuração padrão da arquitetura. Para comparar a diferença entre as resoluções e assim a proporção do tamanho dos ácaros nas convoluções, também usou-se a rede com tamanho de imagens igual a 1200×1200 pixels.

- ***Baseline***: Inicialmente, todas as imagens foram redimensionadas do tamanho original para 897×897 pixels. Em seguida, foram geradas amostras do tamanho 299×299 pixels, pelo uso de um *grid* contendo nove *patches* de mesmo tamanho e foram selecionadas

visualmente imagens com ácaros (providas por imagens da classe positiva) e sem ácaros (de imagens da classe negativa) como amostras de *patches* das classes positivas e negativas das imagens. Este é um processo demorado e que, possivelmente, gerou erros de anotação por causa das distorções devido ao foco da lupa (alguns recortes estavam muito borrados, o que confundia pontos brancos ou coloridos com a existência ou não de ácaros) e à fadiga (foram verificados aproximadamente 66.000 *patches*).

- ***MIL-Guided***: Nesta tese de doutorado, deseja-se mostrar que o tamanho das RIs nas imagens é mais importante do que o tamanho da própria imagem. Portanto, para realizar uma comparação justa com o *Baseline*, foram escolhidos *patches* de tamanho 400×400 para imagens com tamanho 1200×1200, mantendo a mesma proporção do número de *patches* por imagem $\left(\frac{1200\times1200}{400\times400} = \frac{897\times897}{299\times299} = 9\right)$.

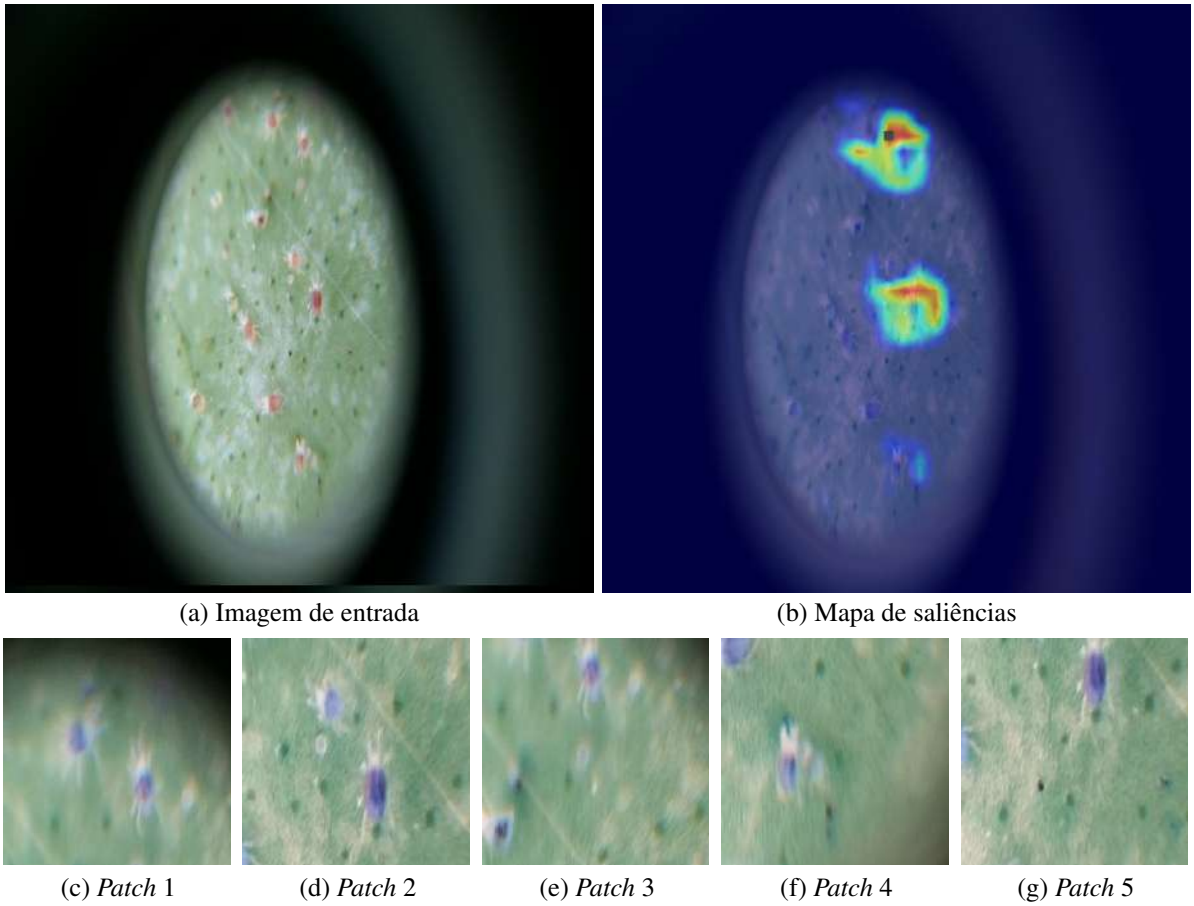
Tabela 6.4: Resultados da acurácia e medida F1 (em %) dos experimentos no conjunto de validação da CPB. Os resultados foram divididos em três partes: Tradicional, *Baseline* e *MIL-Guided*. Os valores destacados correspondem aos melhores resultados.

EfficientNet-B0	Acur. (%)	F1 (%)
Tradicional		
Sem *patches*, 224 × 224 pixels	75,9 $\pm_{0,8}$	71,4 $\pm_{1,5}$
Sem *patches*, 1200 × 1200 pixels	81,2 $\pm_{1,1}$	80,9 $\pm_{3,2}$
Baseline		
Manualmente selecionada, 299 × 299 pixels	86,0 $\pm_{0,2}$	85,4 $\pm_{0,5}$
MIL-Guided		
Gerados automaticamente, 400 × 400 pixels	**91,8** $\pm_{1,1}$	**91,2** $\pm_{2,0}$

A Tabela 6.4 pode ser resumida da seguinte forma: o *MIL-Guided* supera o desempenho da classificação em todos os cenários. A comparação entre os resultados do cenário Tradicional mostra que, como geralmente observado para a classificação [128, 179], imagens de alta resolução levam a um melhor desempenho. O cenário do *Baseline* (*patches* anotados manualmente) mostra resultados promissores, no entanto, a anotação de *patches* é uma tarefa tediosa, demorada e propensa a erros. Comparando os resultados do cenário Tradicional com o resultado do *Baseline* (sem *patches*, 1200×1200 pixels), a acurácia e medida F1 aumentam significativamente na classificação, indicando que o modelo pode se beneficiar das representações de *patches*. Comparando o *MIL-Guided* com o *Baseline* (*patches* gerados automaticamente *versus patches* anotados manualmente), foi obtido um aumento de 86,0% para 91,8%, quando se compara as proporções parecidas (299×299 contra 400×400). O melhor modelo no conjunto de teste alcançou uma acurácia de 92,1%.

Para ilustrar os *patches* gerados automaticamente pelo *Patch-SaliMap*, a Figura 6.1 foi criada. Os *patches* são classificados e elencados de acordo com sua ativação, começando pela maior (Figura 6.1c) e seguindo de forma decrescente até a menor ativação (Figura 6.1g). Os *patches* gerados destacam o impacto positivo do método desenvolvido.

Os experimentos sobre o *MIL-Guided* desta subseção ajudaram a responder parcialmente a questão de pesquisa Q1: *"métodos de aprendizado fracamente supervisionados por múltiplas*



(a) Imagem de entrada (b) Mapa de saliências

(c) *Patch* 1 (d) *Patch* 2 (e) *Patch* 3 (f) *Patch* 4 (g) *Patch* 5

Figura 6.1: *Patches* automaticamente gerados a partir do mapa de saliências. Figura reproduzida de Bollis et al. [24].

instâncias são eficazes para a classificação de pragas da citricultura em regiões pequenas?" e aproximaram esta tese de cumprir os objetivos O4: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"* e O2: *"verificar a importância do tamanho das regiões de interesses nas imagens"*.

Nesta subseção, o objetivo O3 *"avaliar a integração dos métodos de múltiplas instâncias e mapas de ativação"* foi cumprido para a tarefa binária, pois mostrou-se que, considerando a proposta do *MIL-Guided*, o uso dos mapas de ativação conjuntamente com MIL traz bons resultados para a classificação de pequenas regiões e, com isso, que é possível seu uso integrado.

6.1.5 Discussão dos Experimentos

Nesta seção, foram exibidas avaliações preliminares para mostrar a efetividade do processo chamado *MIL-Guided* na base de dados CPB, que foi a primeira base de dados contendo imagens adquiridas via dispositivos móveis de pragas cítricas invisíveis a olho nu. O *MIL-Guided* foi desenvolvido para classificar pequenas RI utilizando: uma arquitetura pré-existente (EfficientNet-B0 [173]), uma estratégia de seleção de múltiplos *patches* (*Patch-SaliMap*) e um Método de Avaliação Ponderada. Dentre as contribuições apresentadas, foram mostrados os primeiros resultados para a base de dados CPB.

A partir dos experimentos desta seção, a arquitetura EfficientNet-B0, utilizada pelo *MIL-Guided*, mostrou ser capaz de alcançar efetividade superior quando comparada com outras ar-

quiteturas de propósito geral no conjunto de dados da IP102 1.0 e, por isso, foi escolhida como base para o *MIL-Guided*. Além disso, o *Patch-SaliMap* mostrou ser eficaz no conjunto de dados da CPB, superando dois cenários experimentais diferentes. As avaliações das arquiteturas de propósito geral mostram que estas não são adequadas para lidar com RIs muito pequenas. Este fato também é apresentado na literatura [190]. Para melhorar a efetividade das arquiteturas gerais no contexto desta tese, foram necessárias a investigação e inserção de novas técnicas que lidaram com imagens contendo RIs pequenas e minúsculas.

As abordagens de aprendizado fracamente supervisionado testadas nesta seção, quando em conjunto (MIL e mapas de ativação), produziram bons resultados na identificação e classificação de RIs. A estratégia de seleção de múltiplos *patches*, o *Patch-SaliMap*, reduziu a probabilidade de perder regiões relevantes e, consequentemente, produziu resultados de classificação melhores do que as arquiteturas gerais experimentadas. Entretanto, o uso de gradientes da técnica Grad-CAM, que mostra as regiões mais discriminativas para a classificação e, com isso, a possível localização, pode não ser uma opção adequada para aplicação em dispositivos móveis e suas inferências de localização podem não ser acuradas o suficiente para um detector de objetos. Experimentos foram realizados para verificar estas hipóteses (Subseções 6.2.2 e 6.2.3).

6.2 Experimentos para o *Attention-based MIL-Guided*

Esta seção exibe experimentos relativos ao método fracamente supervisionado de aprendizado por múltiplas instâncias baseado em atenção e guiado por mapas de saliências ou *Attention-based MIL-Guided*, bem como o próprio *MIL-Guided*. A Subseção 6.2.1 apresenta as configurações que não foram anteriormente apresentadas. Em seguida, vários experimentos para analisar o impacto de imagens ruidosas no treinamento dos processos propostos nesta tese são apresentados na Subseção 6.2.2. Em sequência, são mostrados os resultados do *Attention-based MIL-Guided*, *MIL-Guided* e métodos fracamente supervisionados da literatura, como o *Attention-based Deep MIL* [83] e o WILDCAT [56]. Essas comparações são apresentadas para a base de dados CPB (Subseção 6.2.3) e a base IP102 1.1 (Subseção 6.2.4). Antes das discussões (Seção 6.2.8), são apresentados os resultados de localização para a base de dados IP102 1.0 (Subseção 6.2.5), os resultados qualitativos para a localização fracamente supervisionada de ácaros da CPB (Subseção 6.2.6) e os motivos pelos quais não é aconselhável o uso do *Attention-based MIL-Guided* como uma arquitetura ponta a ponta (Seção 6.2.7).

Os experimentos relatados nas Subseções 6.2.2, 6.2.3 e 6.2.4 foram publicados no periódico *Computers and Electronics in Agriculture* (COMPAG 2022) [25].

6.2.1 Configuração dos Experimentos

Para esta seção, a EfficientNet-B0 [173], pré-treinada na ImageNet [91], foi utilizada como principal arquitetura extratora de características (conforme experimentos da Seção 6.1). Para as abordagens *Attention-based Deep MIL* e WILDCAT, também foram considerados os extratores de características das propostas originais, LeNet [94] e ResNet-101 [71], respectivamente.

A configuração dos experimentos referentes ao *Attention-based MIL-Guided* é a mesma que a dos experimentos para o *MIL-Guided* (Subseção 6.1.1). Foi utilizado o valor $c = 10$ (Se-

ção 5.2.2) para o *Two-WAM*. Em relação ao *Attention-based Deep MIL*[3], foram usadas taxas de aprendizado variando entre 10^{-6} e 10^{-8} conjuntamente com o mecanismo *gated attention* [83]. Para o caso do WILDCAT[4], foi considerada uma taxa de aprendizado de 2×10^{-3}, um valor referente às regiões de 0,4 e um número de 8 mapas para cada classe utilizada nos treinamentos. Uma GPU Nvidia Quadro RTX 8000 foi utilizada em todos os experimentos.

6.2.2 Avaliação da Remoção de Ruídos para Classificação Binária da CPB

Nesta subseção, foi realizado um estudo de comparação para compreender o impacto de diferentes estratégias relativas ao *MIL-Guided* e *Attention-based MIL-Guided*, mais precisamente, um estudo do impacto de imagens ruidosas no treinamento dessas estratégias. Para isso, foi necessária a criação de um subconjunto de imagens da CPB que não contivesse imagens com problemas de luminosidade e borramento, chamado *Noiseless Citrus Pest Benchmark* (NCPB), descrito na Seção 4.1.3. Foram utilizados apenas os conjuntos de treinamento e validação da CPB e NCPB para selecionar a melhor estratégia e não sobreajustar os hiperparâmetros nos conjuntos de testes.

Avaliação do *Bag Model*

A Tabela 6.5 exibe os resultados para o *Bag Model*. Nela, são descritos a influência (i) da remoção de imagens ruidosas nos treinamentos ("NCPB") e (ii) o impacto do uso do *dropout* na camada totalmente conectada dos modelos ("Dropout"). Foram comparados os resultados dos *Bag Models* para o *Attention-based MIL-Guided* (modelo que utiliza o *Two-WAM*) e *MIL-Guided* nos conjuntos de validação da CPB e NCPB.

Tabela 6.5: Acurácia (em %) e medida F1 (em %) para diferentes estratégias avaliadas para *Bag Models* nos conjuntos de *validação* da CPB e NCPB. "*Attention-based MIL-Guided*" refere-se aos modelos treinados com o *Two-WAM*; e "Dropout" modelos treinados com *dropout* nas camadas totalmente conectadas. Os destaques correspondem aos melhores resultados.

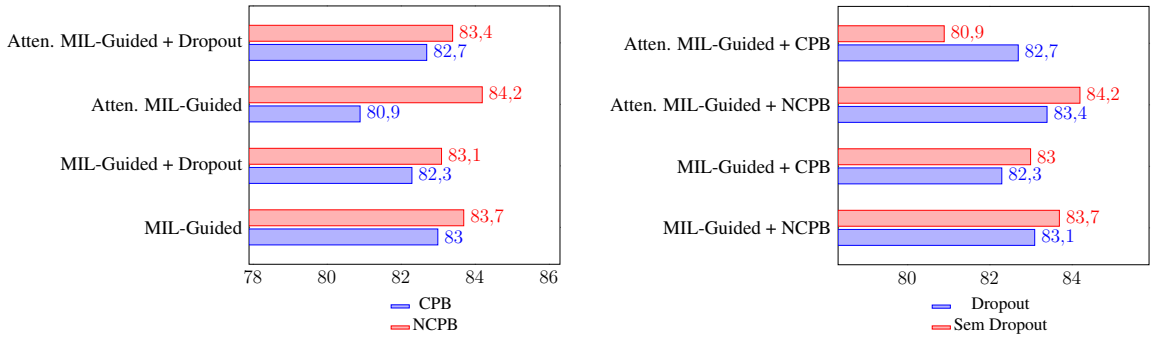
Método	NCPB	Dropout	CPB Validação		NCPB Validação	
			Acur. (%)	F1 (%)	Acur. (%)	F1 (%)
MIL-Guided			80,9 $\pm$1,9	78,4 $\pm$2,1	83,0 $\pm$0,9	80,8 $\pm$0,8
MIL-Guided		•	81,2 $\pm$1,1	78,4 $\pm$1,3	82,3 $\pm$0,9	**82,0 $\pm$0,8**
MIL-Guided	•		81,4 $\pm$0,9	79,2 $\pm$1,1	83,7 $\pm$0,9	81,8 $\pm$0,8
MIL-Guided	•	•	80,7 $\pm$1,6	78,6 $\pm$1,7	83,1 $\pm$1,8	81,0 $\pm$1,9
Attention-based MIL-Guided			80,7 $\pm$1,3	76,8 $\pm$2,3	80,9 $\pm$1,1	77,2 $\pm$1,3
Attention-based MIL-Guided		•	81,7 $\pm$1,3	77,8 $\pm$2,0	82,7 $\pm$1,1	79,2 $\pm$1,9
Attention-based MIL-Guided	•		**82,3 $\pm$1,5**	**79,4 $\pm$1,8**	**84,2 $\pm$1,4**	81,8 $\pm$1,3
Attention-based MIL-Guided	•	•	81,5 $\pm$1,2	78,6 $\pm$1,1	83,4 $\pm$1,4	80,4 $\pm$1,5

Os resultados mostram que a abordagem voltada ao *Two-WAM* ("*Attention-based MIL-Guided*") melhora a acurácia de classificação na maior parte dos casos considerando configurações semelhantes aos *Bag Models*, com exceção do melhor resultado do *MIL-Guided*, ou seja, sem a remoção das imagens ruidosas ("NCPB") e sem o uso da camada de regularização

[3] https://github.com/AMLab-Amsterdam/AttentionDeepMIL
[4] https://github.com/durandtibo/wildcat.pytorch

imposta pelo *dropout* (“Dropout”). Isso ilustra a relevância do *Two-WAM* como abordagem baseada em atenção. Além disso, os resultados avaliados nos conjuntos de validação da CPB e NCPB parecem ter comportamentos semelhantes para o mesmo modelo, aumentando e diminuindo as acurácias e medidas F1, dependendo do modelo analisado.

A Figura 6.2a resume visualmente os resultados da coluna “NCPB Validação” para acurácia que foi apresentada na Tabela 6.5. Ela mostra um melhor desempenho usando modelos treinados na NCPB do que treinados na CPB (todas as imagens). O melhor resultado foi obtido treinando o *Attention-based MIL-Guided* na NCPB, com 82,3% de acurácia e 79,4% de medida F1 no conjunto de validação da CPB, e 84,2% de acurácia e 81,8% de medida F1 na NCPB. Em contraste, a Figura 6.2b evidencia que o *dropout* influencia negativamente no treinamento do *Bag Model* em três dos quatro experimentos. De fato, o *dropout*, sem outro cenário presente, melhora o desempenho da classificação, mas a combinação da remoção de imagens ruidosas e *dropout* (“NCPB” e “Dropout”) não supera outras estratégias individualmente.



(a) Imagens ruidosas no treinamento do *Bag Model*

(b) *Dropout* no treinamento do *Bag Model*

Figura 6.2: Os efeitos do uso: (a) remoção de imagens ruidosas e (b) *dropout* do *Bag Model*. Significados: “*Atten. MIL-Guided*”, *Bag Models* do método *Attention-based MIL-Guided* (usando *Two-WAM*); “*MIL-Guided*”, *Bag Models* do método *MIL-Guided* (sem *Two-WAM*); “Dropout”, modelos treinados com *dropout*; “NCPB”, modelos treinados apenas com imagens da NCPB (sem ruídos); e “CPB”, experimentos treinados com a base original. Figura modificada de Bollis et al. [25].

Em relação à avaliação do *dropout* da Tabela 6.5, diferentes modelos do *Attention-based MIL-Guided* obtiveram os melhores resultados. Um modelo obteve 81,7% de acurácia na CPB (sexta linha da Tabela 6.5) e outro modelo obteve 83,4% na NCPB (oitava linha da Tabela 6.5). Considerando a medida F1, dois modelos atingiram 78,6% na CPB (quarta e oitava linhas da Tabela 6.5) e um modelo alcançou 82,0% na NCPB (segunda linha da Tabela 6.5).

Os experimentos do primeiro cenário proposto (“NCPB” na Tabela 6.5) explicam como a remoção de imagens ruidosas afetam o processo de treinamento do *Bag Model* (imagens inteiras). Considerando os pares de experimentos com mesma configuração para o *Attention-based MIL-Guided* no conjunto de validação da NCPB, o cenário de treinamento sem ruído melhora a acurácia em até 3 pontos percentuais. A remoção de imagens ruidosas afeta todos os modelos, impactando positivamente no processo de treinamento independentemente da regularização, neste caso, do *dropout*. No segundo cenário, a regularização (“Dropout” na Tabela 6.5) reduz a acurácia e a medida F1 no *Bag Model* do *MIL-Guided* não dependendo da presença de ruído nas imagens. O *Attention-based MIL-Guided* melhora os valores dos resultados avaliados ao lidar com pequenas RI na presença de ruídos. Apesar do resultado com o uso do *Two-WAM*

treinado nas imagens da CPB ser favorável para o uso de *dropout*, o melhor resultado alcançado foi o cenário sem ele e com treinamento na NCPB. Assim, não se pode estabelecer uma regra em relação à regularização trazida pelo *dropout*.

Portanto, os *Bag Models* do *Attention-based MIL-Guided* treinados na NCPB e sem *dropout* são a configuração arquitetural mais adequada para gerar os melhores resultados e, possivelmente, mapas de saliências. Porém, a diferença entre essa melhor abordagem e seu par para o *MIL-Guided* é menor que 1 ponto percentual, deixando as duas opções praticamente empatadas. A Tabela 6.5 mostra que a melhor configuração utilizando o *Two-WAM* alcançou 82,3% de acurácia e 79,4% de medida F1. Em relação ao conjunto de validação da NCPB, o mesmo modelo chegou a 84,2% de acurácia e 81,8% de medida F1.

Avaliação do *Instance Model*

A Tabela 6.6 mostra os resultados para os *Instance Models* e foi idealizada para analisar o impacto do *fine tuning* do *Bag Model* ("FT") e da remoção de imagens ruidosas ("NCPB") no treinamento do *Instance Model* com instâncias de tamanho 400×400 pixels. Essas instâncias, *patches* da CPB e NCPB, contêm RIs relativamente maiores em comparação com as imagens originais, ou seja, ácaros mais salientes. A Tabela 6.6 apresenta os resultados obtidos nos conjuntos de validação da CPB e NCPB. Esses resultados são divididos em dois grupos de acordo com o método de mapa de ativação que oferece as localizações para o *Patch-SaliMap* (Subseção 5.1.5), ou seja, em instâncias geradas pelo Grad-CAM ou *Two-WAM*. Nos experimentos referentes ao *Bag Model*, o *Attention-based MIL-Guided*, que utiliza o *Two-WAM*, foi comparado com o *MIL-Guided*. Para este experimento, algumas opções de *fine tuning* não são possíveis devido às diferenças arquiteturais como, por exemplo, o *fine tuning* do *Bag Model* do *Attention-based MIL-Guided* como um *Instance Model* do *MIL-Guided* ("*Two-WAM*" + "FT").

Tabela 6.6: Acurácia de classificação (em %) e medida F1 (em %) para diferentes estratégias no conjunto de instâncias geradas a partir dos conjuntos de *validação* da CPB e NCPB. "FT" refere-se aos experimentos de adaptação (*fine tuning*) do *Bag Model* no treinamento do *Instance Model*. Os destaques correspondem aos melhores resultados.

	Método	NCPB	FT	CPB Validação Acur. (%)	CPB Validação F1 (%)	NCPB Validação Acur. (%)	NCPB Validação F1 (%)
Grad-CAM	*MIL-Guided*		•	91,8 ±2,4	91,0 ±2,2	–	–
	MIL-Guided			88,0 ±0,8	86,8 ±1,3	87,9 ±0,6	86,6 ±0,5
	MIL-Guided	•		85,6 ±1,0	84,4 ±1,1	85,6 ±1,2	84,4 ±1,1
	Attention-based MIL-Guided			89,0 ±1,1	87,8 ±1,3	88,8 ±1,3	87,8 ±1,3
	Attention-based MIL-Guided		•	89,3 ±0,6	88,0 ±0,7	89,6 ±0,7	88,6 ±0,9
	Attention-based MIL-Guided	•		86,6 ±2,0	85,2 ±2,2	85,5 ±2,6	84,4 ±2,9
	Attention-based MIL-Guided	•	•	86,7 ±3,4	84,2 ±5,0	87,5 ±3,2	85,8 ±4,1
Two-WAM	*MIL-Guided*			93,3 ±0,8	92,2 ±0,8	93,5 ±0,7	92,6 ±0,5
	MIL-Guided	•		90,4 ±0,9	89,2 ±0,8	91,2 ±1,0	90,4 ±1,1
	Attention-based MIL-Guided			**94,0 ±0,6**	**93,4 ±0,5**	**94,2 ±0,6**	**93,2 ±0,4**
	Attention-based MIL-Guided		•	92,8 ±0,6	92,2 ±0,4	92,9 ±0,6	91,8 ±0,8
	Attention-based MIL-Guided	•		91,7 ±1,1	90,6 ±0,9	92,2 ±0,8	91,2 ±1,1
	Attention-based MIL-Guided	•	•	88,3 ±1,2	87,2 ±1,1	89,7 ±2,7	88,6 ±2,9

A maior parte das configurações baseadas no *Two-WAM* melhoram o desempenho de clas-

sificação sobre as opções baseadas no Grad-CAM. A melhor localização e, com isso, a geração de instâncias mais propícias ao processo, mostra que o *Bag Model* com uso do *Attention-based MIL-Guided* tem vantagem sobre o do *MIL-Guided* para o processo como um todo. Esse resultado ilustra a relevância da camada baseada em atenção. O melhor resultado alcançado pelos cortes do Grad-CAM foi de 91,8% de acurácia e 91,0% de medida F1.

A Tabela 6.6 mostra a comparação entre quatro elementos ao mesmo tempo. São exibidas as opções de instâncias produzidas por (i) Grad-CAM ou *Two-WAM*, (ii) *Instance Models* usando o *Two-WAM* como módulo de atenção espacial ("*Attention-based MIL-Guided*") e o impacto da (iii) NCPB ou (iv) do *fine tuning* ("FT") do *Bag Model* no treinamento com instâncias. Os resultados mostraram que, adicionar o *Two-WAM* ao *Instance Model* ou empregá-lo para cortar instâncias, causa uma mudança no paradigma de treinamento entre o *Attention-based MIL-Guided* e o *MIL-Guided*. Em (i), pretendeu-se estabelecer o melhor mapa de ativação para geração de instâncias pelo *Patch-SaliMap* (Subseção 5.1.5). O melhor modelo usando o *MIL-Guided* foi dado pelos cortes do Grad-CAM (primeira linha de resultados da Tabela 6.6), enquanto os cortes do *Two-WAM* (décima linha contando apenas os resultados da Tabela 6.6) tiveram melhor desempenho no *Attention-based MIL-Guided*. Uma possível razão pela qual isso ocorreu é que cada *Bag Model* aprendeu a extrair características adequadas para a estratégia utilizada e gerou as instâncias de forma que as regiões por elas contidas fossem as regiões mais discriminativas também para o treinamento do *Instance Model*, que contém a mesma arquitetura do *Bag Model*, com isso, capacidade semelhante de derivar características das mesma regiões.

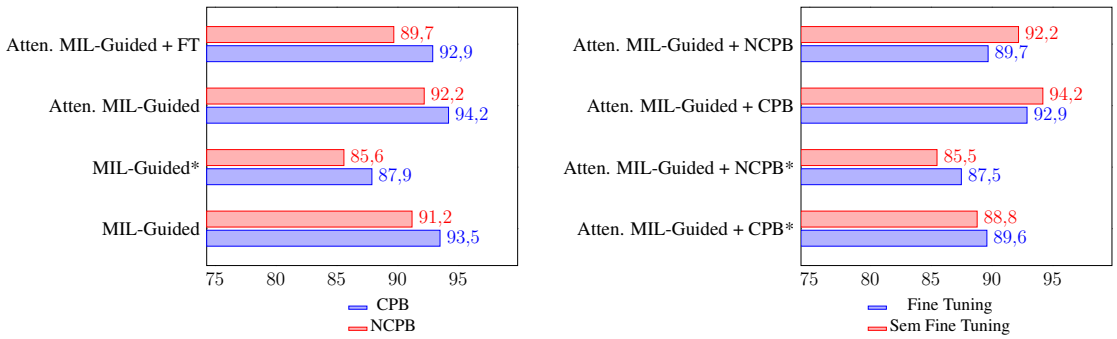
A melhor opção para o *MIL-Guided* é a opção relativa ao *fine tuning*, enquanto para o *Attention-based MIL-Guided* é o *fine tuning* de um modelo pré-treinado na ImageNet.

A Figura 6.3a mostra o melhor desempenho de modelos que consideram imagens da CPB no treinamento ("NCPB Validação – Acurácia"). Esse comportamento é oposto ao obtido nos experimentos do *Bag Model*. Assim, é necessário cuidado para lidar com pequenas RIs na presença de imagens ruidosas, ou seja, imagens capturadas em condições naturais. Além disso, os resultados sugerem que as RIs salientes se beneficiariam dos ruídos nas imagens. O melhor resultado para o *Instance Model* do *Attention-based MIL-Guided* treinado com instâncias da "NCPB" é de 91,7% de acurácia e 90,6% de medida F1 no conjunto de instâncias recortadas da CPB e 92,2% de acurácia e 91,2% de medida F1 nas instâncias da NCPB.

A Figura 6.3b apresenta os resultados referentes ao *fine tuning* do *Bag Model* no treinamento do *Instance Model*. O *fine tuning* não melhora o desempenho da classificação usando instâncias apontadas pelo *Two-WAM*, mas melhora usando instâncias apontadas pelo Grad-CAM. O melhor resultado da estratégia de *fine tuning* é 92,8% de acurácia e 92,2% de medida F1 no conjunto de validação da CPB e 92,9% de acurácia e 91,8% medida F1 no conjunto de validação da NCPB. O melhor resultado foi alcançado para o *Instance Model* usando o *Two-WAM* e todas as imagens da CPB para treinamento. Isso melhorou a acurácia em 2,2 pontos percentuais, atingindo 94,0% de acurácia e 93,4% de medida F1 no conjunto de validação da CPB.

Comparando a acurácia e a medida F1 da Tabela 6.5 com as mesmas medidas da Tabela 6.6, pode-se observar uma diferença de até 3,9 pontos percentuais em relação às duas medidas da Tabela 6.6. A acurácia balanceada, métrica mais adequada em dados desbalanceados, calcula a média da proporção correta de cada classe (taxas de verdadeiros positivos e verdadeiros negativos), enquanto que a medida F1 não considera os verdadeiros negativos. Então, diferenças podem ser notadas entre as métricas da medida F1 e acurácia.

Os resultados da avaliação do *Bag Model* e *Instance Model* parecem contraditórios, porém,



(a) Ruído no treinamento do *Instance Model* (b) *Fine tuning* do *Bag Model* para o *Instance Model*

Figura 6.3: Os efeitos da: (a) remoção de imagens ruidosas e (b) aplicação do *fine tuning* do *Bag Model* no treinamento do *Instance Model*. "*" representa os experimentos usando Grad-CAM para produzir instâncias para o *Instance Model*, enquanto que a não existência de marca indica o uso do *Two-WAM*. "*Atten. MIL-Guided*" refere-se aos modelos treinados com o *Two-WAM* (*Attention-based MIL-Guided*); "NCPB" refere-se aos experimentos treinados apenas com instâncias da NCPB; "CPB" refere-se aos experimentos treinados em instâncias da CPB; e "FT" refere-se aos experimentos que adaptaram os *Bag Models* para gerar os *Instance Models*. Figura modificada de Bollis et al. [25].

é possível que isso se deva aos ruídos nas imagens originais da CPB. Se o ruído cobrir a área do ácaro, isso afetará a previsão e o processo de treinamento. É difícil distinguir entre pequenas regiões e é ainda mais difícil distinguir se elas apresentarem ruídos. Por outro lado, as instâncias usadas para treinar o *Instance Model* já foram filtradas pelo *Bag Model* e, provavelmente, contêm ácaros mais fáceis de se identificar, ocupando uma parcela significativa das instâncias. Assim, as instâncias borradas ou com problemas de luminosidade tornam-se amostras mais desafiadoras para o treinamento.

Os experimentos utilizando a NCPB, ou seja, removendo as imagens ruidosas do treinamento dos modelos, foram essenciais para a resposta da questão de pesquisa Q4: *"Os ruídos como luminosidade e borramento, presentes na captura de imagens em campo, afetam o treinamento dos modelos de redes neurais profundas?"* e aproximaram esta tese de cumprir os objetivos O4: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"* e O2: *"verificar a importância do tamanho das regiões de interesses nas imagens"*.

6.2.3 Métodos Fracamente Supervisionados para Classificação Binária da CPB

Nesta subseção, o *Attention-based MIL-Guided* foi comparado com dois métodos fracamente supervisionados largamente estudados da literatura, o *Attention-based Deep MIL* e o WILDCAT (Seção 3.2), considerando o conjunto de *teste* da CPB. Foram explorados dois cenários: (i) imagens da CPB com tamanho 800×800 pixels sem aumento de escala e tamanho do *batch* equivalente a uma imagem ou número de instâncias equivalentes a uma imagem; e (ii) imagens da CPB com tamanho 1200×1200 pixels (tamanho original) e número máximo de imagens ou instâncias que uma GPU pode suportar em um *batch*. Esses dois cenários foram escolhidos devido à sobrecarga de memória no treinamento dos modelos do método *Attention-based Deep MIL* usando a arquitetura LeNet [94] como extratora de características. A arquitetura referente ao

Attention-based Deep MIL original contém 552 milhões de parâmetros (Tabela 6.7). Este fato, juntamente com o tamanho das imagens da CPB (1200×1200 pixels), inviabilizou a realização do experimento na GPU disponível.

Tabela 6.7: Acurácia de classificação (em %) e medida F1 (em %) de diferentes métodos fracamente supervisionados no conjunto de testes da CPB. "Param. (M)" significa o número de parâmetros em milhões para cada método. "Treinados na NCPB" significa que os experimentos foram treinados usando a NCPB e avaliados na CPB e "Melhores Abordagens" refere-se ao melhor processo de treinamento anteriormente verificado (Seção 6.2.2). Os destaques correspondem aos melhores resultados.

Métodos Fracamente Supervisionados	800×800		1200×1200		
	Acur. (%)	F1 (%)	Acur. (%)	F1 (%)	Param. (M)
Treinados na NCPB					
Attention-based Deep MIL (LeNet) [83]	63,6 ±0,5	62,2 ±0,8	–	–	552,0
Attention-based Deep MIL (EfficientNet-B0)	63,3 ±4,1	66,8 ±5,4	67,1 ±3,3	63,4 ±2,7	10,0
WILDCAT (ResNet-101) [56]	65,5 ±4,9	55,4 ±2,6	71,9 ±3,3	68,2 ±6,9	42,5
WILDCAT (EfficientNet-B0)	70,0 ±2,6	67,9 ±1,9	76,4 ±0,2	73,0 ±1,2	**4,03**
Attention-based MIL-Guided (*Bag Model*)	74,1 ±3,4	72,4 ±3,1	79,2 ±1,5	76,6 ±1,5	4,05
Attention-based MIL-Guided (*Bag + Inst. Models*)	78,1 ±2,3	76,8 ±1,9	90,2 ±1,0	89,0 ±1,0	8,1
Melhores Abordagens					
MIL-Guided (*Bag + Inst. Models*)	78,8 ±1,8	73,4 ±3,3	90,9 ±1,2	89,0 ±1,6	8,1
Attention-based MIL-Guided (*Bag + Inst. Models*)	**82,1** ±1,2	**80,1** ±1,1	**92,4** ±0,7	**91,8** ±0,8	8,1

O método *Attention-based Deep MIL* requer instâncias para treinar os modelos (*patches* recortados da imagem original sem sobreposição) e o conjunto destas instâncias representam uma *bag* que é inserida inteiramente como um *batch*. Assim, as imagens foram recortadas para terem o exato tamanho da imagem que os *Bag Models* foram treinados, ou seja, (i) 9 instâncias com tamanho maiores de *patches* e (ii) mais de 500 instâncias com tamanhos de *patch* de 32×32 pixels, o menor tamanho possível como indicado pelo trabalho original de Ilse et al. [83]. Para efeito de comparação, foi considerada a maior pontuação alcançada entre os dois tipos de cortes. No primeiro cenário (800×800), foram aplicados tamanhos de *patch* de 266×266 pixels, totalizando 9 instâncias, e 32×32 pixels, resultando em 625 instâncias. No segundo cenário (1200×1200), as imagens foram cortadas em 9 instâncias de 400×400 pixels e 1444 instâncias de 32×32 pixels. Para o *Attention-based MIL-Guided*, foram usados 5 instâncias de 266×266 pixels para o primeiro cenário e 5 instâncias de 400×400 pixels para o segundo cenário.

Os resultados foram organizados em dois grupos: (i) todos os modelos foram treinados na NCPB e avaliados em seu conjunto de teste (o mesmo da CPB), e (ii) os resultados do melhor modelo alcançado na Seção 6.2.2 (*Bag Model* treinado na NCPB e *Instance Model* treinado na CPB) foi comparado com a melhor arquitetura obtida anteriormente para a CPB na Subseção 6.1.4. O *Attention-based MIL-Guided* ("*Bag + Instance Models*", 92,4% de acurácia na Tabela 6.7) supera os métodos *Attention-based Deep MIL* e WILDCAT em todos os cenários em até 25,3 pontos percentuais. O *Bag Model* do *Attention-based MIL-Guided* (79,2% de acurácia) supera os mesmos métodos em até 12,1 pontos percentuais.

O melhor resultado em ambos os cenários para o *Attention-based Deep MIL* é de 67,1% de acurácia e 63,4% de medida F1, usando 9 instâncias em imagens de tamanho 1200×1200

pixels. O único resultado para instâncias de tamanho pequeno (32×32 pixels), que foi melhor que o seu análogo usando instâncias grandes (266×266), é de 63,3% de acurácia e 66,8% de medida F1 (segunda linha de resultados da Tabela 6.7) e usou 25 instâncias. Para o WILDCAT, o melhor resultado é de 76,4% para acurácia e 73,0% para medida F1. O *Bag Model* do *Attention-based MIL-Guided* fornece melhores mapas de ativação e resultados do que os métodos da literatura, atingindo 79,2% de acurácia e 76,6% de medida F1. Considerando o melhor resultado geral ("Melhores Abordagens"), no qual o *Bag Model* é treinado em imagens da NCPB e o *Instance Model* em imagens da CPB, o *Attention-based MIL-Guided* chegou a 92,4% de acurácia e 91,8% de medida F1, superando o melhor resultado do *MIL-Guided*, 90,9% de acurácia e 89,0% de medida F1. Além disso, a diferença de até 12,4 pontos percentuais entre o *Instance Model* (89,0%) e o *Bag Model* (76,6%) mostra o desempenho de classificação do *Attention-based MIL-Guided*.

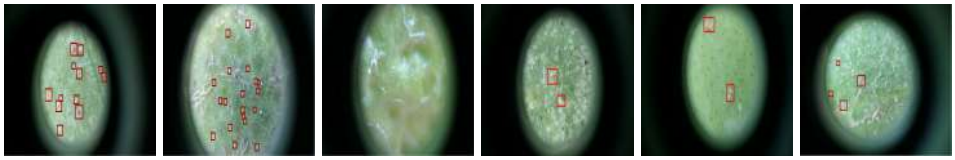
É importante ressaltar a diferença de pelo menos 5 pontos percentuais entre os resultados dos dois cenários, 800×800 pixels e 1200×1200 pixels. O tamanho das imagens do primeiro cenário e a não aplicação do aumento de escala impactaram negativamente na capacidade de aprendizado das DNNs, o que diminuiu o desempenho final de classificação.

A Figura 6.4 fornece uma comparação qualitativa para a localização fracamente supervisionada produzida utilizando a EfficientNet-B0 como extrator de características dos métodos fracamente supervisionados comparados nesta seção. A Figura 6.4a mostra os locais dos ácaros anotados manualmente (retângulos vermelhos), incluindo uma imagem sem ácaros (terceira imagem da esquerda para a direita).

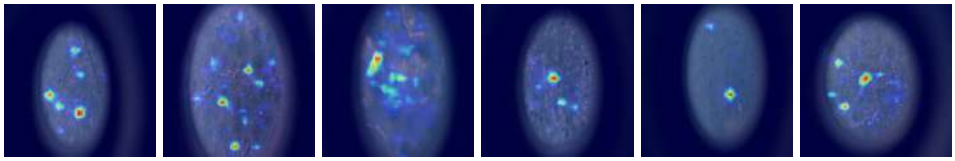
Os resultados dos mapas de saliências produzidos pelo *Attention-based MIL-Guided* são mostrados na Figura 6.4b. É possível verificar que o *Two-WAM* infere um número maior de regiões que têm alta probabilidade de conter ácaros do que os métodos da literatura. As regiões demarcadas em vermelho se adaptam melhor aos corpos dos ácaros, enquanto as regiões apontadas pelos outros métodos normalmente transbordam suas áreas corporais. Possivelmente, esse comportamento é devido à melhor localização oferecida pelo *Two-WAM*. Na Figura 6.4c, pode-se notar que as áreas vermelhas produzidas pelo Grad-CAM têm tamanhos maiores do que os apontados pelo *Attention-based MIL-Guided*, mas, na maior parte dos casos, as áreas vermelhas mostram RIs corretas. No entanto, o número de possíveis localizações é significativamente menor do que o número de retângulos vermelhos. Além disso, o Grad-CAM apresenta falhas significativas em suas localizações de ácaros, como a imagem mais à direita da Figura 6.4c, onde as regiões apontadas estão um pouco distantes das localizações corretas dos ácaros. Diferentemente, o WILDCAT mostra regiões mais confiáveis do que as regiões apresentadas pelo Grad-CAM, entretanto, ainda apresenta grandes áreas destacadas e um número de RIs muito menor do que o número de ácaros, como pode ser observado na Figura 6.4d. A Figura 6.4e mostra os mapas de saliências do *Attention-based Deep MIL* criados com base no peso de atenção de cada instância usada para classificação de suas *bags*, conforme explicado na Seção 3.1. As cores representam o quanto cada instância influencia na predição final. As regiões onde possivelmente existem ácaros aparecem em tons de vermelho ou amarelo.

Considerando as regiões que o *Two-WAM* inferiu para a imagem sem ácaros, pode-se observar que ele produziu mais regiões, o que também ocorre nos demais mapas de ativação do *Two-WAM*. Esse aspecto é útil para o treinamento de um *Instance Model* mais robusto, pois mais amostras corretas são consideradas. Como essas amostras representam casos difíceis para o *Bag Model*, o *Instance Model* do *Attention-based MIL-Guided* pode analisar mais profun-

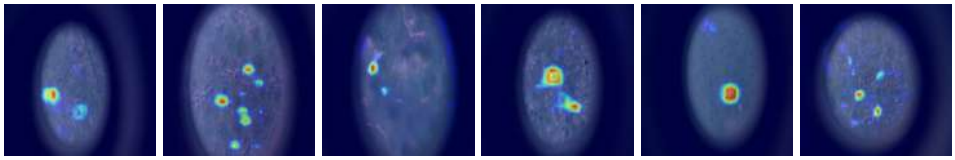
damente as instâncias (imagens recortadas dos pontos de dúvida) para detectar a existência de falsos positivos. Embora o *Two-WAM* gere mais RIs, seu número não é tão grande quanto o número de ácaros. Ainda assim, o *Two-WAM* é mais confiável na identificação das áreas onde os ácaros possivelmente estão localizados.



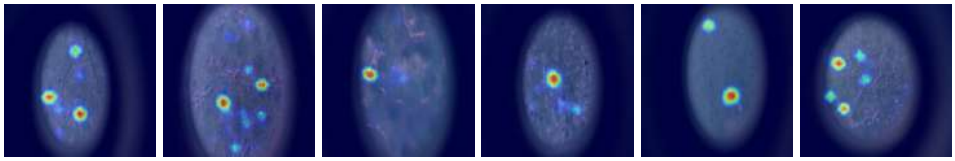
(a) Exemplos de Imagens da CPB [24]



(b) *Attention-based MIL-Guided* (*Two-WAM*)



(c) *MIL-Guided* (Grad-CAM) [24]



(d) WILDCAT [56]

(e) *Attention-based Deep MIL* (ativação das instâncias) [83]

Figura 6.4: (a) Imagens da CPB. (b) Mapas de saliências do *Attention-based MIL-Guided* gerados pelo *Two-WAM*. (c) Mapas de saliências do *MIL-Guided* gerados pelo Grad-CAM. (d) Mapas de saliências gerados pelo WILDCAT. (e) Mapas de saliências baseados em pesos de ativação de instâncias do *Attention-based Deep MIL*. Todos os mapas de saliências foram gerados usando a EfficientNet-B0 como extratora de características. Os mapas do *Attention-based Deep MIL* mostram as instâncias mais importantes para a tarefa de classificação, conforme explicado na Seção 3.1. A terceira coluna contém uma classe negativa e ilustra os falsos positivos de localizações de ácaros. Os retângulos vermelhos em (a) ilustram as localizações dos ácaros anotadas manualmente. Figura traduzida de Bollis et al. [25].

Os experimentos relativos a esta seção estabeleceram o estado da arte para a tarefa binária da base CPB, ajudando a cumprir o objetivo O4: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"* e tornam-se parte da contribuição C5: "resultados competitivos em relação aos métodos disponíveis na literatura para duas bases de dados distintas". Os experimentos aqui relatados também são parte dos esforços para cumprir os objetivos O2: *"verificar a importância do tamanho das regiões de interesse nas imagens"* e O5: *"mostrar a viabilidade do uso de métodos fracamente supervisionados na localização de pragas em imagens"*.

6.2.4 Métodos Fracamente Supervisionados Aplicados à IP102 1.1

Nesta subseção, métodos fracamente supervisionados foram avaliados no conjunto de dados de *teste* da IP102 1.1. Os resultados do *Attention-based MIL-Guided* e *MIL-Guided* foram comparados com o *Attention-based Deep MIL* e WILDCAT.

Os modelos desta seção foram treinados com o maior tamanho de *batch* possível para cada método. Ressalta-se que os experimentos para a base de dados IP102 1.1 contêm imagens muito menores do que a CPB, ou seja, de tamanho 224×224 pixels. Para o *Attention-based Deep MIL*, as imagens foram divididas em 4 instâncias de 112×112 pixels, 9 instâncias de 74×74 pixels e 16 instâncias de 56×56 pixels. Foi relatado somente o melhor resultado de acurácia, pois não houve diferença relevante relacionada ao número de cortes. Em relação ao *Attention-based MIL-Guided*, foram usados 5 *patches* de 74×74 pixels, ou seja, 1/9 das imagens originais, como proposto para a CPB.

A Tabela 6.8 mostra que o *Bag Model* do *MIL-Guided* obteve a melhor acurácia e medida F1 na IP102 1.1, seguido pelo *Bag Model* do *Attention-based MIL-Guided*. O *Attention-based Deep MIL* usando a LeNet [83] alcançou as pontuações mais baixas, 23,0% de precisão e 24,5% de medida F1. Os resultados considerando 4 e 16 instâncias para a arquitetura EfficientNet-B0 [173], não relatados na Tabela 6.8, foram, respectivamente, de 37,8% de acurácia e 37,7% de medida F1 e 36,3% de acurácia e 36,5% de medidas F1. Do mesmo modo, os resultados relatados para o *Attention-based MIL-Guided* ("*Bag + Instance Model*") foram muito baixos. Uma possível explicação para isso é que as RIs na base IP102 1.1 são geralmente maiores do que nas imagens da CPB. Cada instância contém apenas parte dessas regiões, conforme ilustrado em 6.6g. Da mesma forma, *Instance Models* apresentam piores resultados do que *Bag Models* na IP102 1.1.

Tabela 6.8: Acurácia de classificação (em %) e medida F1 (em %) de diferentes métodos fracamente supervisionados no conjunto de *teste* da IP102 1.1. "Param. (M)" indica número de parâmetros em milhões. Os destaques correspondem aos melhores resultados.

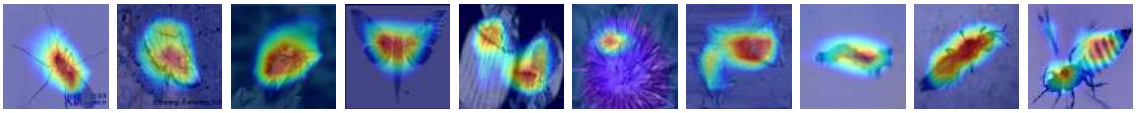
Métodos Fracamente Supervisionados	224 × 224		
	Acur. (%)	F1 (%)	Param. (M)
Attention-based Deep MIL (LeNet) [83]	23,0 ±1,1	24,5 ±1,0	552,0
Attention-based Deep MIL (EfficientNet-B0)	38,7 ±3,3	36,8 ±3,8	10,0
MIL-Guided (*Bag + Instance Model*)	64,8 ±0,2	64,0 ±0,1	8,1
Attention-based MIL-Guided (*Bag + Instance Model*)	58,8 ±2,5	58,2 ±2,3	8,1
WILDCAT (ResNet-101) [56]	67,6 ±0,4	67,5 ±0,6	42,5
WILDCAT (EfficientNet-B0)	65,1 ±0,4	64,4 ±0,9	**4,03**
MIL-Guided (*Bag Model*)	**69,5** ±0,1	**69,0** ±0,1	4,05
Attention-based MIL-Guided (*Bag Model*)	68,3 ±0,3	68,0 ±0,1	4,05

Em relação ao WILDCAT, seus resultados estão mais próximos dos *Bag Models* do *MIL-Guided* e *Attention-based MIL-Guided*. Isso significa 65,1% de acurácia e 64,4% de medida F1 com a EfficientNet-B0 e 67,6% de acurácia e 67,5% de medida F1 com ResNet-101. O *MIL-Guided* superou em 1 ponto percentual o *Attention-based MIL-Guided*. Ambos os *Bag Models* superaram o WILDCAT em pelo menos 2 pontos percentuais.

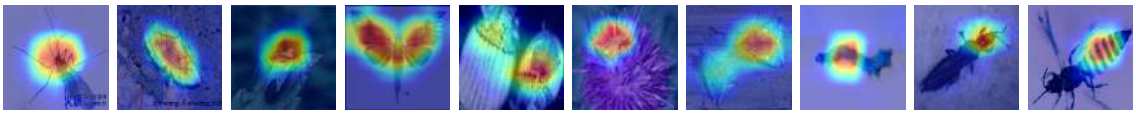
A Figura 6.5 ilustra porque o desempenho de classificação do *Two-WAM* reduziu 1 ponto percentual em imagens de insetos salientes (Tabela 6.8), ou seja, que ocupam grande parte das

imagens. Como a literatura descreve [161] e a Tabela 6.5c mostra, as saliências geradas pelo Grad-CAM mapeiam áreas que influenciam fortemente as predições finais. Isso significa que o Grad-CAM geralmente destaca um número pequeno de regiões (áreas vermelhas), onde estão localizadas as características mais descritivas para classificação. Por outro lado, a Figura 6.5b (*Two-WAM*) destaca áreas diferentes, mas essas áreas sempre têm uma intersecção com as áreas geradas pelo Grad-CAM. Portanto, as regiões geradas pelo *Attention-based MIL-Guided* também podem conter características que não são discriminativas o suficiente ou que atrapalhem a classificação. Por esse motivo, a melhor localização gerada pelo *Two-WAM* influenciou negativamente na tarefa de classificação.

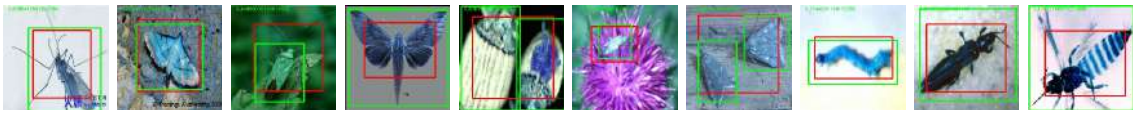
(a) Exemplos de Imagens da IP102 1.0 [197]



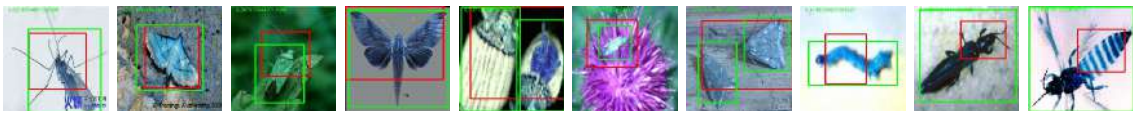
(b) *Attention-based MIL-Guided* (*Two-WAM*)



(c) *MIL-Guided* (Grad-CAM) [24]



(d) *Attention-based MIL-Guided* (*Two-WAM*) *bounding boxes*



(e) *MIL-Guided* (Grad-CAM) *bounding boxes* [24]

Figura 6.5: (a) Imagens da IP102 1.0. (b) Mapas de saliências do *Attention-based MIL-Guided* produzidos pelo *Two-WAM*. (c) Mapas de saliências do *MIL-Guided* produzidos pelo Grad-CAM. (d) *Bounding boxes* para o *Attention-based MIL-Guided*. (e) *Bounding boxes* para o *MIL-Guided*. Os retângulos verdes são as marcações da base IP102 1.0 e os vermelhos são as inferências dos métodos. Os valores em verde (no canto esquerdo superior) são as intersecções sobre as uniões (*intersection over union*, IoU). As imagens da IP102 1.0 foram utilizadas, pois as anotações referentes à localização na IP102 1.1 não estão disponíveis. Figura traduzida de Bollis et al. [25].

Além disso, as áreas vermelhas destacadas pelo Grad-CAM estão centradas em algumas regiões dos insetos e geralmente excedem seus corpos. Em contraste, as regiões cobertas pelo *Two-WAM* são desenhadas ao longo das áreas dos corpos dos insetos e se ajustam melhor a esses corpos. As áreas grandes em vermelho, produzidas pelo Grad-CAM, podem indicar que os contornos dos corpos dos insetos são essenciais para a produção de características de classificação. Portanto, como mencionado anteriormente, os resultados desta seção sugerem que a capacidade de localização para se adaptar melhor aos corpos dos insetos ou ácaros não melhora pontualmente a classificação, porém melhora a etapa de localização e recorte de instâncias, que indiretamente melhora o *Instance Model* do *Attention-based MIL-Guided*. O *Two-WAM*

mostrou mais de uma área vermelha em alguns casos, enquanto o Grad-CAM mostrou apenas regiões conectadas para dois ou mais insetos.

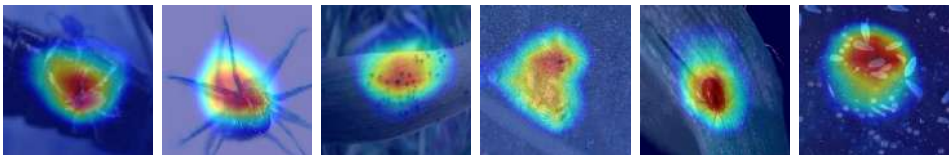
A Figura 6.5d mostra o desempenho da localização do *Two-WAM* na geração de localizações ou *bounding boxes* (caixas delimitadoras) com base nos mapas de saliências por ele produzidos. A Figura 6.5e, a qual exibe os *bounding boxes* gerados pelo Grad-CAM, mostra áreas menos precisas, que, na maior parte do tempo, contêm apenas uma pequena parte dos corpos dos insetos. O *Two-WAM* gera *bounding boxes* mais precisos e infere regiões de localização para fornecer resultados mais confiáveis, e sem o uso de rótulos de localização. No entanto, o gerador de *bounding boxes* deste capítulo, inspirado em Lu et al. [121], não produz mais de um *bounding box* por imagem.

A Figura 6.6 mostra exemplos de mapas de saliências para ácaros da IP102 1.1. Ressalta-se que algumas imagens possuem RIs menores, mas não são tão pequenas quanto as imagens de ácaros da CPB. O *Attention-based MIL-Guided*, ou seu método interno *Two-WAM*, geralmente destaca todos os ácaros (Figura 6.6b), entretanto, de forma semelhante aos outros métodos, não localiza um dos ácaros na primeira coluna da esquerda para direita e não destaca o ácaro inteiro na segunda coluna. O *MIL-Guided* (Figura 6.6e), que usa o Grad-CAM, foca em certas regiões, uma no máximo, e não destaca áreas inteiras. Os mapas de saliências do WILDCAT (Figura 6.6f) mostram regiões mais concisas e que se ajustam melhor aos grupos de ácaros. No entanto, as regiões marcadas não são cobertas inteiramente nos casos em que os ácaros estão espalhados e, geralmente, os contornos dos corpos dos ácaros são transbordados. A Figura 6.6g ilustra o número de instâncias que foram ativadas por seus pesos e que mais influenciaram nas predições do *Attention-based Deep MIL*, entretanto, elas nunca estão todas marcadas em vermelho ao mesmo tempo.

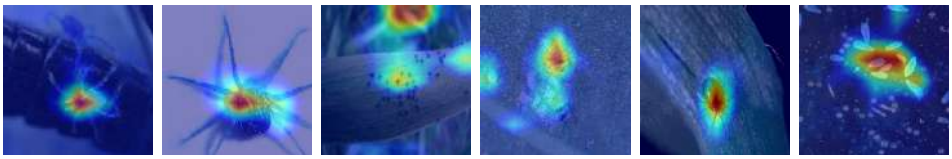
As Figuras 6.6c e 6.6d ilustram o comportamento dos modelos da CPB ao inferir mapas de saliências para ácaros da IP102 1.1. Essa comparação mostra que os modelos treinados na CPB podem identificar a localização correta dos ácaros em um conjunto de dados gerado com requisitos diferentes dos requisitos da CPB. No entanto, o treinamento em outra base propiciou a inferência e aparecimento de erros, como aqueles nas imagens do meio das Figuras 6.6c e 6.6d. Isso significa que os modelos da CPB aprenderam a inferir quando os ácaros estão presentes nas imagens e onde eles estão, mas apenas destacaram pequenas regiões porque trouxeram o viés da CPB. Na última coluna da Figura 6.6d, o *Instance Model* treinado em instâncias da CPB evidenciou cada pequeno grupo de ácaros, mantendo uma demarcação mais refinada do que os outros métodos. Os ácaros mostrados na última coluna da figura em questão são ácaros da ferrugem, também presentes na CPB.

Os experimentos relativos a esta subseção estabeleceram o estado da arte para métodos fracamente supervisionados na IP102 1.1, ajudando a cumprir o objetivo O4: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"* e tornam-se parte da contribuição C5: *"resultados competitivos em relação aos métodos disponíveis na literatura para duas bases de dados distintas"*. Os experimentos desta subseção finalizam o objetivo O2: *"verificar a importância do tamanho das regiões de interesse nas imagens"* e respondem a questão de pesquisa Q5: *"uma localização fracamente supervisionada mais acurada das pragas implica em maior taxa de classificação?"*.

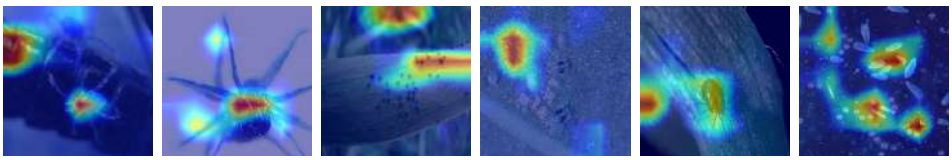
(a) Exemplos de Imagens da IP102 1.1 [197]



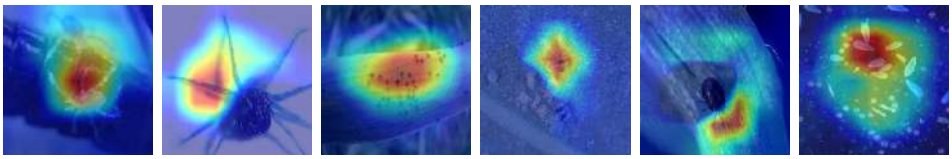
(b) *Attention-based MIL-Guided* (*Two-WAM*)



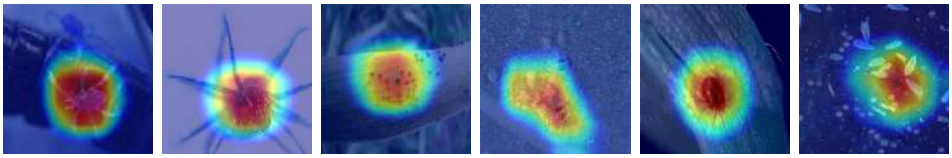
(c) *Attention-based MIL-Guided* (*Two-WAM* com *Bag Model* treinado na CPB)



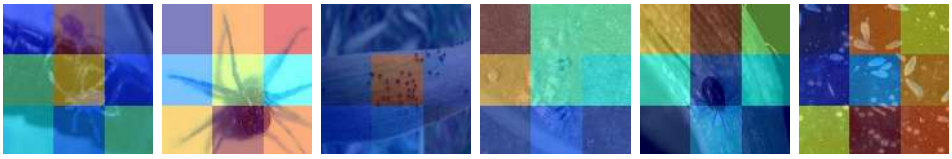
(d) *Attention-based MIL-Guided* (*Two-WAM* com *Instance Model* treinado na CPB)



(e) *MIL-Guided* (Grad-CAM) [24]



(f) WILDCAT [56]



(g) *Attention-based Deep MIL* (ativação das instâncias) [83]

Figura 6.6: (a) Imagens da IP102 1.1 [197]. (b) Mapas de saliências do *Attention-based MIL-Guided* gerados pelo *Two-WAM*. (c) Mapas de saliências do *Attention-based MIL-Guided* gerados pelo *Two-WAM* com *Bag Model* treinado na CPB. (d) Mapas de saliências do *Attention-based MIL-Guided* gerados pelo *Two-WAM* com *Instance Model* treinado na CPB. (e) Mapas de saliências do *MIL-Guided* gerados pelo Grad-CAM. (f) Mapas de saliências gerados pelo WILDCAT. (g) Mapas de saliências baseados em pesos de ativação de instâncias do *Attention-based Deep MIL*. Todos os mapas de saliências foram gerados usando a EfficientNet-B0 como extratora de características. As imagens em (c) e (d) mostram que os modelos treinados na CPB podem inferir a localização de ácaros mesmo em uma base diferente da treinada, apesar de alguns erros. Figura traduzida de Bollis et al. [25].

6.2.5 Localização Fracamente Supervisionada de Insetos da IP102 1.0

Esta subseção compara os resultados de localização presentes na literatura com os resultados de localização fracamente supervisionada produzidos pelo *Bag Model* do *Attention-based MIL-Guided* para a base de dados IP102 1.0 [197]. A versão da IP102 1.1 não foi utilizada, pois as anotações referentes à localização não estão disponíveis. A precisão média (AP), medida padrão de localização para a base, foi utilizada para as análises. Avaliou-se a intersecção sobre a união (IoU) como uma medida puramente de localização para os propósitos desta tese de doutorado. A AP foi calculada considerando diferentes limiares de IoU: média, 50% e 75%.

A base de dados IP102 1.0 apresenta diferentes conjuntos para os protocolos de localização de objetos e classificação de imagens. Na localização de objetos, são apresentados os conjuntos de treinamento/validação e teste, enquanto que, na classificação, são apresentados os conjuntos de treinamento/validação e teste [197]. Imagens contidas nas partições do conjunto de treinamento/validação e teste da localização estão presentes nos três conjuntos do protocolo de classificação. Assim, as imagens do conjunto de teste de localização foram removidas do conjunto de treinamento e validação da tarefa de classificação. Isso ocorreu para manter todas as imagens do conjunto de teste de localização fora dos treinamentos.

As Tabelas 6.9 e 6.10 apresentam os resultados de duas opções de treinamento do *Bag Model* para o *Attention-based MIL-Guided*: (i) uso do conjunto de treinamento/validação segundo a tarefa de localização (15.178 imagens) e (ii) uso do conjunto de treinamento segundo a tarefa de classificação, mas sem as imagens que também estavam no conjunto de teste da tarefa de localização (42.648 imagens). Essas duas opções foram exibidas para avaliar corretamente as localizações relativas ao protocolo original da base IP102 1.0. Elas também foram utilizadas para mostrar que o uso de um número maior de imagens pode melhorar a localização fracamente supervisionada.

Tabela 6.9: Precisão média (AP em %) para diferentes limiares da intersecção sobre união (IoU) mensurada no conjunto de *teste* de localização da IP102 1.0. O *Bag Model* indicado como "original" representa o treinamento com imagens do conjunto de treinamento/validação do protocolo original de localização da IP102 1.0 (15.178 imagens). "+ imagens" indica o treinamento com as imagens do conjunto de treinamento para classificação da IP102 1.0 e que não estão no conjunto de testes de localização (42.648 imagens), ou seja, o treinamento ocorreu com mais imagens. Nenhum dos dois treinamentos dos *Bag Models* utilizou rótulos de localização.

Método	Extrator	AP	AP^{50}	AP^{75}
Totalmente Supervisionados				
RPN [148]	VGG-16 [167]	21,1	47,9	15,2
FPN [109]	ResNet-50 [71]	28,1	54,9	23,3
SSD300 [114]	VGG-16 [167]	21,5	47,2	16,6
RefineDet [221]	VGG-16 [167]	22,8	49,0	16,8
YOLO-v3 [146]	DarkNet-53 [146]	25,7	50,6	21,8
Fracamente Supervisionados				
Attention-based MIL-Guided (original)	EfficientNet-B0 [173]	15,6	41,3	11,8
Attention-based MIL-Guided (+ imagens)	EfficientNet-B0 [173]	19,1	48,0	10,9

A Tabela 6.9 mostra a comparação entre métodos de localização totalmente supervisiona-

dos (“Totalmente Supervisionados”) da literatura com modelos do *Attention-based MIL-Guided* (“Fracamente Supervisionados”), que produzem localizações fracamente supervisionadas.

A abordagem RPN [148] apresenta uma rede de propostas de regiões que compartilha características das convoluções entre a classificação e a localização de objetos, permitindo que as propostas de regiões sejam feitas quase sem custos. Ela foi criada para ser mais rápida e simples do que suas antecessoras.

A abordagem FPN [109] explora a pirâmide hierárquica multiescala de mapas de características para redes convolucionais profundas. Ela pode ser usada como extratora de características para qualquer rede neural, porém, foi utilizada na FPN conjuntamente com uma RPN para derivar mapas de características utilizando diversas escalas e inferir a localização de objetos.

A abordagem SSD [109] discretiza o espaço de saída das inferências de localizações (*bounding boxes*) em um conjunto padrão contendo diferentes proporções e escalas por localização do mapa de características. A rede gera pontuações para cada classe de objeto e, através dessas pontuações, produz ajustes nas localizações, escolhendo os *bounding boxes* mais adequados.

A abordagem RefineDet [221] consiste em um módulo de refinamento de localizações, providas por meio de âncoras, interconectado a um módulo de detecção de objetos. O primeiro filtra âncoras negativas e ajusta as âncoras para fornecer uma melhor inicialização para a etapa de regressão. O segundo módulo refina as âncoras para melhorar ainda mais a regressão e prever as múltiplas classes.

A abordagem YOLO-v3 [146] prediz localizações usando aglomerados de dimensões como âncoras. Eles melhoraram a versão anterior através de pequenas alterações. A YOLO-v3 usa: predição de três diferentes escalas baseadas em pirâmides de características; regressão logística para classificação multirrótulos de cada objeto; e extração de características através da DarkNet-53, que contém convoluções 3×3, 1×1 e conexões residuais.

O *Bag Model* treinado com a quantidade de imagens do conjunto original (“original”), que segue estritamente o protocolo original de localização da IP102 1.0, reportou de 6 a 14 pontos percentuais abaixo dos métodos totalmente supervisionados considerando todos os limiares. Quando são consideradas mais imagens (“+ imagens”), a proposta desta tese alcançou resultados equivalentes aos métodos RPN [148] e SSD300 [114], considerando um limiar de 50% para a IoU. Esse tipo de treinamento apenas aconteceu porque o *Attention-based MIL-Guided* trabalha com supervisão fraca. Os resultados da tabela mostram a efetividade das localizações produzidas pelo *Attention-based MIL-Guided*, ou seja, utilizando o *Two-WAM*. Porém, a AP tem um decaimento mais acentuado para limiares de IoU maiores. Isso indica que, possivelmente, as localizações fracamente supervisionadas estão nos lugares corretos. Entretanto, os *bounding boxes* gerados não se adaptam aos objetos da mesma forma que os métodos totalmente supervisionados. O melhor resultado (19,1%) para a AP média fracamente supervisionada dos modelos desta tese foi obtido pelo *Bag Model* do *Attention-based MIL-Guided* treinado com mais imagens. Esse resultado ficou de 1 a 9 pontos percentuais abaixo dos resultados referentes aos métodos treinados com rótulos de localização.

A Tabela 6.10 mostra a comparação dos resultados nos diferentes conjuntos de dados da base IP102 1.0 entre os dois tipos de treinamentos (i) (“Treinados com Imagens do Protocolo Original”) e (ii) (“Treinados com + Imagens”). O treinamento com mais imagens se sobressai na maior parte dos resultados da tabela. Considerando a IoU, o *Bag Model* treinado com mais imagens se sobressai relativamente ao treinado somente com imagens que pertencem ao conjunto de treinamento/validação da tarefa de localização. Em média, a diferença é de 4,1

pontos percentuais. Provavelmente, com um maior número de imagens que variem o tamanho dos objetos, o *Two-WAM* poderia produzir resultados ainda mais próximos dos métodos de localização totalmente supervisionada. Porém, é necessário ressaltar que a base IP102 1.0 facilita o cálculo de AP para as localizações do *Attention-based MIL-Guided*, pois cada imagem contém apenas um tipo de praga (base multiclasses) e um classificador. O *Bag Model* produz predições para a imagem inteira e não para cada localização dos objetos (cada *bounding box* diferente). A pequena variação de valores da IoU entre todos os conjuntos avaliados para os dois casos (i) e (ii) indica que é improvável a existências de sobreajuste (*overfitting*) de localização quando modelos são treinados de maneira fracamente supervisionada, ou seja, sem rótulos de localização.

Esta subseção estabelece o estado da arte para métodos fracamente supervisionados na tarefa de localização da IP102 1.0, ajudando com a contribuição C5: *"resultados competitivos em relação aos métodos disponíveis na literatura para duas bases de dados distintas"*. Os experimentos desta subseção finalizam o objetivo O5 relativamente aos insetos: *"mostrar a viabilidade do uso de métodos fracamente supervisionados na localização de pragas em imagens"* e respondem a questão de pesquisa Q2: *"métodos de aprendizado fracamente supervisionados de mapas de ativação são eficazes para localização de pragas?"*.

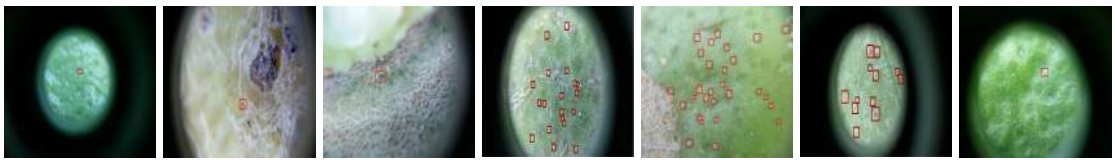
Tabela 6.10: Precisão média (AP em %) e intersecção sobre união (IoU em %) para *Bag Models* do *Attention-based MIL-Guided* mensurados nos diferentes conjuntos da IP102 1.0. Na parte superior da tabela ("Treinados com Imagens do Protocolo Original") está a avaliação do *Bag Model* treinado no conjunto de treinamento/validação segundo o protocolo de localização da IP102 1.0. Na parte inferior ("Treinados com + Imagens") está o *Bag Model* do método *Attention-based MIL-Guided* treinado no conjunto de treinamento da IP102 1.0 segundo o protocolo de classificação da base, mas sem as imagens que também estão no conjunto de testes do protocolo de localização. Para avaliação, foram utilizadas as imagens de cada conjunto que continham anotações de localização: 15.178 imagens do treinamento/validação da localização; 3.798 imagens do teste da localização; 12.256 imagens do treinamento da classificação; 1.731 imagens da validação da classificação; e 4.989 imagens do teste da classificação.

Conjunto	Tarefa	AP	AP^{50}	AP^{75}	IoU
Treinados com Imagens do *Protocolo Original*					
Treinamento/Validação	Detecção Objetos	18,0	50,5	10,8	44,2
Teste	Detecção Objetos	15,6	41,3	11,8	44,5
Treinamento	Classificação	18,2	52,3	9,3	41,5
Validação	Classificação	19,1	54,8	8,9	45,2
Teste	Classificação	20,1	53,6	11,3	41,3
Treinados com + *Imagens*					
Treinamento/Validação	Detecção Objetos	20,1	52,5	12,6	47,5
Teste	Detecção Objetos	19,1	48,0	10,9	47,3
Treinamento	Classificação	21,0	54,4	12,7	47,1
Validação	Classificação	17,6	40,9	14,5	48,1
Teste	Classificação	17,1	44,1	11,4	47,2

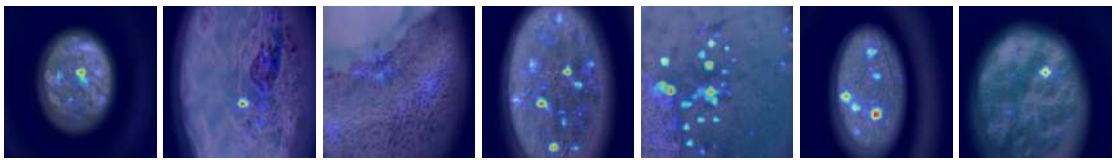
6.2.6 Localização Fracamente Supervisionada de Ácaros da CPB

Esta subseção apresenta uma avaliação qualitativa do *Attention-based MIL-Guided* em relação às localizações de ácaros produzidas pelo *Two-WAM* para as imagens da base CPB. Como a CPB é uma base para classificação, não há rótulos de localização para proporcionar uma avaliação quantitativa. O método utilizado para gerar os *bounding boxes* desta subseção foi inspirado no trabalho de Lu et al. [121], entretanto, foi modificado com a inserção de operações morfológicas de abertura e fechamento das regiões demarcadas pelos mapas de saliências. Isso foi feito para que regiões próximas não se interceptassem. Devido às múltiplas inferências de localização criadas pelo *Two-WAM* e às características dos ácaros, foi possível a geração de vários *bounding boxes* por imagem. Então, múltiplos ácaros puderam ser localizados em cada exemplo de imagem da CPB.

A Figura 6.7 mostra as localizações geradas para um conjunto de imagens originais da CPB (Figura 6.7a). A Figura 6.7b apresenta os mapas de saliências de cada uma das imagens originais. A Figura 6.7c mostra em vermelho as regiões que têm a maior probabilidade de conter ácaros. Essas regiões foram geradas a partir dos mapas de saliências através do uso de um limiar de 0,5 (quando a pontuação mostrada em cada valor do mapa de saliências é mais baixa do que o limiar, a área não é demarcada) precedido pelas operações morfológicas já citadas em sequência (abertura e fechamento). A Figura 6.7d mostra o resultado final das localizações, inclusive as marcações de localização para múltiplos objetos, ou seja, os ácaros.

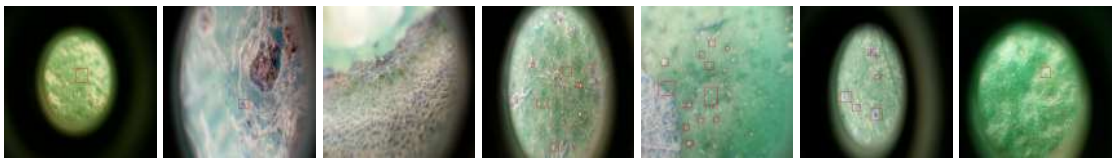


(a) Exemplos de Imagens da CPB



(b) *Attention-based MIL-Guided* (*Two-WAM*)

(c) Marcações Após Limiar



(d) Geração de Múltiplos *Bounding Boxes*

Figura 6.7: (a) Exemplos de imagens da CPB com a localização correta dos ácaros. (b) Mapas de saliências do *Attention-based MIL-Guided* produzidos pelo *Two-WAM*. (c) Marcações depois do uso do limiar de 0,5 sobre as inferências dos mapas de saliências e das operações morfológicas. (d) *Bounding boxes* para o *Attention-based MIL-Guided*.

Um problema recorrente nas imagens com mais ácaros é a falta de inferências de locali-

zações para todos esses ácaros. A quarta e quinta colunas da Figura 6.7 mostram um grande número de ácaros existentes, entretanto, a quantidade de *bounding boxes* criados pelo localizador não é tão grande quanto o número total de ácaros que aparecem nas imagens. Outros pontos importantes que devem ser considerados são: a existência de mais de um ácaro dentro de uma mesma marcação (quinta coluna); as marcações podem não cobrir totalmente o corpo dos ácaros (última coluna); e as marcações podem ser muito grandes para apenas um ácaro (primeira coluna). Apesar desses erros, o método de localização fracamente supervisionado apresentou resultados satisfatórios ao se considerar que nenhum rótulo para localização foi utilizado.

Nas imagens da Figura 6.7, apenas uma coluna representa um falso negativo para classificação (terceira coluna), onde o ácaro rajado é bem visível, entretanto, as localizações do mapa de saliências foram ativadas fracamente (terceira coluna da Figura 6.7b). Essas ativações fracas não foram suficientes para ultrapassar o limiar e gerar uma inferência adequada para a localização desse ácaro. Isso geralmente ocorre quando a predição da imagem é um falso negativo, como no caso em questão. O *Bag Model* do *Attention-based MIL-Guided* predisse, com uma pontuação de 33,0% (valor variando de 0 a 100%), que não existiam ácaros na imagem.

Os experimentos desta subseção finalizaram o objetivo O5 relativamente aos ácaros: *"mostrar a viabilidade do uso de métodos fracamente supervisionados na localização de pragas em imagens"* e reforçam a resposta para a questão Q2: *"métodos de aprendizado fracamente supervisionados de mapas de ativação são eficazes para localização de pragas?"*.

6.2.7 Problemas com os Classificadores Ponta a Ponta para a CPB

Esta subseção apresenta uma análise do *Attention-based MIL-Guided* como uma arquitetura única, treinada de ponta a ponta utilizando um erro unificado para o *Bag Model* e *Instance Model*. Para isso, as predições referentes ao *Bag Model* foram adicionadas ao Método de Avaliação Ponderada, o qual calculava anteriormente uma média ponderada utilizando pesos estáticos sobre as predições de cada instância (Seção 5.1.6). Os pesos da versão utilizada nesta seção passaram a ser calculados durante o treinamento, ou seja, foram calculados pelo método do gradiente descendente estocástico considerando as predições das instâncias e da imagem original. O *Bag Model*, *Instance Model* e *Patch-SaliMap* não sofreram alterações. O erro foi calculado depois da predição final produzida pelo agregador de predições, que substituiu o Método de Avaliação Ponderada. Essa mudança ocorreu para que as perdas fossem unificadas, calculadas e distribuídas para todas as camadas do modelo durante cada passo de treinamento. Todos os experimentos foram treinados no conjunto de validação da NCPB e avaliados no conjunto de validação da CPB.

A Tabela 6.11 mostra os resultados para o *Attention-based MIL-Guided* ("Original") e seu *Bag Model*, comparando-os com diferentes versões do processo unificado ("Unificado"). A tabela também compara diferentes configurações na geração do números de instâncias: (i) uso de 3 instâncias contendo alta probabilidade de conter ácaros, criadas com as áreas de maiores valores dos mapas de saliências, e 3 com baixa probabilidade, criadas com as áreas de menores valores ("3 positivas e 3 negativas"); e (ii) uso de 5 instâncias contendo alta probabilidade de conter ácaros. Por último, a possibilidade do *Bag Model* ser pré-treinado para produzir instâncias no começo do treinamento unificado também se faz presente na tabela ("Pré-treinado").

O melhor resultado da Tabela 6.11 é a versão treinada com múltiplas etapas, 91,7% de acurácia e 90,6% de medida F1, ou seja, a versão original (Seção 5.2). Os resultados apresentados

Tabela 6.11: Avaliação do *Attention-based MIL-Guided* criado como um processo unificado no conjunto de validação da CPB, denotado como "Unificado". O termo "Instâncias" representa o número de instâncias recortadas da imagem original através do *Patch-SaliMap*. Foram avaliados o uso de: 5 instâncias utilizando somente o mapa de saliências referente à localização dos ácaros ("5 positivas"); e 3 instâncias utilizando os locais apontados pelo mapa e 3 apontados pelo seu oposto ("3 positivas e 3 negativas"), ou seja, 3 instâncias contendo alta probabilidade de conter ácaros e 3 contendo a menor probabilidade possível segundo a pontuação dos mapas de saliências. "Pré-treinado" indica que os pesos do *Bag Model* foram pré-treinados na CPB antes de inicializar o processo unificado. "N/A" significa que a opção não pode ser aplicada. Os valores destacados correspondem aos melhores resultados.

Método	Instâncias	Pré-treinado	Acur. (%)	F1 (%)
Original	5		**91,7** $\pm$**1,1**	**90,6** $\pm$**0,9**
Bag Model	N/A	N/A	82,3 $\pm$1,5	79,4 $\pm$1,8
Unificado	3 positivas e 3 negativas		79,2 $\pm$2,2	77,0 $\pm$1,9
Unificado	3 positivas e 3 negativas	•	79,0 $\pm$2,6	79,0 $\pm$2,6
Unificado	5 positivas		79,6 $\pm$3,4	77,3 $\pm$2,9
Unificado	5 positivas	•	78,3 $\pm$2,2	77,5 $\pm$2,9

para a versão unificada não são intuitivos, mas mostram que o treinamento conjunto do *Bag Model* e do *Instance Model* não é aconselhável. Foram avaliados diferentes tipos de criação de instâncias para verificar se somente o uso de áreas com alta probabilidade de conter ácaros estaria impactando os resultados negativamente ("5 positivas" e "3 positivas e 3 negativas"), entretanto, considerando os resultados, o uso de diferentes tipos de instâncias para versão unificada não altera drasticamente a efetividade dos modelos, chegando a uma diferença máxima de 0,4 pontos percentuais entre os pares de experimentos ("3 positivas e 3 negativas" e "5 positivas").

Bag Models pré-treinados na CPB, que podem oferecer localizações fracamente supervisionadas no início do treinamento, parecem afetar suavemente a versão unificada do *Attention-based MIL-Guided* ("Pré-treinado"). A variação dos resultados em relação ao pré-treinamento dos *Bag Models* não forneceu um impacto significativo na acurácia e medida F1, chegando apenas a uma diferença máxima de 2 pontos percentuais.

Os resultados da Tabela 6.11 mostram que o treinamento do *Instance Model* atrapalha o treinamento do *Bag Model* quando unificados. Independentemente das configurações avaliadas da versão unificada, todos os resultados de acurácia decaem ao menos 2,6 pontos percentuais relativamente aos resultados do *Bag Model*. Seria intuitivo que, ao se calcular automaticamente os pesos relativos ao agregador, eles recebessem valores baixos relacionados às instâncias para que não interfiram na predição final. Assim, as predições do *Bag Model* prevaleceriam e as predições finais estariam, ao menos, ao redor das predições do *Bag Model*. Porém, depois do treinamento, todas as predições decaíram e, assim, até mesmo o preditor do *Bag Model* foi impactado.

Resultados com pouco impacto foram encontrados no trabalho de Shen et al. [166], em que o treinamento unificado trouxe uma pequena melhora, entretanto, eles não obtiveram uma variação muito grande entre o resultado do modelo que utiliza as imagens inteiras (preditor global, denominação de Shen et al. [166], e *Bag Model*, para esta tese) e o modelo que usa as imagens recortadas. A grande variação entre os modelos do *Attention-based MIL-Guided* (*Bag*

Model e *Instance Model*) vem da forma como eles são treinados, utilizando etapas separadas. O trabalho de Chen et al. [37] corrobora com esta hipótese, pois usa um classificador baseado em duas etapas. Eles expõem uma diferença considerável entre o classificador de *patches* e o de imagens originais.

Os experimentos desta subseção ajudaram na resposta da questão Q3: *"é possível desenvolver uma arquitetura ponta a ponta para gerar mapas de ativação que selecionem múltiplas regiões de interesse para a classificação automática de pragas?"*.

6.2.8 Discussão dos Experimentos

Esta seção expôs diversos experimentos para determinar se o *Attention-based MIL-Guided*, juntamente com a proposta de mapas de ativação baseada em atenção, chamada *Two-WAM*, seria efetivo na classificação e localização de pragas em imagens obtidas sob condições naturais. Por isso, a maior parte dos experimentos simulou, com as bases CPB e NCPB, situações que são encontradas em campo, como a presença de ruídos nas imagens. A remoção desses ruídos melhorou os resultados para a tarefa de classificação binária usando o *Bag Model*, enquanto *patches* ruidosos foram essenciais para a tarefa de classificação usando o *Instance Model*. Na literatura, são comuns trabalhos com pragas que contenham RIs salientes [193, 197], assim o uso de ruído como aumentação pode influenciar positivamente seus treinamentos. No entanto, a presença de ruído resultou em modelos menos eficazes nos treinamentos com imagens contendo RIs proporcionalmente pequenas.

Os experimentos mostraram que o *Bag Model* do *Attention-based MIL-Guided* pôde destacar um número significativo de ácaros em uma imagem, mas, quando existia um número grande deles, o *Bag Model*, mais precisamente o método *Two-WAM*, não conseguiu destacar todos eles. Entretanto, a capacidade mostrada pelo *Two-WAM* é adequada para aplicações que estabeleçam contagens com base em limiares para aplicações de insumos, como, por exemplo, ausência total (nenhum ácaro), número baixo (menor que um limiar) e número alto (acima do limiar) (Seção 2.2). O *Two-WAM* melhorou a classificação de ácaros enquanto não permitia diminuir o desempenho de classificação em pragas com regiões corporais grandes, como as pragas da IP102 1.0 e 1.1. O *Two-WAM*, diferentemente do Grad-CAM, pode ser implantado conjuntamente com o *Attention-based MIL-Guided* em dispositivo que suportam modelos simples de redes neurais profundas, ou seja, é apropriado para aplicações de campo. Suas regiões se adaptaram melhor aos corpos dos ácaros e insetos do que outros métodos fracamente supervisionados. Dessa forma, o *Two-WAM* pôde localizá-los com treinamento utilizando somente rótulos de classificação.

O *Attention-based MIL-Guided*, como processo em etapas, é propício para ser aplicado na classificação binária de regiões de interesse muito pequenas, entretanto, não é aconselhável juntar seu *Bag Model* e *Instance Model* em um treinamento único para geração de uma arquitetura ponta a ponta. Seu *Bag Model*, além de um modelo de classificação, é uma ferramenta que infere localizações fracamente supervisionadas tanto para ácaros quanto para insetos salientes e seu *Instance Model* melhorou os resultados de classificação. Sugestões para melhorar e modificar a forma como o *Bag Model* é treinado são apresentadas, na Seção 6.3, pois foram encontrados problemas com sua classificação multirrótulos.

6.3 Experimentos Relacionados à Adaptação de Domínio Não Supervisionada e Remoção dos Rótulos da Classe Negativa

Esta seção descreve os experimentos relativos à tarefa de classificação multirrótulos da base de dados CPB, o que inclui a análise dos métodos fracamente supervisionados propostos, *Attention-based MIL-Guided* e *MIL-Guided*. A Subseção 6.3.1 apresenta novas configurações para os experimentos. Na Subseção 6.2.2, a avaliação para os métodos apresentados nesta tese é discutida para a tarefa de classificação multirrótulos da CPB. Sua análise mostrou que o uso do *Instance Model* não é adequado para inferências multirrótulos. Então, na Subseção 6.3.3, são apresentados experimentos e análises com rótulos para melhorar a forma como os classificadores dos *Bag Models* do *Attention-based MIL-Guided* e *MIL-Guided* podem ser treinados. Na Subseção 6.3.4, foram descritos os resultados da adaptação de domínio não supervisionada aplicada ao *Bag Model* do *Attention-based MIL-Guided* na tarefa multirrótulos. Finalizando a seção e o capítulo, são apresentadas as discussões sobre a tarefa de classificação multirrótulos da CPB.

6.3.1 Configuração dos Experimentos

Nesta seção, todas as melhores configurações apontadas por experimentos anteriores foram utilizadas, como: o uso da EfficientNet-B0 [173] pré-treinada na ImageNet para extrair características para o *MIL-Guided* e *Attention-based MIL-Guided*; a adaptação dos *Bag Models* no conjunto de treinamento da base NCPB e avaliado nos conjuntos da CPB; e a adaptação dos *Instance Models* treinados e avaliados com instâncias recortadas dos conjuntos da CPB. Como há mais de um tipo de ácaro presente nas imagens para classificação multirrótulos, a função de ativação *sigmoid* foi utilizada na camada de geração das predições. As configurações dos hiperparâmetros foram aplicadas como nas seções anteriores (Subseções 6.1.1 e 6.2.1), entretanto, quando modificadas, as novas configurações foram explicitamente citadas nas subseções. Uma GPU Nvidia Quadro RTX 8000 foi utilizada em todos os treinamentos. O maior tamanho de *batch* que a GPU suportou foi adotado para todos os experimentos (8 imagens da CPB original).

6.3.2 *MIL-Guided* e *Attention-based MIL-Guided* na Classificação Multirrótulos da CPB

O objetivo desta subseção é avaliar o uso do *Attention-based MIL-Guided* e *MIL-Guided* em um contexto multirrótulos da base de dados CPB. Foi utilizado o conjunto de *validação* da CPB para não sobreajustar os hiperparâmetros e os resultados dos modelos são apresentados na Tabela 6.12. O processo de treinamento é parecido com o avaliado nas seções que consideraram a classificação binária, com exceção do número de *patches* utilizado, que foi avaliado para $k = 1, 2, 3, 4, 5$. Também, foram analisados o uso de *Bag Models* treinados em duas tarefas diferentes (binária e multirrótulos) para geração dos *patches* para os *Instance Models* multirrótulos. Foi denominado *Bag Model* binário o modelo treinado para a tarefa de classificação binária da CPB. Ele produz *patches* a partir do *Two-WAM* ou Grad-CAM. Assim, foi denominado *Bag Model* multirrótulos o modelo treinado a partir da tarefa multirrótulos da CPB e que

também produz *patches* a partir dos mesmos métodos.

Os melhores resultados foram alcançados sem o uso do *Instance Model* para os experimentos da Tabela 6.12. O *Bag Model* do *Attention-based MIL-Guided* foi o melhor classificador multirrótulos da CPB com 59,3% de acurácia e 65,0% de medida F1. O *Bag Model* do *MIL-Guided* foi o segundo melhor classificador com 54,8% de acurácia e 62,4% de medida F1. Todos os resultados empregando o processo completo do *Attention-based MIL-Guided* e do *MIL-Guided* ficaram ao menos 10 pontos percentuais abaixo do melhor resultado apresentado.

Ao comparar os resultados da Tabela 6.12, que reporta o processo completo e considera o mesmo número de instâncias para cada classificador, pode-se ranquear, do melhor para o pior, considerando a acurácia de classificação: o *Attention-based MIL-Guided* usando o *Bag Model* multirrótulos ("*Attention-based MIL-Guided Bag Model* Multirrótulos + *Instance Model*"), como melhor resultado entre eles; o *Attention-based MIL-Guided* com *Bag Model* binário ("*Attention-based MIL-Guided Bag Model* Binário + *Instance Model*"), como segundo melhor resultado; o *MIL-Guided* com *Bag Model* binário ("*MIL-Guided Bag Model* Binário + *Instance Model*"), em terceiro lugar; e o *MIL-Guided* com *Bag Model* multirrótulos ("*MIL-Guided Bag Model* Binário + *Instance Model*"), como a pior opção de classificação multirrótulos do processo unificado.

Os experimentos da Tabela 6.12 mostram que os processos inteiros do *Attention-based MIL-Guided* e *MIL-Guided*, como originalmente propostos, não são adequados para a tarefa multirrótulos da base de dados CPB, ou precisam ser adaptados. Um possível motivo é o desbalanceamento entre o número de exemplos das classes da CPB. No caso binário, os processos aumentavam o número de exemplos para treinamento ao mesmo tempo que simulavam o efeito de aumentar o tamanho dos ácaros proporcionalmente ao tamanho das novas imagens (instâncias). No caso das imagens multirrótulos, (i) não se tem o controle de quais as classes de ácaros estarão realmente nas instâncias, pois cada instância assume o mesmo rótulo da imagem original, e (ii) não se tem o controle da existência dos ácaros nas instâncias, que era um problema menor com a localização binária. Então, obter uma situação favorável e adequada para o treinamento do *Instance Model* é improvável, diferentemente da tarefa binária. Isso ocorre porque os métodos de múltiplas instâncias foram criados para tarefas de classificação binária [142] e é difícil adaptar sua premissa básica, apesar de alguns trabalhos publicados (Seção 3.1).

É comum observar problemas de localização do Grad-CAM quando se erra a classe do indivíduo (quarta coluna da Figura 6.6e) e, por isso, os modelos do *MIL-Guided* na Tabela 6.12 tiveram resultados piores do que o *Attention-based MIL-Guided*. Também, por esse motivo, o processo usando o *Bag Model* do *MIL-Guided* binário teve melhores resultados do que o multirrótulos, pois é mais fácil obter a localização se a única classe a ser apontada é a positiva. O *Two-WAM* é agnóstico a classes e aprende as localizações independentemente das categorizações, entretanto, por não fazer essa distinção, ele não sabe apontar uma localização contendo uma classe específica para gerar as instâncias por tipo de ácaro. Assim, o método não produz um número balanceado de *patches* e um bom desempenho para ajudar no treinamento do *Instance Model*.

A Figura 6.8 exibe as matrizes de confusão para os *Bag Models* multirrótulo e binário do *Attention-based MIL-Guided*, além de mostrar a matriz de confusão do *Instance Model* do *Attention-based MIL-Guided* treinado com imagens recortadas a partir do *Bag Model* multirrótulos (as imagens contendo mais de um rótulo foram removidas da avaliação).

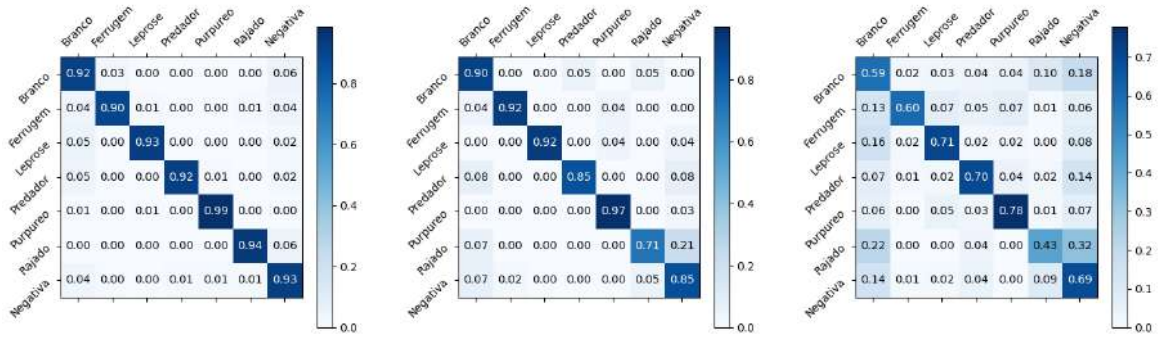
É notável a diferença entre os valores de acurácia em relação às imagens da classe negativa

Tabela 6.12: Propostas fracamente supervisionadas avaliadas para a tarefa multirrótulos no conjunto de *validação* da CPB. O *Attention-based MIL-Guided* e o *MIL-Guided* foram avaliadas utilizando o processo completo ou parcial (somente o *Bag Model*), com a variação do uso do *Bag Model* treinado como uma tarefa binária ou multirrótulos quando o processo é apresentado como completo (*Bag Model* + *Instance Model*). "Instâncias" significa a quantidade de instâncias usadas no treinamento do *Instance Model*. "–" significa que a opção não pode ser aplicada. Todas as opções que exibem *Instance Models* têm saídas multirrótulos pelo método de avaliação ponderada, ou seja, o classificador final de todas as opções é multirrótulos. Os valores destacados correspondem aos melhores resultados.

Método	Instâncias	Acur. (%)	F1 (%)
Attention-based MIL-Guided Bag Model Multirrótulos	–	**59,3** ±1,4	**65,0** ±2,1
MIL-Guided Bag Model Multirrótulos	–	54,8 ±1,1	62,4 ±0,6
Attention-based MIL-Guided Bag Model Multirrótulos + *Instance Model*	1	47,1 ±1,4	50,6 ±1,5
	2	49,2 ±1,5	51,6 ±1,8
	3	49,1 ±1,5	50,8 ±2,4
	4	48,0 ±1,3	49,6 ±2,7
	5	46,6 ±1,6	48,2 ±2,8
Attention-based MIL-Guided Bag Model Binário + *Instance Model*	1	45,4 ±1,9	49,8 ±2,6
	2	46,0 ±2,0	49,6 ±3,1
	3	47,4 ±2,2	49,2 ±4,3
	4	46,0 ±1,9	48,0 ±4,3
	5	44,5 ±2,3	46,2 ±4,6
MIL-Guided Bag Model Binário + *Instance Model*	1	39,7 ±1,8	44,5 ±1,9
	2	41,4 ±1,6	45,3 ±1,5
	3	40,0 ±1,5	42,8 ±2,1
	4	38,8 ±1,3	43,5 ±3,3
	5	37,2 ±1,1	39,3 ±2,1
MIL-Guided Bag Model Multirrótulos + *Instance Model*	1	29,4 ±1,3	35,6 ±2,0
	2	30,4 ±1,6	36,8 ±1,5
	3	30,0 ±1,3	35,6 ±2,1
	4	29,4 ±1,3	34,6 ±2,5
	5	30,5 ±1,5	36,8 ±2,4

da Figuras 6.8c (69,0%), Figuras 6.8d (73,0%) e Figuras 6.8e (82,0%). Há uma diminuição de 4 pontos percentuais ao se comparar as matrizes de confusão para o problema binário (Figuras 6.8d) e multirrótulos (Figuras 6.8c e 6.8e) das imagens originas da CPB. Essa diminuição afeta o *Bag Model* quando este é treinado utilizando multirrótulos. Isso é digno de atenção, pois as imagens da classe negativa e seus rótulos são exatamente os mesmos para o treinamento das duas tarefas. Esta situação é discutida na Seção 6.3.3, a qual mostra que a presença dos rótulos da classe negativa no treinamento traz um pior resultado para a tarefa de classificação multirrótulos da CPB. Na comparação entre o *Bag Model* multirrótulos e *Instance Model* multirrótulos (Figuras 6.8c), pode-se notar que os resultados da classe negativa melhoraram, mas os resultados das outras classes decaíram significativamente.

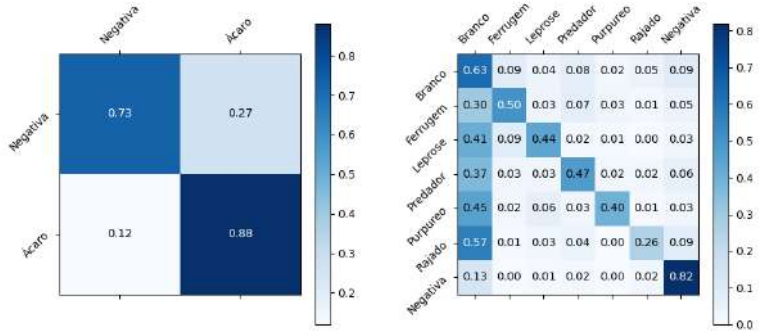
Outro ponto importante a se analisar sobre as matrizes de confusão da Figura 6.8 é o pequeno impacto do sobreajuste (*overfitting*) no treinamento do *Bag Model*, ou seja, as pontuações nas



(a) *Bag Model* para o conjunto de treinamento

(b) *Bag Model* para o conjunto de pseudovalidação do treinamento

(c) *Bag Model* para o conjunto de validação



(d) *Bag Model* para o conjunto de validação binário

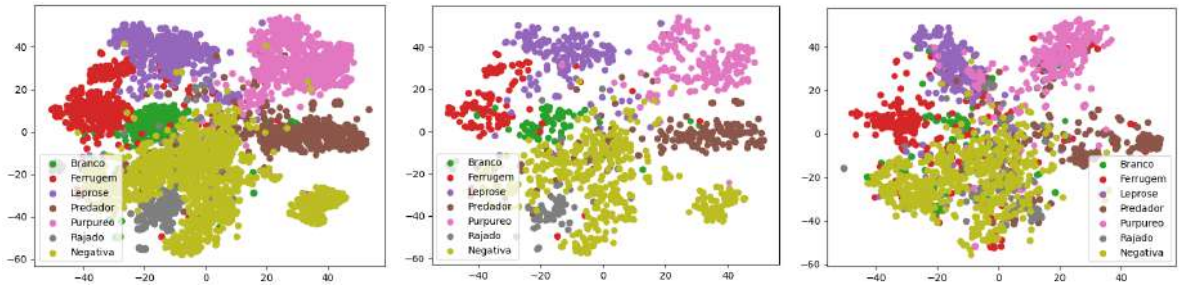
(e) *Instance Model* para o conjunto de validação

Figura 6.8: Matrizes de confusão para o *Attention-based MIL-Guided* para: (a) o conjunto de treinamento efetivamente usado na adaptação do *Bag Model*; (b) o conjunto particionado do treinamento que age como pseudovalidação, enquanto se treina o *Bag Model*; (c) o conjunto de validação da CPB relativamente ao *Bag Model*; (d) o conjunto de validação relativamente ao *Bag Model* treinado para a tarefa binária da CPB; e (e) o conjunto de validação relativamente ao *Instance Model* considerando as instâncias da CPB.

Figuras 6.8a e 6.8b têm valores não muito distantes, entretanto, a Figura 6.8c mostra uma queda de até 30 pontos percentuais para os ácaros das classes branco e ferrugem. Como o processo é efetuado utilizando somente o conjunto de treinamento particionado em dois ("treinamento" e "pseudovalidação"), a predição em um terceiro conjunto deveria trazer resultados similares ou ligeiramente piores. Isso indica uma mudança na distribuição dos dados para os conjuntos de treinamento e validação da CPB e, com isso, a possibilidade do uso de métodos de adaptação de domínio não supervisionado, como apresentado na Seção 6.3.4. A Figura 6.9, cujas partes foram geradas por meio da técnica t-SNE [180], ilustra essa situação.

Algumas classes, como a de ácaros rajados, têm uma generalização ruim em relação a outros tipos de ácaros, por exemplo, ácaros da leprose, predadores e purpúreos. Esses resultados podem ser atribuídos à quantidade de exemplos da classe em questão. A classe de ácaros rajados tem o menor número de exemplos da base (696 imagens), enquanto as outras três classes citadas contêm mais de mil imagens cada (Capítulo 4). Além disso, ácaros brancos são muito confundidos com rajados e ambos apresentam um número significativo de falsos negativos, ou

seja, sua presença não é detectada pelo classificador e eles são apontados como classe negativa. O conjunto contendo os exemplares de ácaros brancos é o segundo menor e, do mesmo modo que os ácaros rajados, seu conjunto não é maior do que mil exemplares (806 imagens). Na Figura 6.9c, pode-se observar uma sobreposição dos conjuntos de ácaros brancos (cor verde), rajados (cor cinza) e classe negativa (cor amarela). Essa sobreposição é menor nas Figuras 6.9a e 6.9b. Os ácaros rajados não são considerados pragas, ou seja, sua presença não ocasiona danos para a produção, no entanto, eles são muito confundidos com ácaros brancos em suas fases iniciais de crescimento. Essa informação merece ser ressaltada, uma vez que os resultados dos modelos multirrótulos também trouxeram esse erro comum aos especialistas.



(a) t-SNE do *Bag Model* para o conjunto de treinamento

(b) t-SNE do *Bag Model* para o conjunto de pseudovalidação do treinamento

(c) t-SNE do *Bag Model* para o conjunto de validação

Figura 6.9: Distribuição segundo a técnica t-SNE [180] para os conjuntos da CPB, segundo o *Bag Model* do *Attention-based MIL-Guided* para: (a) o conjunto de treinamento efetivamente usado na adaptação do *Bag Model*; (b) o conjunto particionado do treinamento que age como pseudovalidação, enquanto se treina o *Bag Model*; (c) o conjunto de validação da CPB relativamente ao *Bag Model*.

Os experimentos desta subseção ajudaram a responder a questão de pesquisa Q1 relativamente à tarefa multirrótulos da CPB: *"métodos de aprendizado fracamente supervisionados por múltiplas instâncias são eficazes para classificação de pragas pequenas da citricultura?"*

6.3.3 Experimentos com Rótulos das Classes Negativa e Positiva para a Classificação Multirrótulos da CPB

Os experimentos relatados nesta subseção têm o objetivo de avaliar a efetividade do *Bag Model* utilizando diferentes tipos de classificadores multirrótulos para os processos do *Attention-based MIL-Guided* e *MIL-Guided*. Em suma, foram utilizados classificadores totalmente conectados e totalmente convolucionais (Subseção 2.4.1) ao final das arquiteturas avaliadas e foi analisado o impacto da utilização de um rótulo para a classe negativa (que especifica um rótulo para inexistência de ácaros) e um rótulo para a classe positiva (especificando um rótulo para existência de ácaros). Foram convencionadas as palavras "negativa" para se referir ao rótulo da classe negativa já existente na CPB e "positiva" para o rótulo que especifica a existência de algum tipo de ácaro nas imagens, ou seja, o rótulo para classe positiva quando o treinamento binário ocorre. Os outros seis rótulos referentes às classes de ácaros pertencentes à CPB foram utilizadas sem alterações (ácaro purpúreo, ácaro predador, ácaro da ferrugem, ácaro da leprose, ácaro branco e ácaro rajado). As imagens da classe negativa foram identificadas pela não existência

de um índice que apontasse a presença de uma das seis classes de ácaros para uma imagem (Seção 2.3).

A Tabela 6.13 apresenta a avaliação de dezesseis experimentos relacionados com o poder de classificação de arquiteturas treinadas em imagens sem cortes, ou seja, foram utilizados os *Bag Models* para os métodos fracamente supervisionados propostos. Não foi utilizado o balanceamento de classes no treinamento de cada *batch*, pois seu uso requer um tempo de treinamento mais prolongado. No entanto, a Tabela 6.14 faz uma comparação dos melhores resultados da Tabela 6.13 com os resultados multirrótulos previamente estabelecidos (Subseção 6.3.2) considerando o balanceamento do *batch*.

Tabela 6.13: Comparação do uso do rótulo para a classe positiva e classe negativa no treinamento do *Bag Model* para o *MIL-Guided* e *Attention-based MIL-Guided*. Foram aplicados os classificadores totalmente conectados ("tot. conectado") e totalmente convolucionais ("tot. convolucional") (descritos na Subseção 2.4.1). As variações no número de classes são: o uso padrão dos 7 rótulos de classes da CPB ("6 classes + negativa"); a adição do rótulo da classe positiva ("6 classes + negativa + positiva"); a remoção do rótulo da classe negativa das classes da CPB ("6 classes"); e a troca do rótulo da classe negativa pelo rótulo da classe positiva ("6 classes + positiva"). Os valores destacados correspondem aos melhores resultados.

Método	Classes	Classificador	Acur. (%)	F1 (%)
MIL-Guided	6 classes	tot. conectado	58,3 $\pm_{2,2}$	55,2 $\pm_{2,4}$
Attention-based MIL-Guided	6 classes	tot. conectado	62,8 $\pm_{1,5}$	61,8 $\pm_{1,5}$
MIL-Guided	6 classes	tot. convolucional	55,8 $\pm_{1,6}$	54,2 $\pm_{1,8}$
Attention-based MIL-Guided	6 classes	tot. convolucional	60,0 $\pm_{1,8}$	59,4 $\pm_{1,7}$
MIL-Guided	6 classes + negativa	tot. conectado	53,5 $\pm_{3,3}$	57,0 $\pm_{1,9}$
Attention-based MIL-Guided	6 classes + negativa	tot. conectado	52,7 $\pm_{7,0}$	56,4 $\pm_{12,0}$
MIL-Guided	6 classes + negativa	tot. convolucional	53,5 $\pm_{0,8}$	56,6 $\pm_{1,1}$
Attention-based MIL-Guided	6 classes + negativa	tot. convolucional	58,5 $\pm_{1,9}$	**64,5** $\pm_{1,3}$
MIL-Guided	6 classes + positiva	tot. conectado	60,6 $\pm_{1,5}$	57,8 $\pm_{1,7}$
Attention-based MIL-Guided	6 classes + positiva	tot. conectado	64,8 $\pm_{0,3}$	61,8 $\pm_{2,1}$
MIL-Guided	6 classes + positiva	tot. convolucional	60,0 $\pm_{0,4}$	57,0 $\pm_{1,4}$
Attention-based MIL-Guided	6 classes + positiva	tot. convolucional	**65,1** $\pm_{0,9}$	63,0 $\pm_{1,4}$
MIL-Guided	6 classes + negativa + positiva	tot. conectado	43,6 $\pm_{1,7}$	47,0 $\pm_{1,4}$
Attention-based MIL-Guided	6 classes + negativa + positiva	tot. conectado	58,6 $\pm_{4,5}$	62,4 $\pm_{2,3}$
MIL-Guided	6 classes + negativa + positiva	tot. convolucional	53,1 $\pm_{1,5}$	55,4 $\pm_{1,4}$
Attention-based MIL-Guided	6 classes + negativa + positiva	tot. convolucional	59,8 $\pm_{2,1}$	63,8 $\pm_{2,1}$

Nas opções de configurações da Tabela 6.13, são exibidos: (i) os métodos "*MIL-Guided*"e "*Attention-based MIL-Guided*"; (ii) o número de rótulos de classes empregadas com as opções originais da CPB ("6 classes + negativa"), remoção do rótulo da classe negativa ("6 classes"), remoção do rótulo da classe negativa com adição do rótulo da classe positiva ("6 classes + positiva") e classes originais mais a adição do rótulo da classe positiva ("6 classes + negativa + positiva"); (iii) o tipo de preditor, cujos valores são classificador totalmente conectado ("tot. conectado") e classificador totalmente convolucional ("tot. convolcional") (Subseção 2.4.1).

Dentre os experimentos da Tabela 6.13 que utilizam o número pré-estabelecido de rótulos para as classes da CPB ("6 classes + negativa"), a configuração do uso do *Attention-based MIL-Guided* tot. convolucional obteve o melhor resultado, com 58,5% de acurácia e 64,5% de

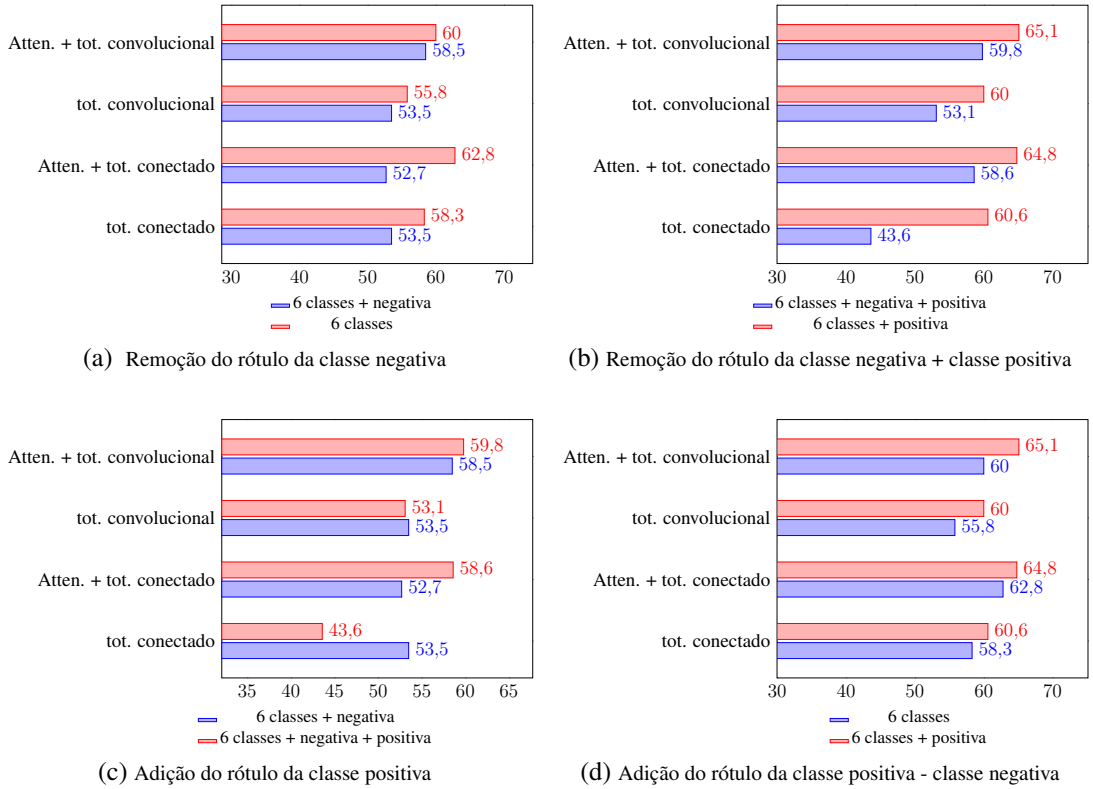
medida F1, inclusive esse é o melhor valor de medida F1 da Tabela 6.13. Em relação à adição do rótulo da classe positiva nas classes da CPB ("6 classes + negativa + positiva"), o melhor resultado também foi o uso do *Attention-based MIL-Guided* tot. convolucional, com 59,8% de acurácia e 63,8% de medida F1. O *Attention-based MIL-Guided* tot. conectado obteve o melhor resultado quando o rótulo da classe negativa foi removido dos rótulos das classes originais da CPB ("6 classes"), com 62,8% de acurácia e 61,8% de medida F1. Por fim, o *Attention-based MIL-Guided* tot. convolucional obteve o melhor resultado de acurácia da tabela ("6 classes + positiva"), com 65,1% de acurácia e 63,0% de medida F1.

Para melhor avaliar os resultados em relação ao uso dos rótulos da classe positiva e da classe negativa, foram gerados os gráficos da Figura 6.10. Nota-se claramente nas Figuras 6.10a e 6.10b que a remoção do rótulo da classe negativa contribuiu com o treinamento dos modelos independentemente do uso do rótulo da classe positiva em seu treinamento. A Figura 6.10c não confirma se a adição do rótulo da classe positiva funciona na presença da classe negativa. Por outro lado, a Figura 6.10d mostra uma melhora em todos os cenários quando o rótulo da classe positiva é adicionado ao treinamento de modelos que removeram o rótulo da classe negativa. Esses resultados mostram que o uso do rótulo da classe negativa no treinamento multirrótulos diminui a efetividade dos modelos. Possivelmente, as características relacionadas à classe negativa, que são derivadas e procuradas durante o *fine tuning* dos parâmetros, atrapalham o treinamento de modelos quando são consideradas RIs muito pequenas para classificação, como os ácaros presentes na CPB. Ao mesmo tempo, o uso de um rótulo para a classe positiva propicia a busca de características na presença de ácaros.

A primeira e segunda melhores opções de acurácia (*Attention-based MIL-Guided* com 6 classes, mais o rótulo da classe positiva totalmente conectado e totalmente convolucional) e a melhor opção de medida F1 da Tabela 6.13, foram retreinadas para uma nova comparação considerando o balanceamento dos *batches* de treinamento. Os resultados são apresentados na Tabela 6.14.

Os valores de acurácia da Tabela 6.14 evidenciam um empate entre os dois melhores modelos quando o *batch* é balanceado (os dois modelos com 6 classes, mais o rótulo da classe positiva). A opção com maior valor chegou a 66,5% de acurácia ("6 classes + positiva", conjuntamente com "tot. conectado") e a segunda com maior valor chegou a 66,2% de acurácia ("6 classes + positiva", conjuntamente com "tot. convolucional"). Possivelmente, o uso de um classificador totalmente convolucional é mais efetivo relativamente à medida F1, mas o uso do balanceamento, além de melhorar a acurácia, também tem efeito na medida F1. No caso da opção "6 classes + negativa" treinada com o classificador totalmente convolucional e balanceamento, o resultado da medida F1 aumentou relativamente ao não uso do balanceamento. Esse bom resultado para a medida F1 utilizando o rótulo da classe negativa ("6 classes + negativa") pode ser explicado devido ao fato de que medida F1 não considera a taxa de verdadeiros negativos, cuja influência está diretamente ligada à classe negativa.

Todos os valores da medida F1 para as três últimas linhas da Tabela 6.14 estão muito próximos ao se considerar sua média e seu desvio padrão (casos de *batch* balanceado). Então, foi considerado um empate entre esses três resultados. O melhor valor entre eles é de 66,4% de medida F1, obtido pela opção sem o rótulo da classe negativa e com a adição do rótulo da classe positiva totalmente convolucional ("6 classes + positiva", conjuntamente com "tot. convolucional"). Levando-se em conta a acurácia, os modelos que obtiveram melhores resultados foram as opções com adição da classe positiva e remoção da classe negativa independentemente do tipo



(a) Remoção do rótulo da classe negativa

(b) Remoção do rótulo da classe negativa + classe positiva

(c) Adição do rótulo da classe positiva

(d) Adição do rótulo da classe positiva - classe negativa

Figura 6.10: Impacto do uso dos rótulos da classe negativa e classe positiva: (a) remoção do rótulo da classe negativa no treinamento do *Bag Model*; (b) remoção do rótulo da classe negativa no treinamento do *Bag Model* com adição do rótulo da classe positiva; (c) adição do rótulo da classe positiva no treinamento do *Bag Model*; (d) adição do rótulo da classe positiva no treinamento do *Bag Model* sem o uso do rótulo da classe negativa. "tot. conectado" modelos treinados utilizando um classificador totalmente conectado; "tot. convolucional" modelos treinados utilizando um classificador totalmente convolucional; e "*Atten.*" *Bag Models* do método *Attention-based MIL-Guided*. Os resultados representam a acurácia da Tabela 6.13.

Tabela 6.14: Avaliação do balanceamento do *batch* para as melhores opções do *Attention-based MIL-Guided* (melhor arquitetura sem o balanceamento do *batch*). As variações no número de rótulos de classes são: o uso dos rótulos para cada classe da CPB ("6 classes + negativa"); e a troca do rótulo da classe negativa pelo rótulo da classe positiva no treinamento ("6 classes + positiva"). As diferentes estratégias para os classificadores são: um classificador totalmente conectado ("tot. conectado") e um classificador para uma arquitetura totalmente convolucional ("tot. convolucional'). Os valores destacados correspondem aos melhores resultados.

Batch	Classes	Classificador	Acur. (%)	F1 (%)
Não balanceado	6 classes + negativa	tot. conectado	52,7 $\pm_{7,0}$	56,4 $\pm_{12,0}$
Não balanceado	6 classes + negativa	tot. convolucional	58,5 $\pm_{1,9}$	64,5 $\pm_{1,3}$
Não balanceado	6 classes + positiva	tot. conectado	64,8 $\pm_{0,3}$	61,8 $\pm_{2,1}$
Não balanceado	6 classes + positiva	tot. convolucional	65,1 $\pm_{0,9}$	63,0 $\pm_{1,4}$
Balanceado	6 classes + negativa	tot. conectado	59,3 $\pm_{1,4}$	65,0 $\pm_{2,1}$
Balanceado	6 classes + negativa	tot. convolucional	60,9 $\pm_{2,3}$	66,2 $\pm_{1,8}$
Balanceado	6 classes + positiva	tot. conectado	**66,5** $\pm_{\mathbf{0,6}}$	65,8 $\pm_{1,0}$
Balanceado	6 classes + positiva	tot. convolucional	66,2 $\pm_{0,8}$	**66,4** $\pm_{\mathbf{0,5}}$

de classificador (“6 classes + positiva”).

6.3.4 Adaptação de Domínio Não Supervisionada para a Classificação Multirrótulos da CPB

Esta subseção apresenta a avaliação da técnica de adaptação de domínio não supervisionada (Seção 5.3) para a tarefa de classificação multirrótulos da base CPB. O conjunto de validação da base CPB foi utilizado para reportar os resultados na Tabela 6.15. A melhor opção foi escolhida para reportar o resultado final do trabalho no conjunto de teste. Para a avaliação, os melhores resultados multirrótulos conseguidos anteriormente na CPB foram utilizados, ou seja, o treinamento do *Bag Model* do *Attention-based MIL-Guided* utilizando o rótulo da classe positiva no lugar do rótulo da classe negativa para classificadores totalmente conectados e totalmente convolucionais (Seção 6.3.3).

A Tabela 6.15 compara os resultados com e sem uso de adaptação de domínio não supervisionada (“ADNS”) no conjunto de *validação* da CPB. Os melhores resultados alcançados são de 70,7% de acurácia e 69,5% de medida F1. Esses resultados foram obtidos utilizando a adaptação de domínio não supervisionada. A abordagem final desta seção, exposta nesta subseção, obteve uma melhora de 10,4 pontos percentuais de acurácia e 4,5 pontos percentuais de medida F1 em relação ao *Bag Model* do *Attention-based MIL-Guided* com melhor resultado, cujo treinamento ocorreu sem a remoção do rótulo da classe negativa, adição do rótulo da classe positiva e adaptação de domínio não supervisionada, para a tarefa binária da CPB (valores de referência: 59,3% de acurácia e 65,0% de medida F1 da Tabela 6.14 na Subseção 6.3.3).

A Tabela 6.15 apresenta os melhores valores para as opções com uso de adaptação de domínio não supervisionada (“ADNS”), independentemente do tipo de classificador. Então, o melhor resultado para o conjunto de validação da CPB é a opção de *Bag Model* do *Attention-based MIL-Guided* com o rótulo da classe positiva no lugar do rótulo da classe negativa, o uso da adaptação de domínio não supervisionada e o emprego de um classificador totalmente conectado (o classificador totalmente conectado sempre foi utilizado nas seções anteriores). O resultado para o conjunto de *teste* da CPB do melhor modelo treinado para a tarefa de classificação multirrótulos é de 69,1% de acurácia e 64,0% de medida F1. Esses valores estabelecem o melhor resultado para classificação de diferentes tipos de ácaros da CPB.

Tabela 6.15: Avaliação da influência da adaptação de domínio não supervisionada (“ADNS”) na classificação multirrótulos do *Bag Model* do *Attention-based MIL-Guided* com o uso do rótulo da classe positiva no lugar do rótulo da classe negativa. As diferentes estratégias para os classificadores são: um classificador totalmente conectado e um classificador para uma arquitetura totalmente convolucional. Os valores destacados correspondem aos melhores resultados.

Classificador	ADNS	Acur. (%)	F1 (%)
tot. conectado		66,5 $\pm$0,6	65,8 $\pm$1,0
tot. convolucional		66,2 $\pm$0,8	66,4 $\pm$0,5
tot. conectado	•	**70,1** $\pm$**0,7**	**69,0** $\pm$**0,7**
tot. convolucional	•	68,8 $\pm$1,8	68,5 $\pm$1,3

Esta subseção melhora o estado da arte para métodos fracamente supervisionados na tarefa

de classificação multirrótulos da CPB, ajudando com o objetivo O4 para tarefa multirrótulos: *"melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas"*.

6.3.5 Discussão dos Experimentos

Os experimentos desta seção descreveram uma situação comum nos trabalhos de automatização do MIP, em que estações do ano e meses diferentes podem mostrar fases de crescimento distintas dos ácaros. A base de dados CPB foi organizada por classe e por data da coleta, propiciando que ácaros em fases de crescimento distintas estivessem em conjuntos diferentes. Este é um problema comum em campo e a CPB foi desenvolvida para experienciar esses tipos de problemas. Por isso, as distribuições dos três conjuntos pertencentes à base, treinamento, validação e teste, parecem não ser próximas e as características intraclasses dos ácaros parecem variar muito de um conjunto para o outro.

Pode-se dizer que a tarefa de classificação multirrótulos é muito mais complexa do que a tarefa binária e a solução multirrótulos encontrada desvia completamente da solução anterior (Seção 6.3.2). Outro resultado não intuitivo é a necessidade de remover o rótulo da classe negativa nos treinamento para a tarefa multirrótulos (Seção 6.3.3). Provavelmente, as características de regiões que propiciem a classificação das imagens na categoria negativa se confundam com as características das pequenas regiões onde estão os ácaros.

Finalmente, mostrou-se que as distribuições referentes aos três conjuntos citados são diferentes, pois foram gerados melhores resultados ao se treinar os modelos utilizando uma técnica de adaptação de domínio, ou seja, conseguiu-se melhorar o alinhamento entre as distribuições dos conjuntos, entretanto, o resultado do conjunto de teste ainda pode ser melhorado considerando-se a classificação multirrótulos.

A técnica de adaptação de domínio não supervisionada desta seção foi utilizada por ser uma das mais simples e de fácil adaptação para o *Bag Model* do *Attention-based MIL-Guided*. Estudos adicionais com outras técnicas devem ser efetuados, entretanto, a partir dos resultados observados, a adaptação de domínio mostrou-se promissora para melhorar os resultados da tarefa de classificação multirrótulos da base CPB.

Capítulo 7

Conclusões

Esta tese de doutorado apresentou estratégias para automatizar o processo de classificação de pragas e vetores de doenças da lavoura de citros descrito no Manejo Integrado de Pragas (MIP). Ela teve como foco principal os ácaros, pragas invisíveis a olho nu, cujas regiões de interesse nas imagens possuem dimensões bem pequenas.

A Seção 7.1 deste capítulo final apresenta as motivações e os melhores resultados obtidos neste trabalho. A Seção 7.2 recapitula as contribuições fornecidas por esta tese. A Seção 7.3 responde, baseado nos resultados alcançados, as questões de pesquisa formuladas nesta tese que guiaram o trabalho de pesquisa. A Seção 7.4 descreve como os métodos apresentados poderiam ser aplicados em campo. A Seção 7.5 apresenta os desafios a se enfrentar no futuro e os pontos que requerem mais estudos para que a automatização de pragas esteja presente em campo. A Seção 7.6 relata as considerações finais desta tese.

7.1 Motivações e Resultados

A motivação inicial desta tese surgiu da convivência cotidiana do aluno de doutorado com a lavoura (a residência onde nasceu e onde seus pais moram está localizada no interior da fazenda onde as imagens foram coletadas) e dos problemas existentes que foram verificados em campo para a classificação e controle de pragas e vetores de doenças (Subseção 4.1.1). A automatização do MIP permitirá agilizar o trabalho em campo e aumentar o grau de robustez do manejo, sendo menos propenso a erros. Além da área de agricultura, o mercado consumidor será beneficiado com a melhor qualidade de frutas e subprodutos cítricos e o meio ambiente pode deixar de receber cargas excessivas de insumos.

Métodos fracamente supervisionados baseados em redes neurais profundas foram desenvolvidos para reduzir a necessidade de rotulação das imagens (Capítulo 5) e o número de visitas a campo, reduzindo, por consequência, a necessidade do tempo de especialistas e custo inerente a eles. Para demonstrar a eficácia dessas técnicas, resultados na base IP102 (1.0 e 1.1) e na CPB foram alcançados, chegando a competir com o estado da arte para a IP102, mas sempre com menor número de parâmetros (60,7% de acurácia e 59,6% de medida F1 para a IP102 1.0, 69,5% de acurácia e 69,0% de medida F1 para a IP102 1.1 e 4,1 milhões de parâmetros, Subseção 6.1.3) e ao estado da arte na base CPB em sua tarefa binária (92,4% de acurácia e medida F1 de 91,8% com 8,1 milhões de parâmetros, Subseção 6.2.3).

Resultados apresentados para a tarefa de localização da base IP102 1.1 mostraram o desem-

penho dos métodos fracamente supervisionados propostos para a localização de pragas. Esses resultados foram promissores (AP de 19,1% e IoU de 47,3% com 4,1 milhões de parâmetros, Subseção 6.2.5) e chegaram muito próximos ao conjunto de resultados dos métodos totalmente supervisionados (uma diferença de 1 a 9 pontos percentuais de AP).

Em relação à tarefa de classificação multirrótulos da CPB, o melhor resultado foi obtido por meio da abordagem proposta nesta tese de mapas de ativação baseados em atenção, a substituição do rótulo da classe negativa pelo rótulo da classe positiva, o treinamento com um método de adaptação de domínio não supervisionada e a utilização do classificador totalmente conectado (69,1% de acurácia e 64,0% de medida F1, Subseção 6.3.4).

7.2 Contribuições do Trabalho

As principais contribuições deste trabalho são:

1. A criação de uma base de dados para pragas dos citros (Seção 4.1, C1: *"um novo conjunto de dados de referência (benchmark) para o problema do reconhecimento de pragas dos citros, em que pequenas regiões de interesse contendo diferentes tipos de ácaros estão presentes nas imagens"*);

2. O desenvolvimento de pesquisas relacionadas ao aprendizado por múltiplas instâncias (MIL) e mapas de ativação com base no aprendizado profundo (Seções 5.1, 5.2 e 5.3, C2: *"métodos fracamente supervisionados para classificação e localização de pragas"*);

3. A apresentação de um novo método de atenção espacial usando dois pesos (Subseção 5.2.2, C3: *"uma nova formulação matemática que utiliza dois pesos de treinamento para criação de mapas de ativação baseados em atenção"*);

4. Uma estratégia de seleção de múltiplos *patches* baseada em mapas de saliências (Subseção 5.1.5, C4: *"uma estratégia eficaz de seleção de múltiplas regiões de interesse baseadas em mapas de saliências para localizar automaticamente pragas dos citros"*); e

5. A realização de experimentos relacionados a todos os pontos citados, além do uso de técnicas para realce e localização de regiões de interesse muito pequenas em relação ao tamanho das imagens (Capítulo 6, C5: *"resultados competitivos em relação a métodos disponíveis na literatura"*).

A contribuição representada pela criação de um novo conjunto de dados é expressiva no contexto do MIP para pragas invisíveis a olho nu, uma vez que a maior parte das bases contém pragas macroscópicas. A *Citrus Pest Benchmark* (CPB) é a primeira base de dados anotada com rótulos de classificação que traz ácaros coletados em campo e que são extremamente nocivos para a citricultura paulista. A forma de sua coleta já é inovadora, visto que nenhuma base havia usado dispositivos móveis e lupas até sua proposta. A CPB pode servir para automatizar a identificação das pragas e ensinar a tarefa de reconhecer os ácaros em campo aos novos inspetores.

Os processos *Attention-based MIL-Guided* e *MIL-Guided* são contribuições originais para os métodos fracamente supervisionados e fazem a integração de duas de suas abordagens, aprendizado por múltiplas instâncias (MIL) e mapas de ativação. Esses métodos, representados por

suas *Bag Models* e *Instance Models*, são abordagens para (i) reproduzir o efeito de aumento óptico para que os classificadores possam focar nas imagens dos ácaros e (ii) aumentar o número de imagens de treinamento para uma avaliação mais refinada (*Instance Models*).

A proposta para um mapa de ativação baseado em atenção, chamada de *Two-WAM*, é um novo método baseado em atenção que demonstrou ser eficaz ao inferir localizações fracamente supervisionadas de múltiplos objetos. Essa formulação é propícia para uso em qualquer problema que exija a junção de duas ou mais matrizes em apenas uma, como no caso das imagens RGB (Subseção 5.2.2). O *Bag Model* do *Attention-based MIL-Guided*, que usa o *Two-WAM*, conseguiu gerar rótulos de localizações (*bounding boxes*) suficientes para permitir a contagem de um número significativo de ácaros (Subseção 6.2.6).

O *Patch-SaliMap*, a proposta de abordagem de seleção de múltiplos *patches* baseada em mapas de saliências, efetuou adequadamente a criação de instâncias que continham ácaros, quando os modelos foram treinados para a tarefa binária (Subseção 6.1.4).

A partir dos experimentos realizados para demonstrar a efetividade dos métodos propostos, vários conhecimentos foram adquiridos ao longo de sua produção, inclusive os resultados competitivos já citados (Seção 7.1), que foram apresentados nas Subseções 6.1.3, 6.2.3, 6.2.4 e 6.2.5.

7.3 Respostas às Questões de Pesquisa

As questões de pesquisa deste trabalho foram importantes para nortear os experimentos desenvolvidos. Elas e suas respectivas respostas estão listadas a seguir:

Q1: *"Métodos de aprendizado fracamente supervisionados por múltiplas instâncias são eficazes para a classificação de pragas da citricultura em regiões pequenas?"*

Os métodos de aprendizado por múltiplas instâncias ou MIL são eficazes para tarefas binárias de classificação de ácaros, entretanto, enfrentam problemas ao se trabalhar com a classificação multirrótulos. Acredita-se que seja necessário aperfeiçoar ainda mais os métodos de mapas de ativação para que os métodos MIL sejam melhores preditores de diferentes tipos de ácaros. Os experimentos que corroboram com esta resposta estão nas Subseções 6.1.4 e 6.3.2.

Q2: *"Métodos de aprendizado fracamente supervisionados de mapas de ativação são eficazes para localização de pragas?"*

Os experimentos reportados mostraram que a localização fracamente supervisionada dos métodos propostos trazem resultados aceitáveis e, possivelmente, aplicáveis de localização. Esses resultados mostraram que a localização fracamente supervisionada se aproximou da efetividade das localizações totalmente supervisionadas. Os experimentos que corroboram com esta resposta estão nas Subseções 6.2.5 e 6.2.6.

Q3: *"É possível desenvolver uma arquitetura ponta a ponta para gerar mapas de ativação que selecionem múltiplas regiões de interesse para a classificação automática de pragas?"*

Foram desenvolvidos múltiplos testes para gerar uma arquitetura de rede neural profunda ponta a ponta para efetuar a tarefa de classificação binária da CPB. Essa arquitetura uniu todos as etapas das propostas do *MIL-Guided* e *Attention-based MIL-Guided* em apenas uma. Os modelos foram treinados corretamente, entretanto, seus resultados não foram satisfatórios. Portanto, é possível desenvolver a arquitetura ponta a ponta, mas seu uso ainda não é recomendável para a CPB. Para a IP102, com ganho de aproximadamente 2 pontos percentuais, três ramos

(fluxos) diferentes foram aplicados, um para a imagem original, o outro para a imagem sem o plano de fundo e um para múltiplos *patches* da imagem sem o plano de fundo [178, 217]. Apesar disso, o ganho dos resultados é baixo comparado com a complexidade do treinamento. Os experimentos que corroboram com esta resposta estão na Subseção 6.2.7.

Q4: *"Ruídos, tais como luminosidade e borramento presentes na captura de imagens em campo, afetam o treinamento dos modelos de redes neurais profundas?"*

As imagens coletadas diretamente em campo contendo RIs muito pequenas representam corretamente o contexto do trabalho de campo, entretanto, os ruídos contidos nessas imagens prejudicam o treinamento de modelos de redes neurais profundas, produzindo resultados menos satisfatórios. Por outro lado, o ruído em imagens com regiões de interesse grandes e salientes pode beneficiar o processo de treinamento, melhorando os resultados dos modelos. Os experimentos que corroboram com esta resposta estão na Subseção 6.2.2.

Q5: *"Uma localização fracamente supervisionada mais acurada das pragas em um modelo causa uma maior taxa de classificação?"*

Há indícios de que isso não necessariamente ocorre. O método proposto *Two-WAM* mostrou-se efetivo em produzir localizações fracamente supervisionadas melhores do que outros métodos, mas seu uso em RIs muito pequenas nas imagens originais da CPB não trouxe valores de acurácia e medida F1 muito superiores aos *Bag Models*, com diferença de 1 ponto percentual entre os *Bag Models* do *MIL-Guided* e *Attention-based MIL-Guided*. Do mesmo modo, o *Two-WAM* trouxe uma pequena diferença representada por 1 ponto percentual na classificação de imagens da IP102 1.1, mesmo mostrando bons resultados de localização na IP102 1.0. As variações nas médias de classificação são pequenas e talvez insignificantes estatisticamente quando o *Two-WAM* é aplicado, mas suas inferências de localização sempre mostram melhores resultados. O experimento que corrobora com esta resposta está na Seção 6.2.4.

7.4 Aplicações

É comum uma unidade de produção conter até 1.000 árvores de citros (laranjeiras, limeiras e limoeiros), divididas em grupos organizados em ruas. Para aplicar uma solução que automatize as análises descritas no MIP nessas unidades de produção, os inspetores teriam que percorrer as linhas, como já fazem atualmente, e escolher amostras de frutas e folhas, não próximas às bordas (já que são mais propensas a conter ácaros), enquanto manuseiam o celular acoplado à lupa. Eles apontariam a lupa em direção às amostras a serem analisadas, o celular usaria os modelos treinados para predizer se ácaros estariam presentes e, sem ajuda humana, o celular armazenaria as imagens desses ácaros quando presentes. O uso do *Attention-based MIL-Guided* seria a melhor opção para a coleta, pois, além de classificar a região da imagem, pode mostrar em tempo real as localizações fracamente supervisionadas dos ácaros para os inspetores. Assim, o operador humano não efetuaria a procura e análise dos ácaros como faria com as lupas manuais, o que economizaria tempo na aplicação do MIP.

Em campo, ao notar a presença de pragas, os celulares devem armazenar as localizações conjuntamente com as imagens da infecção (usando um GPS ou um número identificador da árvore). Posteriormente, com servidores nos escritórios, um sistema mais robusto, utilizando os modelos para classificação multirrótulos (*Bag Model* do *Attention-based MIL-Guided* com o rótulo da classe positiva no lugar do rótulo da classe negativa e com adaptação de domínio não

supervisionada), carregaria as imagens e quantificaria estatisticamente o número de ácaros de cada classe para cada área das unidades de produção. Com o cálculo da quantificação das infecções por área, o sistema recomendaria as necessidades de manejo de acordo com o percentual de infecção descrito no protocolo do MIP e com as especificidades de cada insumo.

7.5 Trabalhos Futuros

Como trabalhos futuros, pretende-se melhorar a tarefa de classificação multirrótulos para a base de dados CPB, pois é necessário que exista um alto grau de acertos ao se fazer o cálculo das infecções de ácaros por áreas. Também, os modelos binários do *Attention-based MIL-Guided* devem ser adaptados em uma aplicação para coleta de dados em campo, onde as localizações fracamente supervisionadas mostrarão em tempo real a presença dos ácaros na tela do celular.

Um sistema de coleta e monitoramento de pragas deve ser construído com duas partes: (i) um aplicativo móvel para classificar a presença de ácaros e coletar os dados; e (ii) um servidor para armazenar e realizar análises multirrótulos. É necessário avaliar se as regras propostas pelo MIP ainda continuam válidas sob um novo método de coleta de dados, agora automatizado. Então, deve existir uma coleta de informações sobre o sistema e suas partes para adaptar as regras do MIP ao novo cenário de automatização. Outras pragas macroscópicas devem ser adicionadas ao processo para que também sejam quantificadas pelo sistema.

7.6 Considerações Finais

Como consequência da aplicação das técnicas e resultados apresentados, espera-se um processo MIP padronizado, a introdução do conceito de aprendizado fracamente supervisionado na área de prevenção de pragas e um novo paradigma de coleta de dados baseado no reconhecimento automatizado da presença dos ácaros.

Em relação aos objetivos propostos e alcançados, avalia-se que o primeiro objetivo, O1: *"criar uma base de dados com diferentes tipos de ácaros da citricultura paulista"*, foi integralmente atendido com a criação da base de dados CPB e seu subconjunto denominado *Noiseless Citrus Pest Benchmark* ou NCPB (Seção 4.1).

O segundo objetivo, O2: *"verificar a importância do tamanho das regiões de interesse nas imagens"*, também foi integralmente atendido e os experimentos mostraram que o tamanho das imagens para classificação não é o único ponto de atenção. É preciso analisar o quão saliente uma RI é em relação ao tamanho total da imagem e, no caso do uso de *patches*, se as RI estão inteiras nas imagens (as Subseções 6.1.4, 6.2.2, 6.2.3 e 6.2.4 contêm experimentos relacionados a esse objetivo).

Avalia-se que o terceiro objetivo, O3: *"avaliar a integração dos métodos de múltiplas instâncias e mapas de ativação"*, foi parcialmente atendido, mostrando que: é possível a integração de múltiplas instâncias e mapas de ativação para a classificação binária da CPB; é inefetivo à integração para a tarefa de classificação da IP102 1.1; e, com os resultados até o momento, é possível sua integração, mas não se consegue bons resultados para classificação multirrótulos. Experimentos adicionais e outras propostas podem surtir melhores resultados e são necessários (as Subseções 6.1.4, 6.2.4 e 6.3.2 contêm experimentos relacionados a esse objetivo).

O quarto objetivo, O4: *melhorar os resultados do estado da arte para o problema de classificação de pragas nas bases avaliadas*, foi parcialmente alcançado, pois foram exibidos ótimos resultados para a localização fracamente supervisionada da IP102 1.0, resultados competitivos em relação à tarefa multiclasses da IP102 1.1, ótimos resultados para a tarefa de classificação binária da CPB e resultados promissores para a tarefa multirrótulos da CPB. Entretanto, melhores resultados para classificação multirrótulos devem ser alcançados para a automatização da classificação de ácaros.

O quinto objetivo, O5: *"mostrar a viabilidade do uso de métodos fracamente supervisionados na localização de pragas em imagens"*, foi integralmente atendido, inclusive com resultados quantitativos (as Seções 6.2.3, 6.2.5 e 6.2.6 contêm experimentos relacionados a este objetivo).

Finalmente, considera-se que o conteúdo produzido por esta tese traz os primeiros passos para a automatização do MIP relacionada a ácaros da citricultura. Os problemas que foram abordados por esta tese são preocupações constantes da citricultura quando em campo. A implementação desta tese em um sistema computacional trará grandes benefícios científicos e econômicos, além de ser um trabalho com aplicações reais e diretas. Para a produção de citros do interior paulista, a automatização melhorará os trabalhos em campo, o embasamento teórico para aplicação de insumos e o processo de decisão para o contingenciamento de pragas e doenças.

Referências Bibliográficas

[1] R. Achanta, F. Estrada, P. Wils e S. Süsstrunk. Salient region detection and segmentation. In *International Conference on Computer Vision Systems (ICCVS)*, pages 66–75. Springer, 2008. 33

[2] S. Adiga, J. Dolz e H. Lombaert. Manifold-driven attention maps for weakly supervised segmentation. *arXiv:2004.03046*, 2020. 34, 44, 47

[3] S. Adke, C. Li, K. Rasheed e F. Maier. Supervised and weakly supervised deep learning for segmentation and counting of cotton bolls using proximal imagery. *Sensors*, 22(10): 3688, 2022. 42

[4] J. Ahn e S. Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4981–4990, 2018. 43, 47

[5] A. Alfarisy, Q. Chen e M. Guo. Deep learning based classification for paddy pests & diseases recognition. In *International Conference on Mathematics and Artificial Intelligence (ICMAI)*, pages 21–25, 2018. 23, 52, 55, 63

[6] S. Ali e M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Transections on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(2):288–303, 2008. 39, 42

[7] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould e L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018. 48, 51

[8] S. Andrews, I. Tsochantaridis e T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 577–584, 2003. 38, 42

[9] E. Ayan, H. Erbay e F. Varçın. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Computers and Electronics in Agriculture*, 179:105809, 2020. 53, 55, 80, 81

[10] B. Babenko, P. Dollár, Z. Tu e S. Belongie. Simultaneous learning and alignment: multi-instance and multi-pose learning. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008. 38, 42

[11] B. Babenko, M.-H. Yang e S. Belongie. Visual tracking with online multiple instance learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 983–990, 2009. 39, 42

[12] D. Bahdanau, K. Cho e Y. Bengio. Neural machine translation by jointly learning to align and translate. In Y. Bengio e Y. LeCun, editors, *International Conference on Learning Representations (ICLR)*, pages 1–11, 2015. 34, 46, 51

[13] J. Barbedo. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144:52–60, 2016. 19, 50, 55

[14] J. Barbedo. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153:46–53, 2018. 51, 55

[15] J. Barbedo, L. Koenigkan e T. Santos. Identifying multiple plant diseases using digital image processing. *Biosystems Engineering*, 147:104–116, 2016. 50, 55

[16] Y. Bazi, L. Bashmal, M. Rahhal, R. Dayil e N. Ajlan. Vision transformers for remote sensing image classification. *Remote Sensing*, 13(3):516, 2021. 48, 51

[17] F. Behlau, R. Bassanezi, J. Barbosa, I. Sala e V. Trombin. Levantamento de doenças dos citros: HLB, CVC e Cancro Cítrico no cinturão citrícola de São Paulo e Triângulo/Sudoeste Mineiro, 2017. 17

[18] E. Bellocchio, T. Ciarfuglia, G. Costante e P. Valigi. Weakly supervised fruit counting for yield estimation using spatial consistency. *Robotics and Automation Letters*, 4(3): 2348–2355, 2019. 42

[19] A. Bereciartua-Pérez, L. Gómez, A. Picón, R. Navarra-Mestre, C. Klukas e T. Eggers. Insect counting through deep learning-based density maps estimation. *Computers and Electronics in Agriculture*, 197:106933, 2022. 54, 55

[20] S. Bhandari, A. Raheja, R. Green e D. Do. Towards collaboration between unmanned aerial and ground vehicles for precision agriculture. In *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping*, volume 10218, 2017. 52, 55

[21] Y. Bin, Y. Yang, F. Shen, N. Xie, H. Shen e X. Li. Describing video with attention-based bidirectional LSTM. *Transactions on Cybernetics*, 49(7):2631–2641, 2018. 48, 51

[22] A. Blum e A. Kalai. A note on learning from multiple-instance examples. *Machine Learning*, 30(1):23–29, 1998. 37, 42

[23] J. Bobadilla e H. Pedrini. Lung nodule classification based on deep convolutional neural networks. In *Iberoamerican Congress on Pattern Recognition*, pages 117–124. Springer, 2016. 50

[24] E. Bollis, H. Pedrini e S. Avila. Weakly supervised learning guided by activation mapping applied to a novel citrus pest benchmark. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, 2020. 58, 59, 62, 63, 65, 66, 74, 77, 78, 80, 83, 92, 94, 96

[25] E. Bollis, H. Maia, H. Pedrini e S. Avila. Weakly supervised attention-based models using activation maps for citrus mite and insect pest classification. *Computers and Electronics in Agriculture*, 195:106839, 2022. 62, 70, 71, 80, 84, 86, 89, 92, 94, 96

[26] A. Borji, D. Sihite e L. Itti. What stands out in a scene? A study of human explicit saliency judgment. *Vision Research*, 91:62–77, 2013. 32

[27] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang e J. Li. Salient object detection: a survey. *Computational Visual Media*, pages 117–150, 2019. 31, 32, 33, 34

[28] Q. Cap, H. Uga, S. Kagiwada e H. Iyatomi. LeafGAN: an effective data augmentation method for practical plant disease diagnosis. *Transactions on Automation Science and Engineering*, pages 1–10, 2020. 46, 47

[29] M.-A. Carbonneau, V. Cheplygina, E. Granger e G. Gagnon. Multiple instance learning: a survey of problem characteristics and applications. *Pattern Recognition*, 77:329–353, 2018. 30, 31

[30] S. Chaudhari, V. Mithal, G. Polatkan e R. Ramanath. An attentive survey of attention models. *Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–32, 2021. 19, 34

[31] C. Chen, J. Wu, C. Chang e Y.-M. Huang. Agricultural pests damage detection using deep learning. In *International Conference on Network-Based Information Systems*, pages 545–554, 2020. 53, 55, 63

[32] H. Chen, Q. Hu, B. Zhai, H. Chen e K. Liu. A robust weakly supervised learning of deep conv-nets for surface defect inspection. *Neural Computing and Applications*, pages 1–16, 2020. 43, 47

[33] J. Chen, W. Chen, A. Zeb, D. Zhang e Y. Nanehkaran. Crop pest recognition using attention-embedded lightweight network under field conditions. *Applied Entomology and Zoology*, 2021. 46, 47

[34] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu e T.-S. Chua. SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5659–5667, 2017. 35, 49, 51, 56, 57

[35] S.-C. Chen, S. Rubin, M.-L. Shyu e C. Zhang. A dynamic user concept pattern learning framework for content-based image retrieval. *Transactions on Systems, Man, and Cybernetics*, 36(6):772–783, 2006. 39, 42

[36] Y. Chen, Y. Kalantidis, J. Li, S. Yan e J. Feng. A^2-Nets: double attention networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018. 49, 51

[37] Y. Chen, X. Zhang, Z. Chen, M. Song e J. Wang. Fine-grained classification of fly species in the natural environment based on deep convolutional neural network. *Computers in Biology and Medicine*, 135(1):104655, 2021. 46, 47, 103

[38] V. Cheplygina, M. de Bruijne e J. Pluim. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280–296, 2019. 20, 27, 28, 40, 42

[39] R. Child, S. Gray, A. Radford e I. Sutskever. Generating long sequences with sparse transformers. *arXiv:1904.10509*, 2019. 48, 51

[40] J. Choe, S. Oh, S. Lee, S. Chun, Z. Akata e H. Shim. Evaluating weakly supervised object localization methods right. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3130–3139, 2020. 34, 44, 47, 56, 57

[41] F. Chollet. Xception: deep learning with depthwise separable convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 53, 81

[42] Y. Choukroun, R. Bakalo, R. Ben-Ari, A. Akselrod-Ballin, E. Barkan e P. Kisilev. Mammogram classification and abnormality detection from nonlocal labels using deep multiple instance neural network. In *Eurographics Workshop on Visual Computing for Biology and Medicine*, pages 11–19, 2017. 40, 42

[43] P. Chudzik, A. Mitchell, M. Alkaseem, Y. Wu, S. Fang, T. Hudaib, S. Pearson e B. Al-Diri. Mobile real-time grasshopper detection and data aggregation framework. *Scientific Reports*, 10(1):1–10, 2020. 54, 55, 63, 64

[44] R. Cinbis, J. Verbeek e C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(1):189–203, 2016. 40, 42

[45] D. Comaniciu e P. Meer. Mean Shift: a robust approach toward feature space analysis. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):603–619, 2002. 68

[46] A. Correia e E. Colombini. Attention, please! A survey of neural attention models in deep learning. *Artificial Intelligence Review*, pages 1–88, 2022. 46, 47, 51

[47] A. Cruz, A. Luvisi, L. de Bellis e Y. Ampatzidis. Vision-based plant disease detection system using transfer and deep learning. In *Annual International Meeting*. American Society of Agricultural and Biological Engineers, 2017. 51, 55

[48] F. de Defesa da Citricultura. Levantamentos: incidência de doenças e insetos pragas dos citros. http://www.fundecitrus.com.br/levantamentos, 2018. 17

[49] F. de Defesa da Citricultura. Doenças e pragas: Cancro cítrico. http://www.fundecitrus.com.br/doencas/cancro/7, Acessado: 02-05-2022. 24

[50] F. de Defesa da Citricultura. Doenças e pragas: Greening / hlb. http://www.fundecitrus.com.br/doencas/greening/10, Acessado: 02-05-2022. 24

[51] F. de Defesa da Citricultura. Ácaros: ácaros da leprose. http://www.fundecitrus.com.br/doencas/acaros/26, Acessado: 02-05-2022. 17

[52] L. Deng, Y. Wang, Z. Han e R. Yu. Research on insect pest image detection and recognition based on bio-inspired methods. *Biosystems Engineering*, 169:139–148, 2018. 23, 50, 53, 55, 63

[53] Z. Deng. Survey on various approaches of saliency detection. In *International Conference on Machine Learning, Big Data and Business Intelligence*, pages 358–363. IEEE, 2019. 33

[54] T. Dietterich, R. Lathrop e T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 29, 37, 39, 42, 56, 57

[55] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit e N. Houlsby. An image is worth 16x16 words: transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 48, 51

[56] T. Durand, T. Mordan, N. Thome e M. Cord. WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 642–651, 2017. 34, 44, 47, 56, 57, 84, 90, 92, 93, 96

[57] M. Everingham, L. Gool, C. Williams, J. Winn e A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 76

[58] X. Feng, J. Yang, A. Laine e E. Angelini. Discriminative localization in CNNs for weakly-supervised segmentation of pulmonary nodules. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 568–576. Springer, 2017. 44, 47

[59] K. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018. 52, 55

[60] A. Fuentes, S. Yoon, S. C. Kim e D. S. Park. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9):2022, 2017. 23, 51, 52, 55

[61] Y. Ganin e V. Lempitsky. Unsupervised domain adaptation by backpropagation. In F. Bach e D. Blei, editors, *International Conference on Machine Learning (ICML)*, volume 37, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. 35, 36, 56, 57, 73, 74

[62] W. Gao, F. Wan, J. Yue, S. Xu e Q. Ye. Discrepant multiple instance learning for weakly supervised object detection. *Pattern Recognition*, 122:108233, 2022. 41, 42

[63] A. Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2017. 30

[64] W. Gondal, J. Köhler, R. Grzeszick, G. Fink e M. Hirsch. Weakly-supervised localization of diabetic retinopathy lesions in retinal fundus images. In *International Conference on Image Processing (ICIP)*, pages 2069–2073. IEEE, 2017. 44, 47

[65] E. Grafton-Cardwell. Huanglongbing (HLB or Citrus Greening). http://cisr.ucr.edu/citrus_greening.html, Acessado: 02-05-2022. 17

[66] S. Gravena. Manejo integrado de pragas é vital na produção de citros. *Visão Agrícola*, 2 (1), 1998. 17, 24

[67] S. Gravena. *Manual prático de manejo ecológico de pragas dos citros*. Gravena, 2005. 24, 60

[68] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng e S.-M. Hu. Visual attention network. *arXiv:2202.09741*, 2022. 49, 51

[69] K. Hacıefendioğlu, H. Başağa, Z. Yavuz e M. Karimi. Intelligent ice detection on wind turbine blades using semantic segmentation and class activation map approaches based on deep learning method. *Renewable Energy*, 182:1–16, 2022. 45, 47

[70] N. Hashimoto, D. Fukushima, R. Koga, Y. Takagi, K. Ko, K. Kohno, M. Nakaguro, S. Nakamura, H. Hontani e I. Takeuchi. Multi-scale domain-adversarial multiple-instance cnn for cancer subtype classification with non-annotated histopathological images. *arXiv:2001.01599*, 2020. 30, 31, 40, 42, 56, 57

[71] K. He, X. Zhang, S. Ren e J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 29, 78, 81, 84, 97

[72] K. He, J. Huo, Y. Shi, Y. Gao e D. Shen. MIDCN: a multiple instance deep convolutional network for image classification. In *Pacific Rim International Conference on Artificial Intelligence*, pages 230–243. Springer, 2019. 31, 40, 42

[73] K. He, W. Zhao, X. Xie, W. Ji, M. Liu, Z. Tang, Y. Shi, F. Shi, Y. Gao, J. Liu et al. Synergistic learning of lung lobe segmentation and hierarchical multi-instance classification for automated severity assessment of COVID-19 in CT images. *Pattern recognition*, 113:107828, 2021. 41, 42

[74] Y. He, H. Zeng, Y. Fan, S. Ji e J. Wu. Application of deep learning in integrated pest management: a real-time system for detection and diagnosis of oilseed rape pests. *Mobile Information Systems*, 2019, 2019. 53, 55, 63

[75] D. Hernández-Rabadán, F. Ramos-Quintana e J. Guerrero J. Integrating soms and a bayesian classifier for segmenting diseased plants in uncontrolled environments. *The Scientific World Journal*, 2014. 50, 55

[76] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen e T. Sainath. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Processing Magazine*, 29(6): 82–97, 2012. 29

[77] A. Hodges, M. Rahmani e T. Spreen. Economic contributions of the Florida citrus industry in 2015-16. *Electronic Data Information Source*, 2018. 17

[78] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam e Q. Le. Searching for MobileNetV3. In *International Conference on Computer Vision (ICCV)*, pages 1314–1324, (2019). 48

[79] J. Hu, L. Shen, S. Albanie, G. Sun e E. Wu. Squeeze-and-Excitation Networks. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(8):2011–2023, 2020. 48, 49, 50, 51

[80] G. Huang, Z. Liu, L. Van Der Maaten e K. Q. Weinberger. Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017. 81

[81] D. Hughes e M. Salathé. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv:1511.08060*, 2015. 19, 51, 55

[82] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally e K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv:1602.07360*, 2016. 75

[83] M. Ilse, J. Tomczak e M. Welling. Attention-based deep multiple instance learning. In *International Conference on Machine Learning (ICML)*, pages 2127–2136. PMLR, 2018. 31, 39, 40, 42, 56, 57, 84, 85, 90, 92, 93, 96

[84] L. Itti, C. Koch e E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(11): 1254–1259, 1998. 33

[85] M. Izadyyazdanabadi, E. Belykh, C. Cavallo, X. Zhao, S. Gandhi, L. Moreira, J. Eschbacher, P. Nakaji, M. Preul e Y. Yang. Weakly-supervised learning-based feature localization for confocal laser endomicroscopy glioma images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 300–308. Springer, 2018. 44, 47

[86] M. Jaderberg, K. Simonyan e A. Zisserman. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2017–2025, 2015. 47, 51

[87] J. Kiefer e J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952. 72

[88] W. Kim, D. Lee e Y. Kim. Machine vision-based automatic disease symptom detection of onion downy mildew. *Computers and Electronics in Agriculture*, 168:105099, 2020. 41, 46, 47

[89] D. Kingma e J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 53

[90] S. Kornblith, J. Shlens e Q. Le. Do better ImageNet models transfer better? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2661–2671, 2019. 66

[91] A. Krizhevsky, I. Sutskever e G. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012. 27, 29, 30, 78, 84

[92] S. Kwak, S. Hong e B. Han. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI Conference on Artificial Intelligence*, 2017. 43, 47

[93] N. Larios, B. Soran, L. Shapiro, G. Martinez-Munoz, J. Lin e T. Dietterich. Haar Random Forest features and SVM spatial matching kernel for stonefly species identification. In *International Conference on Pattern Recognition (ICPR)*, pages 2624–2627. IEEE, 2010. 50, 55

[94] Y. LeCun, L. Bottou, Y. Bengio e P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 29, 30, 84, 89

[95] Y. LeCun, Y. Bengio e G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015. 18, 29

[96] C. Leistner, A. Saffari e H. Bischof. MIForests: multiple-instance learning with randomized trees. In *European Conference on Computer Vision (ECCV)*, pages 29–42. Springer, 2010. 38, 42

[97] B. Li, Y. Li e K. W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, 2021. 41, 42

[98] C. Li, T. Zhen e Z. Li. Image classification of pests with residual neural network based on transfer learning. *Applied Sciences*, 12(9):4356, 2022. 41, 42

[99] H.-C. Li, W.-S. Hu, W. Li, J. Li, Q. Du e A. Plaza. A^3CLNN: spatial, spectral and multiscale attention ConvLSTM neural network for multisource remote sensing data classification. *arXiv:2204.04462*, 2022. 48, 51

[100] J. Li, H. Zhou, D. Jayas e Q. Jia. Construction of a dataset of stored-grain insects images for intelligent monitoring. *Transactions of the ASABE*, page 0, 2019. 53, 55, 63, 64

[101] M. Li, L. Wu, A. Wiliem, K. Zhao, T. Zhang e B. Lovell. Deep instance-level hard negative mining model for histopathology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 514–522, 2019. 40, 42

[102] R. Li, R. Wang, C. Xie, L. Liu, J. Zhang, F. Wang e W. Liu. A coarse-to-fine network for aphid recognition and detection in the field. *Biosystems Engineering*, 187:39–52, 2019. 19, 54, 55, 63, 64

[103] R. Li, R. Wang, C. Xie, H. Chen, Q. Long, L. Liu, J. Zhang, T. Chen, H. Hu, L. Jiao, J. Du e H. Liu. A multi-branch convolutional neural network with density map for aphid counting. *Biosystems Engineering*, 213:148–161, 2022. 54, 55

[104] W. Li, D. Wang, M. Li, Y. Gao, J. Wu e X. Yang. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Computers and Electronics in Agriculture*, 183, 2021. 54, 55, 63, 64

[105] X. Li, Y. Lang, Y. Chen, X. Mao, Y. He, S. Wang, H. Xue e Q. Lu. Sharp multiple instance learning for deepfake video detection. In *International Conference on Multimedia*, pages 1864–1872, 2020. 41, 42

[106] Y. Li e J. Yang. Few-shot cotton pest recognition and terminal realization. *Computers and Electronics in Agriculture*, 169:105240, 2020. 54, 55

[107] B. Liang, Y. Liu, L. He e J. Li. Weakly supervised semantic segmentation based on deep learning. In *International Conference on Modelling, Identification and Control*, pages 455–464, 2020. 40, 42

[108] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár e L. Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 27

[109] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan e S. Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 53, 81, 97, 98

[110] E. Lins, J. Rodriguez, S. Scoloski, J. Pivato, M. Lima, J. Fernandes, P. da S. Pereira, D. Lau e R. Rieder. A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture*, 169:105200, 2020. 54, 55, 63

[111] B. Liu, Y. Zhang, D. He e Y. Li. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1):11, 2018. 51, 55

[112] K. Liu, Y. Shen, N. Wu, J. Chłędowski, C. Fernandez-Granda e K. Geras. Weakly-supervised high-resolution segmentation of mammography images for breast cancer diagnosis. *arXiv:2106.07049*, 2021. 45, 47

[113] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, S. Sudirman e W. Liu. PestNet: an end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access*, 7:45301–45312, 2019. 49, 51, 63

[114] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu e A. C. Berg. Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, pages 21–37. Springer, 2016. 97, 98

[115] W. Liu, G. Wu e F. Ren. Deep multi-branch fusion residual network for insect pest recognition. *Transactions on Cognitive and Developmental Systems*, 2020. 52, 55, 80

[116] X. Liu, W. Min, S. Mei, L. Wang e S. Jiang. Plant disease recognition: a large-scale benchmark dataset and a visual region and loss reweighting approach. *Transactions on Image Processing*, 30:2003–2015, 2021. 54, 55

[117] Y. Liu, S. Liu, J. Xu, X. Kong, L. Xie, K. Chen, Y. Liao, B. Fan e K. Wang. Forest pest identification based on a new dataset and convolutional neural network model with enhancement strategy. *Computers and Electronics in Agriculture*, 192:106625, 2022. URL https://www.sciencedirect.com/science/article/pii/S0168169921006426. 46, 47, 63

[118] Z. Liu, J. Gao, G. Yang, H. Zhang e Y. He. Localization and classification of paddy field pests using a saliency map and deep convolutional neural network. *Scientific Reports*, 6: 20410, 2016. 52, 55, 63

[119] J. Long, E. Shelhamer e T. Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 30

[120] P. Long e L. Tan. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, 30(1):7–21, 1998. 37, 42

[121] J. Lu, J. Hu, G. Zhao, F. Mei e C. Zhang. An in-field automatic wheat disease diagnosis system. *Computers and Electronics in Agriculture*, 142(1):369–379, 2017. 41, 42, 57, 95, 100

[122] Q. Luo, L. Wan, L. Tian e Z. Li. Saliency guided discriminative learning for insect pest recognition. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 46, 47

[123] Z. Luo, D. Guillory, B. Shi, W. Ke, F. Wan, T. Darrell e H. Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *European Conference on Computer Vision (ECCV)*, pages 729–745. Springer, 2020. 40, 42

[124] K. Ma, M.-J. Nie, S. Lin, J. Kong, C.-C. Yang e J. Liu. Fine-grained pests recognition based on truncated probability fusion network via internet of things in forestry and agricultural scenes. *Algorithms*, 14(10):290, 2021. 54, 55

[125] S. Majid, F. Alenezi, S. Masood, M. Ahmad, E. Gündüz e K. Polat. Attention based CNN model for fire detection and localization in real-world images. *Expert Systems with Applications*, 189:116114, 2022. 49, 51

[126] O. Maron. Learning from ambiguity. Technical report, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, 1998. 38, 42

[127] S. Mehta, M. Ghazvininejad, S. Iyer, L. Zettlemoyer e H. Hajishirzi. DeLighT: deep and light-weight transformer. In *International Conference on Learning Representations (ICLR)*, 2021. 48, 51

[128] J. Mendoza e H. Pedrini. Detection and classification of lung nodules in chest X-ray images using deep convolutional neural networks. *Computational Intelligence*, 36(2): 370–401, May 2020. 82

[129] A. Menegola, M. Fornaciali, R. Pires, F. Bittencourt, S. Avila e E. Valle. Knowledge transfer for melanoma screening with deep learning. In *International Symposium on Biomedical Imaging (ISBI)*, pages 297–300, 2017. 30, 50

[130] V. Mnih, N. Heess, A. Graves e K. Kavukcuoglu. Recurrent models of visual attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014. 47, 51

[131] S. Mohanty, D. Hughes e M. Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016. 51, 55

[132] L. Nachtigall, R. Araujo e G. Nachtigall. Classification of apple tree disorders using convolutional neural networks. In *International Conference on Tools with Artificial Intelligence*, pages 472–476, 2016. 23, 51, 55

[133] L. Nanni, G. Maguolo e F. Pancino. Insect pest image detection and recognition based on bio-inspired methods. *Ecological Informatics*, page 101089, 2020. 23, 52, 55, 80

[134] L. Nanni, A. Manfè, G. Maguolo, A. Lumini e S. Brahnam. High performing ensemble of convolutional neural networks for insect pest image detection. *Ecological Informatics*, 67:101515, 2022. 53, 55, 80, 81

[135] P. Nguyen, T. Liu, G. Prasad e B. Han. Weakly supervised action localization by sparse temporal pooling network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6752–6761, 2018. 43, 47

[136] M. Oquab, L. Bottou, I. Laptev e J. Sivic. Is object localization for free? Weakly-supervised learning with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 685–694, 2015. 31, 43, 47, 57

[137] A. Ouadou. *Vehicle detection using morphological shared-weight neural network in the multiple instance learning framework*. PhD thesis, University of Missouri–Columbia, 2017. 37, 38, 42

[138] G. Papandreou, I. Kokkinos e P.-A. Savalle. Modeling local and global deformations in deep learning: epitomic convolution, multiple instance learning, and sliding window detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 390–399, 2015. 39, 42

[139] H. Pei, K. Liu, X. Zhao e A. Yahya. Enhancing aphid detection framework based on ORB and convolutional neural networks. *Scientific Reports*, 10(1):1–15, 2020. 19, 54, 55, 63, 64

[140] F. Perazzi, P. Krähenbühl, Y. Pritch e A. Hornung. Saliency filters: contrast based filtering for salient region detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–740. IEEE, 2012. 32

[141] N. Petrellis. A smart phone image processing application for plant disease diagnosis. In *International Conference on Modern Circuits and Systems Technologies*, pages 1–4, 2017. 52, 55

[142] D. Petti e C. Li. Weakly-supervised learning to automatically count cotton flowers from aerial imagery. *Computers and Electronics in Agriculture*, 194:106734, 2022. 41, 42, 105

[143] P. Pinheiro e R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1713–1721, 2015. 43, 47

[144] B. Praveen e V. Menon. A bidirectional deep-learning-based spectral attention mechanism for hyperspectral data classification. *Remote Sensing*, 14(1):217, 2022. 48, 51

[145] C. Raffel e D. Ellis. Feed-forward networks with attention can solve some long-term memory problems. In *International Conference on Learning Representations (ICLR)*, pages 1–6, 2016. 34, 47, 51

[146] J. Redmon e A. Farhadi. YOLOv3: an incremental improvement. *arXiv:1804.02767*, 2018. 97, 98

[147] F. Ren, W. Liu e G. Wu. Feature reuse residual networks for insect pest recognition. *IEEE Access*, 7:122758–122768, 2019. 52, 55, 80

[148] S. Ren, K. He, R. Girshick e J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 50, 54, 97, 98

[149] H. Robbins e S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951. 72

[150] O. Ronneberger, P. Fischer e T. Brox. U-NET: convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 46

[151] J. Rony, S. Belharbi, J. Dolz, B. Ayed, L. McCaffrey e E. Granger. Deep weakly-supervised learning methods for classification and localization in histology images: a survey. *arXiv:1909.03354*, 2019. 43, 47

[152] C. Rother, V. Kolmogorov e A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004. 46

[153] S. Sabour, N. Frosst e G. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3856–3866, 2017. 40

[154] A. Sanakoyeu, V. Tschernezki, U. Buchler e B. Ommer. Divide and conquer the embedding space for metric learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 471–480, 2019. 44

[155] J. Sánchez, F. Perronnin, T. Mensink e J. Verbeek. Image classification with the fisher vector: theory and practice. *International journal of computer vision*, 105(3):222–245, 2013. 40

[156] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov e L.-C. Chen. MobileNetV2: inverted residuals and linear bottlenecks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 53, 78, 81

[157] S. Sankaran, A. Mishra, J. Maja e R. Ehsani. Visible-near infrared spectroscopy for detection of Huanglongbing in citrus orchards. *Computers and Electronics in Agriculture*, 77(2):127–134, 2011. 50, 55

[158] T. T. Santos, L. L. de Souza, A. A. dos Santos e S. Avila. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Computers and Electronics in Agriculture*, 170:105247, 2020. 50

[159] H. Santos Filho, C. Azevedo, A. do Nascimento e J. de Carvalho. Manual prático para o monitoramento e controle das pragas da lima ácida tahiti. *Embrapa Mandioca e Fruticultura Tropical: Documentos*, 2009. 17, 18, 24, 25

[160] A. See, P. Liu e C. Manning. Get to the point: summarization with pointer-generator networks. *arXiv:1704.04368*, 2017. 46

[161] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh e D. Batra. Grad-CAM: visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 33, 43, 47, 56, 57, 67, 78, 94

[162] M. Seo, A. Kembhavi, A. Farhadi e H. Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv:1611.01603*, 2016. 46, 51

[163] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009. 27, 41

[164] A. Shaqiri, M. Roinishvili, L. Grzeczkowski, E. Chkonia, K. Pilz, C. Mohr, A. Brand, M. Kunchulia e M. Herzog. Sex-related differences in vision are heterogeneous. *Scientific reports*, 8(1):1–10, 2018. 17

[165] Y. Shen, N. Wu, J. Phang, J. Park, G. Kim, L. Moy, K. Cho e K. Geras. Globally-aware multiple instance classifier for breast cancer screening. *Lecture Notes in Computer Science (LNCS)*, 11861:18–26, 2019. 45, 47

[166] Y. Shen, N. Wu, J. Phang, J. Park, K. Liu, S. Tyagi, L. Heacock, S. Kim, L. Moy, K. Cho e K. Geras. An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization. *Medical Image Analysis*, 68:101908, 2021. 45, 47, 102

[167] K. Simonyan e A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. 97

[168] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever e R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 30, 78

[169] M. Sun, T. Han, M.-C. Liu e A. Khodayari-Rostamabad. Multiple instance learning convolutional neural networks for object recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 3270–3275, 2016. 31, 39, 42

[170] W. Sun, J. Zhang e N. Barnes. Inferring the class conditional response map for weakly supervised semantic segmentation. In *Winter Conference on Applications of Computer Vision (WCACV)*, pages 2878–2887, 2022. 43, 47

[171] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke e A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 29, 53, 81

[172] C. Szegedy, S. Ioffe, V. Vanhoucke e A. Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, 2017. 52, 78

[173] M. Tan e Q. Le. EfficientNet: rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019. 48, 51, 78, 81, 83, 84, 93, 97, 104

[174] W. Tan, C. Zhao e H. Wu. Intelligent alerting for fruit-melon lesion image based on momentum deep learning. *Multimedia Tools and Applications*, 75(24):16741–16761, 2016. 51, 55

[175] Z.-X. Tan, A. Goel, T.-S. Nguyen e D. C. Ong. A multimodal LSTM for predicting listener empathic responses over time. In *International Conference on Automatic Face & Gesture Recognition*, pages 1–4. IEEE, 2019. 48, 51

[176] E. Teh, M. Rochan e Y. Wang. Attention networks for weakly supervised object localization. In *British Machine Vision Conference*, pages 1–11, 2016. 43, 47

[177] K. Thenmozhi e S. Reddy. Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture*, 164:104906, 2019. 53, 55, 63, 64

[178] H. Ung, H. Ung e B. Nguyen. An efficient insect pest classification using multiple convolutional neural network based models. *arXiv:2107.12189*, 2021. 53, 55, 80, 81, 117

[179] E. Valle, M. Fornaciali, A. Menegola, J. Tavares, F. Bittencourt, L. Li e S. Avila. Data, depth, and design: learning reliable models for skin lesion analysis. *Neurocomputing*, 383:303–313, 2020. 82

[180] L. Van der Maaten e G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008. 107, 108

[181] J. Vanston e L. Strother. Sex differences in the human visual system. *Journal of Neuroscience Research*, 95(1-2):617–625, 2017. 17

[182] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser e I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 47, 51

[183] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio e Y. Bengio. Graph attention networks. *arXiv:1710.10903*, 2017. 48, 51

[184] P. Vitorino, S. Avila, M. Perez e A. Rocha. Leveraging deep neural networks to fight child pornography in the age of social media. *Journal of Visual Communication and Image Representation*, 50:303–313, 2018. 50

[185] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang e X. Tang. Residual attention network for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2017. 49, 51, 53, 81

[186] F. Wang, R. Wang, C. Xie, P. Yang e L. Liu. Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition. *Computers and Electronics in Agriculture*, 169:105222, 2020. 50, 51

[187] G. Wang, Y. Sun e J. Wang. Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience*, Volume 2017, 2017. 52, 55

[188] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel e X. Hu. Score-CAM: score-weighted visual explanations for convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 24–25, 2020. 43, 47

[189] J. Wang e J.-D. Zucker. Solving the multiple-instance problem: a lazy learning approach. In *International Conference on Machine Learning (ICML)*, page 1119–1126, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 38, 42

[190] R. Wang, L. Jiao, C. Xie, P. Chen, J. Du e R. Li. S-RPN: sampling-balanced region proposal network for small crop pest detection. *Computers and Electronics in Agriculture*, 187(December 2020):106290, 2021. 19, 50, 51, 84

[191] R. Wang, L. Liu, C. Xie, P. Yang, R. Li e M. Zhou. AgriPest: a large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors*, 21(5):1–15, 2021. 54, 55, 63, 64

[192] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling e R. Yang. Salient object detection in the deep learning era: an in-depth survey. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 31

[193] Z. Wang, W. Gong e W. Li. A dynamic feature weighting method for mangrove pests image classification with heavy-tailed distributions. *International Conference Proceeding Series*, 2020. 46, 47, 103

[194] X.-S. Wei e Z.-H. Zhou. An empirical study on image bag generators for multi-instance learning. *Machine Learning*, 105(2):155–198, 2016. 39, 42

[195] G. Wilson e D. Cook. A survey of unsupervised deep domain adaptation. *Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 35

[196] S. Woo, J. Park, J. Lee e I. S. Kweon. CBAM: convolutional block attention module. In *European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 49, 51, 56, 57

[197] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng e J. Yang. IP102: a large-scale benchmark dataset for insect pest recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8787–8796, 2019. 19, 24, 52, 55, 56, 57, 63, 64, 74, 77, 79, 80, 94, 96, 97, 103

[198] Y. Wu e L. Xu. Crop organ segmentation and disease identification based on weakly supervised deep neural network. *Agronomy*, 9(11):737, 2019. 54, 55

[199] Z.-Z. Wu, J. Xu, Y. Wang, F. Sun, M. Tan e T. Weise. Hierarchical fusion and divergent activation based weakly supervised learning for object detection from remote sensing images. *Information Fusion*, 80:23–43, 2022. 45, 47

[200] C. Xie, J. Zhang, R. Li, J. Li, P. Hong, J. Xia e P. Chen. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Computers and Electronics in Agriculture*, 119:123–132, 2015. 50, 55, 63

[201] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen e H. Chen. Multi-level learning features for automatic classification of field crop pests. *Computers and Electronics in Agriculture*, 152:233–241, 2018. 50, 53, 55, 63

[202] Q. Xie, M.-T. Luong, E. Hovy e Q. Le. Self-training with noisy student improves imagenet classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10687–10698, 2020. 27

[203] S. Xing e M. Lee. A study of tangerine pest recognition using advanced deep learning methods, 2018. 52, 55, 63

[204] C. Xiong, V. Zhong e R. Socher. Dynamic coattention networks for question answering. *arXiv:1611.01604*, 2016. 46, 51

[205] G. Xu, Z. Song, Z. Sun, C. Ku, Z. Yang, C. Liu, S. Wang, J. Ma e W. Xu. CAMEL: a weakly supervised learning framework for histopathology image segmentation. In *International Conference on Computer Vision (ICCV)*, pages 10682–10691, 2019. 30, 40, 42

[206] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel e Y. Bengio. Show, attend and tell: neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, pages 2048–2057. PMLR, 2015. 48, 51

[207] L. Xu e Y. Wang. XCloud: design and Implementation of AI Cloud Platform with RESTful API Service. *arXiv:1912.10344*, 2019. 52, 55, 80

[208] W. Xue, C. Cao, J. Liu, Y. Duan, H. Cao, J. Wang, X. Tao, Z. Chen, M. Wu, J. Zhang et al. Modality alignment contrastive learning for severity assessment of covid-19 from lung ultrasound and clinical information. *Medical image analysis*, 69:101975, 2021. 41, 42

[209] Y. Yan, X. Wang, X. Guo, J. Fang, W. Liu e J. Huang. Deep multi-instance learning with dynamic pooling. In *Asian Conference on Machine Learning*, pages 662–677, 2018. 40, 42

[210] G. Yang, G. Chen, C. Li, J. Fu, Y. Guo e H. Liang. Convolutional rebalancing network for the classification of large imbalanced rice pest and disease datasets in the field. *Frontiers in Plant Science*, 12(July):1–14, 2021. 46, 47, 63, 80, 81

[211] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola e E. Hovy. Hierarchical attention networks for document classification. In *Conference of The North American Chapter of The Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016. 46, 51

[212] N. Yao e X. Wu. Classification of press plate image based on attention mechanism. *International Conference on Safety Produce Informatization (IICSPI)*, pages 129–132, 2019. 49, 51

[213] G. Yu, A. Zare, W. Xu, R. Matamala, J. Reyes-Cabrera, F. Fritschi e T. Juenger. Weakly supervised minirhizotron image segmentation with MIL-CAM. In *European Conference on Computer Vision (ECCV)*, pages 433–449. Springer, 2020. 42

[214] Q. Yu, J.-Y. Song, X.-H. Yu, K. Cheng e G. Chen. To solve the problems of combat mission predictions based on multi-instance genetic fuzzy systems. *The Journal of Supercomputing*, pages 1–22, 2022. 41, 42

[215] M. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv:1212.5701*, 2012. 78

[216] W. Zeng e M. Li. Crop leaf disease recognition based on self-attention convolutional neural network. *Computers and Electronics in Agriculture*, 172:105341, 2020. 50, 51

[217] F. Zhang, G. Zhai, M. Li e Y. Liu. Three-branch and mutil-scale learning for fine-grained image recognition (TBMSL-NET). *arXiv:2003.09150*, 2020. 53, 80, 81, 117

[218] M.-L. Zhang e Z.-H. Zhou. Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, 19(1):1–10, 2004. 39, 42

[219] Q. Zhang e S. Goldman. EM-DD: an improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1073–1080, 2002. 38, 42

[220] R. Zhang, F. Meng, H. Li, Q. Wu e K. Ngan. Category boundary re-decision by component labels to improve generation of class activation map. *Neurocomputing*, 469:105–118, 2022. 44, 47

[221] S. Zhang, L. Wen, X. Bian, Z. Lei e S. Li. Single-shot refinement neural network for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4203–4212, 2018. 97, 98

[222] S. Zhang, R. Jing e X. Shi. Crop pest recognition based on a modified capsule network. *Systems Science & Control Engineering*, 10(1):552–561, 2022. 53, 55

[223] X. Zhang, X. Zhou, M. Lin e J. Sun. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 81

[224] Y. Zhang, J. Hare e A. Prügel-Bennett. Learning to count objects in natural images for visual question answering. *arXiv:1802.05766*, 2018. 48, 51

[225] Y. Zhang, K. Li, K. Li, B. Zhong e Y. Fu. Residual non-local attention networks for image restoration. *arXiv:1903.10082*, 2019. 49, 51

[226] X. Zhao, X. Han, W. Su e Z. Yan. Time series prediction method based on convolutional autoencoder and LSTM. In *Chinese Automation Congress (CAC)*, pages 5790–5793. IEEE, 2019. 48

[227] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva e A. Torralba. Learning deep features for discriminative localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016. 29, 32, 33, 43, 47, 56, 57

[228] Z.-H. Zhou. Multi-instance learning from supervised view. *Journal of Computer Science and Technology*, 21(5):800–809, 2006. 38, 42

[229] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018. 26, 27, 38, 42

[230] Z.-H. Zhou e J.-M. Xu. On the relation between multi-instance learning and semi-supervised learning. In *International Conference on Machine Learning (ICML)*, pages 1167–1174, 2007. 37, 38, 42, 56, 57

[231] Z.-H. Zhou e M.-L. Zhang. Neural networks for multi-instance learning. In *International Conference on Intelligent Information Technology*, pages 455–459, 2002. 39, 42

[232] Z.-H. Zhou, Y.-Y. Sun e Y.-F. Li. Multi-instance learning by treating instances as non-iid samples. In *International Conference on Machine Learning*, pages 1249–1256, 2009. 38, 42

[233] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann e J. Jiao. Learning instance activation maps for weakly supervised instance segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPRW)*, pages 3116–3125, 2019. 43, 47

[234] B. Zoph, V. Vasudevan, J. Shlens e Q. Le. Learning transferable architectures for scalable image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8697–8710, 2018. 78