

José Guilherme Mota Esgario

**Aprendizado Profundo para Classificação e
Quantificação de Doenças e Pragas em
Imagens de Folhas de Café**

Vitória, ES

2019

José Guilherme Mota Esgario

Aprendizado Profundo para Classificação e Quantificação de Doenças e Pragas em Imagens de Folhas de Café

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Orientador: Prof. Dr. -Ing. Renato A. Krohling

Vitória, ES

2019

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

M917a Mota Esgario, José Guilherme, 1992-
Aprendizado profundo para classificação e quantificação de doenças e pragas em imagens de folhas de café / José Guilherme Mota Esgario. - 2019.
77 f.

Orientador: Renato Antônio Krohling.
Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Redes neurais (Computação). 2. Visão por computador. 3. Aprendizado do computador. 4. Café - Doenças e pragas. I. Krohling, Renato Antônio. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 004



APRENDIZADO PROFUNDO PARA CLASSIFICAÇÃO E QUANTIFICAÇÃO DE DOENÇAS E PRAGAS EM IMAGENS DE FOLHAS DE CAFÉ

José Guilherme Mota Esgario

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para a obtenção do grau de Mestre em Informática.

Aprovada em 27 de setembro de 2019:

Renato Krohling

Prof. Dr. Renato Antônio Krohling
Orientador(a)

Thiago Oliveira dos Santos

Prof. Dr. Thiago Oliveira dos Santos
Membro Interno

Daniel Cruz Cavalieri

Prof. Dr. Daniel Cruz Cavalieri
Membro Externo

José Aires Ventura

Prof. Dr. José Aires Ventura
Membro Externo

Declaração de autoria

Eu, José Guilherme Mota Esgario, declaro que este trabalho foi escrito e desenvolvido por mim. Toda e qualquer ajuda recebida durante todo o processo de desenvolvimento foi devidamente reconhecida. Além disso, certifico que todas as fontes de informação e literatura utilizadas são indicadas na dissertação.

Vitória, ES

2019

Agradecimentos

Primeiramente agradeço aos meus pais Osmar e Maria, sem eles certamente eu não teria chegado onde cheguei. Seu apoio, amor e carinho me deram força e vontade suficientes para trilhar este caminho.

Agradeço ao meu irmão, avós e todos os demais familiares pelo apoio e palavras de motivação.

Agradeço à minha namorada Larissa, por estar sempre ao meu lado, tolerar meu estresse e até me ajudar nas correções gramaticais.

Agradeço ao meu professor orientador Renato A. Krohling por todo o apoio, incentivo e confiança em mim depositados.

Aos membros da banca, Prof. Dr. Thiago Oliveira dos Santos e Prof. Dr. Daniel Cruz Cavalieri.

Agradeço ao Pedro Biasutti por seu auxílio no desenvolvimento da aplicação móvel gerada a partir deste trabalho.

Agradeço aos meus amigos do LABCIN, pelo apoio, conversas e debates que foram muito proveitosos e repletos de conhecimento.

Agradeço ao professor Dr. José Aires Ventura por ter disponibilizado seu tempo e compartilhado seu profundo conhecimento científico em Fitopatologia que foi fundamental para o desenvolvimento deste trabalho.

Agradeço à FAPES pela bolsa concedida a mim durante meu período de mestrado.

Agradeço também ao Google pela bolsa de pesquisa que motivou a produção desta dissertação.

Essa Dissertação surgiu como fruto do projeto de pesquisa premiado pelo Google Latin America Research Award (LARA 2018-19) com o título *E-farmer - An app based on image recognition to assist farmers in the identification and control of diseases of coffee trees*, elaborado pelo Prof. Renato Krohling com Helder Knidel, a quem agradeço pela ajuda no desenvolvimento do projeto.

*“Deixem que o futuro diga a verdade e avalie cada um
de acordo com o seu trabalho e realizações.”
(Nikola Tesla)*

Resumo

O estresse biótico consiste em danos causados a plantas por meio de outros seres vivos. O controle eficiente de agentes bióticos, tais como pragas e patógenos (vírus, fungos, bactérias, etc), está intimamente relacionado ao conceito de sustentabilidade agrícola. A sustentabilidade agrícola promove o desenvolvimento de novas tecnologias que possibilitam a redução de impactos ambientais, maior acessibilidade ao produtor rural e, conseqüentemente, o aumento da produtividade. O uso de visão computacional com métodos de aprendizado profundo permite a identificação do agente causador do estresse. Dessa forma, medidas corretivas podem ser aplicadas o quanto antes para mitigar o problema. Dentre as abordagens mais comumente utilizadas, as redes neurais convolucionais têm se destacado, mostrando resultados superiores e com altas taxas de acertos. O objetivo deste trabalho é projetar um sistema efetivo e prático capaz de classificar e estimar a severidade do estresse causado por agentes bióticos em folhas do cafeeiro. Neste trabalho, duas abordagens baseadas em redes neurais convolucionais foram investigadas. A primeira abordagem consiste em um sistema de etapa única treinado para classificar a base de imagens de folhas com os rótulos de estresse biótico e severidade. Já a segunda abordagem é formada por um sistema de duas etapas, a etapa de segmentação semântica, treinada com a base de máscaras de segmentação, e a etapa de classificação, treinada com a base de imagens de sintomas. As abordagens foram comparadas destacando-se os pontos positivos e negativos de cada uma. Além disso, foi explorado o uso do aprendizado multi-tarefa e de técnicas de *data augmentation* para tornar o sistema mais robusto e acurado. Os resultados experimentais obtiveram uma acurácia acima de 95% para a identificação do estresse e de 86% para a estimação da severidade usando a base de folhas. Ademais, verificou-se que, ao se classificar apenas os sintomas, os resultados foram superiores a 97%. A severidade calculada através das máscaras de segmentação mostraram um coeficiente de determinação de aproximadamente 0.98, sugerindo que o modelo consegue obter valores de severidade bem próximos dos valores reais. Os resultados obtidos indicam que o sistema proposto pode se tornar uma ferramenta viável para auxiliar tanto especialistas quanto agricultores na identificação e quantificação de estresses bióticos em plantações de café.

Palavras-chaves: Redes neurais convolucionais. Reconhecimento de padrões. Segmentação semântica. Estresse biótico. Severidade. Doenças e pragas em folhas de café.

Abstract

Biotic stress consists of damage to plants through other living organisms. Efficient control of biotic agents such as pests and pathogens (viruses, fungi, bacteria, etc.) is closely related to the concept of agricultural sustainability. Agricultural sustainability promotes the development of new technologies that allows the reduction of environmental impacts, greater accessibility to farmers and, consequently, increase on productivity. The use of computer vision with deep learning methods allows the identification of the stress-causing agent. So, corrective measures can be applied as soon as possible to mitigate the problem. Among the most commonly used approaches, convolutional neural networks present good results and high success rates. The goal of this work is to design an effective and practical system capable of identifying and estimating the stress severity caused by biotic agents on coffee leaves. In this work two approaches based on convolutional neural networks were investigated. The first approach consists of a one-step system trained to classify the leaf dataset with labels of biotic stress and severity. The second approach is formed by a two-step system, the semantic segmentation step, trained with the segmentation masks dataset, and the classification step, trained with the symptom dataset. The approaches were compared by highlighting their strengths and weaknesses. In addition, the use of multi-task learning and data augmentation techniques have been investigated in order to improve robustness and accuracy. The experimental results obtained an overall accuracy of 93% for biotic stress classification and 85% for severity estimation using the leaf dataset. Furthermore, it was found that by classifying only the symptoms, the accuracy was superior to 97%. The calculated severity using the segmentation masks showed a determination coefficient of around 0.98, suggesting that the model can obtain severity values very close to the real ones. The obtained results indicate that the proposed system might be a suitable tool to assist both experts and farmers in the identification and quantification of biotic stresses in coffee plantations.

Keywords: Convolutional neural networks. Pattern recognition. Semantic segmentation. Biotic stress. Severity. Coffee diseases and pests.

Lista de ilustrações

Figura 1 – Principais estresses bióticos que atingem o cafeeiro.	24
Figura 2 – Arquitetura pioneira de rede convolucional LeNet. Fonte: Adaptado de (LECUN et al., 1998).	30
Figura 3 – Exemplo de aplicação da operação de convolução.	30
Figura 4 – Exemplo de aplicação das operações de <i>pooling</i>	32
Figura 5 – Exemplo de aplicação da operação de <i>upsampling</i>	32
Figura 6 – Comparação entre as principais tarefas de visão computacional.	35
Figura 7 – Arquitetura da rede neural convolucional AlexNet. Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).	36
Figura 8 – Arquitetura da rede neural convolucional VGG16.	37
Figura 9 – Módulo <i>inception</i> . Fonte: (SZEGEDY et al., 2015).	37
Figura 10 – Bloco residual da ResNet. Fonte: Adaptado de (HE et al., 2016).	38
Figura 11 – Arquitetura de rede neural convolucional UNet. Fonte: (RONNEBERGER; FISCHER; BROX, 2015).	39
Figura 12 – Arquitetura de rede neural convolucional PSPNet. Fonte: (ZHAO et al., 2017).	39
Figura 13 – Visão geral de um sistema baseado em aprendizado profundo para classificação de doenças de plantas.	42
Figura 14 – <i>Framework</i> utilizado no aprendizado com tarefa única e multi-tarefa.	44
Figura 15 – Etapas para recorte e redimensionamento de imagens de folhas.	45
Figura 16 – Principais etapas que compõem a abordagem de segmentação e classificação de sintomas.	45
Figura 17 – Exemplos de folhas de café afetadas por diferentes estresses bióticos: bicho mineiro (a), ferrugem (b), mancha-de-phoma (c) e cercosporiose (d).	48
Figura 18 – Exemplos de imagens de sintomas.	49
Figura 19 – Exemplos de máscaras de segmentação.	50
Figura 20 – Matriz de confusão com os resultados de predição usando a arquitetura ResNet50 para a base de folhas.	54
Figura 21 – Matriz de confusão com os resultados de predição usando a arquitetura ResNet50 para a base de sintomas.	55
Figura 22 – Visualização da disposição espacial das características extraídas pela ResNet50 para as amostras de teste da base de folhas usando o t-SNE.	56
Figura 23 – Visualização da disposição espacial das características extraídas pela ResNet50 para as amostras de teste da base de sintomas usando o t-SNE.	57
Figura 24 – Gráfico de dispersão da severidade real versus severidade predita.	58
Figura 25 – Resultados visuais de segmentação de algumas imagens da base de teste.	59

Figura 26 – Mariposa (A); danos causados pelo bicho mineiro (B); e casulo em forma de X (C). Fonte: Adaptado de (FORNAZIER et al., 2017).	70
Figura 27 – Pústula de ferrugem na face abaxial da folha de café. Fonte: Adaptado de (VENTURA; COSTA; LIMA, 2017).	71
Figura 28 – Sintomas de mancha-de-phoma em mudas de café. Fonte: (VENTURA; COSTA; LIMA, 2017).	72
Figura 29 – Sintomas de cercosporiose em plantas adultas de café conilon. Fonte: Adaptado de (VENTURA; COSTA; LIMA, 2017).	72
Figura 30 – Etapas de processamento do sistema que permite o acesso às previsões dos modelos de <i>machine learning</i> pelo aplicativo.	75
Figura 31 – Telas capturadas durante o uso da aplicação: login (A); menu (B); e biblioteca (C).	77
Figura 32 – Telas capturadas durante o uso da aplicação: câmera (A); verificação e <i>upload</i> (B); e resultado (C).	77

Lista de tabelas

Tabela 1 – Quantidade de amostras por classe da base de folhas.	48
Tabela 2 – Quantidade de amostras por classe da base de sintomas.	49
Tabela 3 – Quantidade de amostras por classe da base de segmentação.	50
Tabela 4 – Hiper-parâmetros de treinamento da CNN.	51
Tabela 5 – Resultados de teste obtidos com diferentes arquiteturas para a base de folhas.	53
Tabela 6 – Quantidade de parâmetros e tempo médio de treinamento e execução das arquiteturas de classificação investigadas.	54
Tabela 7 – Resultados de teste obtidos com diferentes arquiteturas para a base de sintomas.	55
Tabela 8 – Resultados de teste obtidos com UNet e PSPNet para a base de segmentação.	58
Tabela 9 – Quantidade de parâmetros e tempo médio de treinamento e execução época das arquiteturas de segmentação investigadas.	60
Tabela 10 – Resumo com os melhores resultados experimentais para ambas as abordagens propostas.	60

Lista de abreviaturas e siglas

MTL	<i>Multi-task learning</i>
CNN	<i>Convolutional Neural Network</i>
ReLU	<i>Rectified Linear Unit</i>
ILSVRC	<i>ImageNet Large Scale Visual Recognition Competition</i>
VGG	<i>Visual Geometry Group</i>
ResNet	<i>Residual Network</i>
PSPNet	<i>Pyramid Scene Parsing Network</i>
AC	<i>Accuracy</i>
PR	<i>Precision</i>
RE	<i>Recall</i>
MIoU	<i>Mean Intersection over Union</i>
t-SNE	<i>t-Distributed Stochastic Neighbor Embedding</i>

Sumário

1	INTRODUÇÃO	23
1.1	Motivação	24
1.2	Revisão de literatura	25
1.3	Objetivos	28
1.4	Estrutura da dissertação	28
2	REDES NEURAS CONVOLUCIONAIS	29
2.1	Componentes principais	29
2.1.1	Convolução	29
2.1.2	<i>Pooling</i>	31
2.1.3	<i>Upsampling</i>	31
2.1.4	Camada totalmente conectada	32
2.1.5	Técnicas de regularização	33
2.1.5.1	Decaimento do peso	33
2.1.5.2	<i>Dropout</i>	33
2.1.5.3	<i>Data augmentation</i>	33
2.2	<i>Transfer learning</i>	34
2.3	Visão computacional	35
2.4	Arquiteturas para classificação	35
2.4.1	AlexNet	36
2.4.2	VGG	36
2.4.3	GoogLeNet	36
2.4.4	ResNet	37
2.5	Arquiteturas para segmentação semântica	38
2.5.1	UNet	38
2.5.2	PSPNet	39
3	MÉTODOS BASEADOS EM REDE NEURAL CONVOLUCIONAL PARA CLASSIFICAÇÃO E SEGMENTAÇÃO	41
3.1	Visão geral da abordagem proposta	41
3.1.1	Preparação dos dados	41
3.1.2	Treinamento	43
3.1.3	Implantação	43
3.2	Classificação de folhas	44
3.3	Segmentação e classificação de sintomas	45

4	RESULTADOS EXPERIMENTAIS	47
4.1	Bases de dados	47
4.1.1	Base de folhas	47
4.1.2	Base de sintomas	49
4.1.3	Base de segmentação	49
4.2	Configuração experimental	50
4.3	Métricas	51
4.4	Classificação	52
4.4.1	Folhas	53
4.4.2	Sintomas	55
4.4.3	Visualização	56
4.5	Segmentação semântica	57
4.6	Comparativo entre abordagens	60
5	CONCLUSÃO	61
	REFERÊNCIAS	63
	APÊNDICES	67
	APÊNDICE A – PRAGAS E DOENÇAS DO CAFEIRO	69
A.1	Pragas	69
A.1.1	Bicho mineiro	69
A.1.2	Outras pragas	70
A.2	Doenças	70
A.2.1	Ferrugem	71
A.2.2	Mancha-de-phoma	71
A.2.3	Cercosporiose	72
A.2.4	Outras doenças	72
	APÊNDICE B – APLICATIVO	75
B.1	Funcionalidades	76

1 Introdução

Plantas estão constantemente expostas a uma ampla diversidade de agentes bióticos, tais como pragas e patógenos (insetos, vírus, fungos, bactérias, etc.), e também por fatores abióticos, como déficit hídrico, calor, salinidade e frio (SUZUKI *et al.*, 2014). Nesse contexto, é importante considerar que doenças de plantas causadas por patógenos levam a um comprometimento do estado normal de uma planta, interrompendo e modificando suas funções vitais. Por isso, apresentam-se como um fator limitante na produtividade e levam a perdas significativas na colheita (VENTURA; COSTA; LIMA, 2017). Além disso, doenças de plantas são uma ameaça à segurança alimentar em escala global e afetam principalmente pequenos agricultores, cuja subsistência depende de culturas saudáveis (MOHANTY; HUGHES; SALATHÉ, 2016).

Existem muitas medidas que podem ser tomadas para evitar a disseminação de pragas e doenças nas plantações. Nesse sentido, o manejo integrado de pragas e doenças aplicado de maneira antecipada e controlada reduz as chances de perdas na produção e também a necessidade do uso de agrotóxicos. O uso indiscriminado de agrotóxicos levam a perdas econômicas com a compra e aplicação excessiva, despesas relacionadas a tratamento médico com pessoas eventualmente envenenadas por agrotóxicos e danos causados pela contaminação ambiental (OLIVEIRA *et al.*, 2014). O controle eficiente de tais agentes bióticos está intimamente relacionado ao conceito de sustentabilidade agrícola, cujo objetivo é promover o desenvolvimento de novas tecnologias e práticas que reduzem os impactos ambientais, sejam mais acessíveis ao produtor rural e aumentem a sua produtividade (PRETTY, 2007).

Para um controle eficiente de pragas e doenças é importante que se conheça não só o agente causador, mas também a severidade das doenças, uma vez que o diagnóstico e a quantificação do estresse de plantas são duas funções igualmente importantes para a fitopatologia (KRANZ, 1988). A severidade das doenças mede a porcentagem da área de tecido foliar doente e é importante para prever rendimento da lavoura e para a recomendação de tratamentos (BOCK *et al.*, 2010).

A identificação de doenças de plantas é uma tarefa tão importante quanto desafiadora até mesmo para nós humanos com nossa notável capacidade de reconhecer e interpretar padrões (BARBEDO; KOENIGKAN; SANTOS, 2016). A avaliação visual de sintomas presentes em folhas de plantas, normalmente realizadas por especialistas, possui um alto caráter subjetivo devido à similaridade das doenças e à diversidade de variações em suas características que se manifestam influenciadas por fatores externos.

Muitos produtores rurais não possuem acesso a tais especialistas, o que torna a

identificação de doenças de plantas um desafio. Como consequência, o controle efetuado tende a ser pouco eficiente e a planta fica vulnerável a perdas de produção.

1.1 Motivação

Dentre os setores da economia Brasileira, a agricultura é um dos principais pilares, gerando trabalho e renda para o país (OLIVEIRA et al., 2014). De acordo com a *International Coffee Organization* (2019), o Brasil é o maior produtor de café do mundo sendo esta uma cultura importante para o país. Um dos obstáculos enfrentados pelos agricultores são as pragas e doenças que atingem as plantações de café, causando desfolha e redução da fotossíntese, o que, conseqüentemente, compromete a produção e a qualidade do produto final (VENTURA; COSTA; LIMA, 2017). Dentre as principais pragas e doenças que afetam o cafeeiro, podemos destacar: bicho mineiro, ferrugem, mancha-de-phoma e cercosporiose. Exemplos de sintomas causados por estes estresses podem ser visualizados na Figura 1.

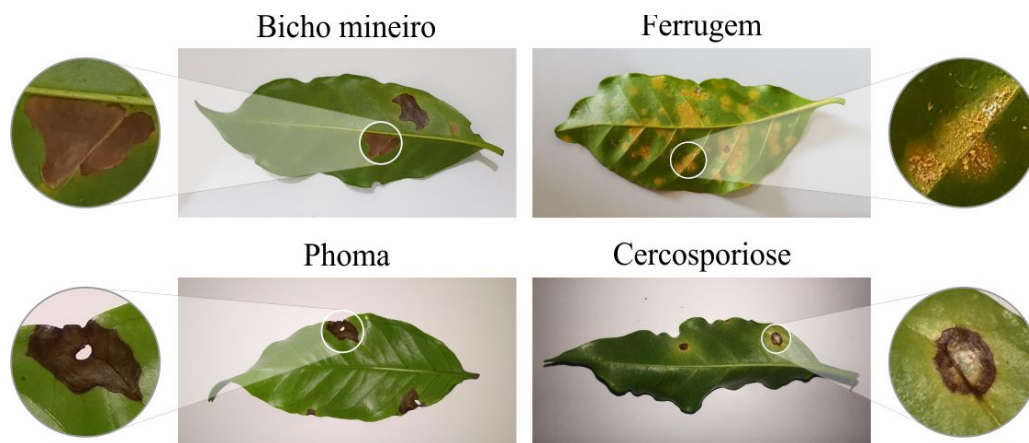


Figura 1 – Principais estresses bióticos que atingem o cafeeiro.

De acordo com o Instituto Capixaba de Pesquisa, Assistência Técnica e Extensão Rural (Incaper), o estado do Espírito Santo é 2º maior produtor de café do país. A sua expressiva produção de ambas as variedades de café, arábica e conilon, o tornam responsável por 22% da produção nacional. Dentre os produtores de café, 73% são de base familiar.

Vários esforços têm sido desenvolvidos utilizando Inteligência Artificial para auxiliar pequenos agricultores a identificarem corretamente doenças e pragas que afetam sua produção e a severidade das lesões foliares. Sistemas de diagnóstico auxiliados por computador permitem que qualquer agricultor que tenha acesso a um *smartphone* possa usufruir de conhecimento especializado de forma prática e de baixo custo (DEHNEN-SCHMUTZ et al., 2016).

Embora diversos trabalhos tenham sido propostos na literatura para o reconhecimento de doenças de folhas, poucos deles possuem aplicação direta à cultura do café, como

exemplo podemos citar os trabalhos de [Manso et al. \(2019\)](#) e [Barbedo \(2019\)](#). No entanto, nenhum trabalho na literatura soluciona o problema de forma integral, todos apresentam limitações. É neste sentido que abrem-se caminhos de pesquisa nesta área, em especial para a cultura do café que possui forte impacto na economia estadual.

1.2 Revisão de literatura

A Inteligência Artificial tem mostrado um crescimento exponencial nos últimos anos com respeito às aplicações de Aprendizado de Máquina, o que levou ao surgimento de uma nova categoria de modelos intitulada Aprendizado Profundo ([LECUN; BENGIO; HINTON, 2015](#)). Métodos de Aprendizado Profundo têm mostrado clara superioridade em relação às abordagens tradicionais de Aprendizado de Máquina na solução de diversos problemas. Dentre os métodos de Aprendizado Profundo, temos as Redes Neurais Convolucionais (CNN) que têm apresentado excelentes resultados nas tarefas de reconhecimento de imagens ([FUENTES et al., 2017](#)). Diferentemente das abordagens tradicionais, a CNN aprende automaticamente características apropriadas da base de treino. Além disso, a etapa de segmentação está intrinsecamente ligada aos filtros de convolução de uma CNN, simplificando ainda mais o seu uso.

De acordo com [Barbedo \(2018\)](#), CNN's são ferramentas poderosas e que lidam bem com o problema de classificação de doenças e pragas de plantas. No entanto, ainda existem muitos desafios associados ao diagnóstico automático de pragas e doenças em plantas. Em [Barbedo \(2016\)](#) são abordados os 6 principais desafios encontrados neste tipo de problema, estes desafios foram separados em fatores extrínsecos e intrínsecos. Fatores extrínsecos estão relacionados principalmente a coleta de imagens em campo, sendo eles: (1) *backgrounds* complexos, por exemplo, folhas, solo, pedras etc. (2) condições de captura de imagens não controladas, ou seja, variação da iluminação, sombras, reflexos e desfoque. Fatores intrínsecos: (3) limites das lesões mal definidos, possuem transição suave entre a lesão e o tecido sadio. (4) Variação dos sintomas para diferentes estágios das doenças, onde a interação entre planta, doença e ambiente leva a variações nas suas características visuais. (5) Múltiplas desordens simultâneas de origem biótica e abiótica podem se manifestar na mesma planta e até na mesma folha. (6) Similaridades na aparência dos sintomas provocados por diferentes tipos de estresse.

A superação de todos os desafios apontados em [Barbedo \(2016\)](#) estão em algum grau relacionados à necessidade de surgimento de novos métodos que satisfaçam os requisitos de confiabilidade e acurácia ([SINGH et al., 2018](#)). Os desafios (4) e (5) dependem fortemente da construção de bases de dados bem representativas e confiáveis. De acordo com [Barbedo \(2019\)](#), este é o principal problema encontrado por abordagens de aprendizado profundo na identificação automática de doenças e pragas de plantas. A criação destas bases de

dados não é uma tarefa simples, posto que, rotular as imagens é um processo custoso e que muitas vezes deve ser realizado por um especialista. Apesar de existirem iniciativas que estão utilizando o conceito de redes sociais para acelerar este processo, esta prática pode levar a dados não confiáveis (BARBEDO, 2018).

A maior base de dados pública atualmente, chamada PlantVillage, foi desenvolvida por Hughes e Salathe (2015) e contém mais de 54 mil imagens de folhas e 38 classes, dentre elas folhas saudáveis e doentes de diferentes espécies. Vários estudos foram realizados utilizando a base de dados PlantVillage (MOHANTY; HUGHES; SALATHÉ, 2016; TOO et al., 2019; KAYA et al., 2019), em sua maioria estes estudos apresentaram resultados superiores a 99% de acurácia. Além disso, é comum na literatura trabalhos que desenvolveram a sua própria base de dados aplicado a um tipo específico de cultura, como Fuentes et al. (2017), que coletaram fotos de folhas do tomateiro utilizando câmeras comuns, Johannes et al. (2017), que desenvolveram um sistema capaz de identificar doenças em fotos de folhas de trigo obtidas por smartphones, Liu et al. (2017), que produziram uma base de imagens de folhas de maçã e propuseram uma nova arquitetura baseada na AlexNet para o reconhecimento das doenças, e Ma et al. (2018), que montaram uma base com imagens de sintomas de doenças comuns que atingem folhas do pepineiro.

Embora o problema de detecção de doenças em folhas de plantas tenha sido abordado em diversos estudos, poucos deles focaram em desenvolver sistemas capazes de estimar a severidade das lesões. Wang, Sun e Wang (2017) propuseram o uso de redes convolucionais para estimar a severidade de doenças em plantas. Eles anotaram as imagens de maçãs, da base PlantVillage, afetadas por podridão negra em quatro graus de severidade. Os resultados experimentais apresentaram acurácia superior a 90%.

Tipicamente em Aprendizado de Máquina, nos concentramos em resolver um único problema, tal como diagnosticar doenças em folhas ou então estimar a sua severidade. Para fazer isto, geralmente um modelo ou um conjunto de modelos são treinados para resolver uma tarefa específica. Por outro lado, existe o conceito de Aprendizado Multi-Tarefa (do inglês: *Multi-task learning*, abreviado por MTL) que consiste em treinar um sistema capaz de resolver múltiplas tarefas utilizando uma arquitetura compartilhada. O MTL leva o modelo a preferir representações que sirvam para ambas as tarefas, de tal forma que as camadas iniciais da rede vão aprender representações conjuntas generalizadas, possibilitando uma melhor capacidade de generalização do modelo, prevenindo o *overfitting* (RUDER, 2017).

Liang et al. (2019) propuseram um sistema de diagnóstico auxiliado por computador multi-tarefa capaz de realizar diagnóstico de doenças, reconhecer a espécie da planta e estimar a severidade das doenças, chamado PD²SE-Net. Para a realização dos experimentos foi utilizada a base de dados PlantVillage. A estimação da severidade das doenças consistiu em classificar as folhas em uma das três classes: saudável, média e séria. Os resultados

apresentaram uma acurácia geral de 91% e 98% para estimação de severidade e classificação das doenças.

[Ghosal et al. \(2018\)](#) desenvolveram um framework baseado em CNN capaz de identificar e classificar um conjunto de 8 estresses (bióticos e abióticos) que acometem as folhas de soja. Além disso, apresentaram um mecanismo para selecionar e extrair os mapas de características da rede que melhor isolam os sintomas visuais de maneira similar a realizada por humanos e utilizaram tais mapas para quantificar a severidade do estresse.

[Barbedo \(2019\)](#) explorou o uso de lesões individuais nas folhas através da segmentação manual, isto permitiu aumentar a base de dados significativamente e identificar múltiplas lesões que afetam a mesma folha, em contrapartida a segmentação manual é um processo trabalhoso principalmente em folhas com muitas lesões pequenas.

Outros trabalhos vêm explorando o uso de métodos de segmentação automáticos como etapa primária na classificação e quantificação de doenças de plantas. Uma vez que a segmentação permite isolar os sintomas das folhas originais, isto facilita a classificação do estresse e a medição da área lesionada.

[Manso et al. \(2019\)](#) desenvolveram um sistema capaz de classificar os sintomas em folhas do cafeeiro, provocados por pragas e doenças, de forma que a segmentação dos sintomas foi realizada utilizando um método não supervisionado baseado em limiar. As características foram calculadas a partir da imagem segmentada e os sintomas foram classificados em duas classes: bicho mineiro e ferrugem. A partir das máscaras de segmentação foi possível quantificar a severidade das doenças e identificar cada sintoma individualmente. Muito embora o trabalho tenha apresentado bons resultados, a segmentação com este tipo de abordagem não leva em consideração a relação posicional dos pixels, podendo ser facilmente afetada por fatores como a variação de iluminação e reflexão especular.

[Hu et al. \(2019\)](#) apresentaram uma abordagem capaz de identificar doenças em folhas de chá-da-índia. A segmentação dos sintomas foi realizada por um método supervisionado onde características de cor e textura foram extraídas e utilizadas no treinamento de uma máquina de vetores de suporte. Novas amostras foram geradas a partir das imagens dos sintomas segmentados usando uma rede convolucional generativa adversarial e posteriormente classificadas pela arquitetura VGG16. Os resultados mostraram uma acurácia média de 90% na identificação das doenças.

Recentemente, [Lin et al. \(2019\)](#) desenvolveram um sistema capaz de segmentar os sintomas de oídio (doença causada por fungos) em folhas de pepino utilizando uma rede convolucional UNet. A segmentação dos sintomas permite calcular de forma precisa a severidade da doença. Um total de 50 imagens foram anotadas e utilizadas nos experimentos que obtiveram resultados superiores quando comparados a métodos tradicionais da literatura.

1.3 Objetivos

Este trabalho estende significativamente o trabalho de [Manso et al. \(2019\)](#) a fim de superar as limitações dos trabalhos existentes e de métodos convencionais de processamento de imagens. Portanto, os objetivos principais deste trabalho são:

1. Construir uma nova base de imagens de folhas com as principais pragas e doenças que atingem o cafeeiro.
2. Desenvolver um sistema totalmente automático capaz de classificar e quantificar estresses bióticos em imagens de folhas de café utilizando redes neurais convolucionais.

1.4 Estrutura da dissertação

- No capítulo 2, é apresentada uma fundamentação teórica sobre redes neurais convolucionais, com uma breve explicação sobre o funcionamento dos principais componentes e a descrição das arquiteturas utilizadas. Este capítulo serve como base para entendimento dos métodos propostos e experimentos realizados.
- No capítulo 3, é apresentada uma visão geral da sistema proposto e detalhes de ambas as abordagens investigadas.
- No capítulo 4, são apresentados os resultados e discussões dos experimentos computacionais.
- No capítulo 5, o conhecimento relevante obtido é sumarizado e direções para trabalhos futuros são apontadas.

2 Redes neurais convolucionais

Abordagens tradicionais para classificação de imagens baseiam-se em uma metodologia de dois estágios. Inicialmente, as características são extraídas usando descritores de imagens e posteriormente servem como entrada para um classificador (RAWAT; WANG, 2017). A maior dificuldade deste tipo de abordagem está em decidir quais características são apropriadas para cada problema. CNN's surgem como uma alternativa promissora, fornecendo um aprendizado automático das características de imagens em diferentes domínios (ALOYSIUS; GEETHA, 2017).

As redes CNN são uma variação das redes neurais *perceptron* multicamadas e foram inspiradas nos estudos da biologia do córtex visual dos animais realizados por Hubel e Wiesel (1968). Elas seguem o padrão de redes *feedforwards*, ou seja, os dados seguem um fluxo de apenas um sentido, da entrada até a saída da rede. Uma CNN pode ser resumida como uma rede que utiliza a operação de convolução no lugar de multiplicação de matrizes em ao menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Este capítulo faz uma revisão sobre as principais camadas que compõem uma CNN, que é a classe de redes neurais no qual toda a metodologia apresentada neste trabalho se baseia. Além disso, é abordado o funcionamento de cada arquitetura utilizada para a realização dos experimentos.

2.1 Componentes principais

De acordo com Lecun et al. (1998), as CNN's são capazes de aprender representações invariantes a transformações geométricas. Esta característica é obtida porque a rede impõe um padrão de conectividade local e compartilhamento dos pesos. Na Figura 2 é apresentada a LeNet, uma das arquiteturas pioneiras de CNN proposta por Lecun et al. (1998). Assim como a LeNet, as arquiteturas de CNN's são formadas por operadores com diferentes propósitos que compõem as suas camadas. As subseções a seguir descrevem os diferentes tipos de operações e camadas que compõem a maioria das redes CNN.

2.1.1 Convolução

A operação de convolução constitui o alicerce das redes CNN. As convoluções servem como extratores de características, dessa forma elas aprendem representações das características das imagens de entrada (RAWAT; WANG, 2017). Matematicamente, convolução é um operador linear que a partir de duas funções produz uma terceira. Os

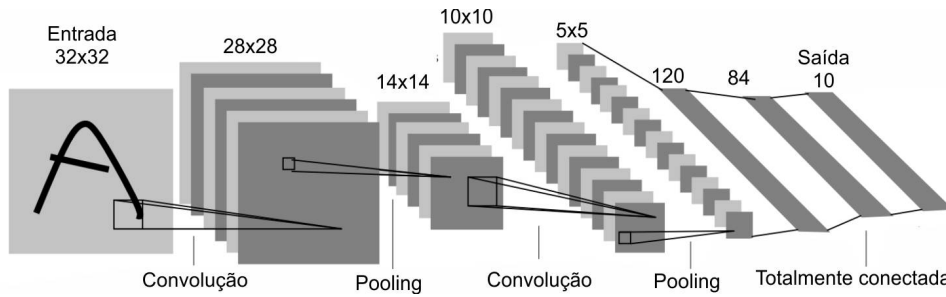


Figura 2 – Arquitetura pioneira de rede convolucional LeNet. Fonte: Adaptado de (LECUN et al., 1998).

neurônios em uma rede neural convolucional são organizados como mapas de ativação. Os parâmetros da camada de convolução são um conjunto de filtros treináveis que são convoluídos sobre os mapas de ativação produzindo um novo mapa cuja profundidade depende do número de filtros utilizados. As regiões dos mapas de ativação processadas pelos filtros são chamadas de campos receptivos locais. Além disso, as CNN's carregam o conceito de compartilhamento de pesos, uma vez que os pesos de um filtro são utilizados em todo o mapa de ativação, isso leva a uma redução substancial da quantidade de parâmetros da rede.

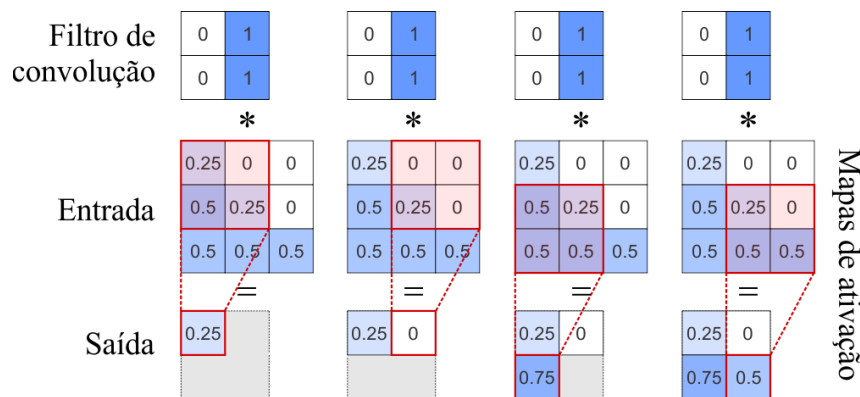


Figura 3 – Exemplo de aplicação da operação de convolução.

A Figura 3 ilustra a operação de convolução com um exemplo simples. Dada uma entrada (uma imagem ou um mapa de ativação resultante de uma convolução anterior) e um filtro, a convolução consiste em deslocar o filtro sobre a entrada de tal forma que os elementos do filtro são multiplicados pelos elementos do campo receptivo local (quadrados de contorno vermelho). Por fim, os elementos são somados produzindo o valor correspondente do mapa de ativação de saída.

Há dois parâmetros que impactam diretamente no tamanho dos mapas de ativação resultantes da operação de convolução, são eles: *stride* e *padding*. O *stride* consiste no tamanho do deslocamento dos campos receptivos locais, normalmente recebe o valor 1. Quanto maior o stride menor o número de unidades de saída. Já o parâmetro *padding* adiciona borda de valores 0 à imagem de entrada, o controle da espessura da borda permite

obter mapas de ativação de tamanhos específicos na saída.

Após aplicada a operação de convolução, os mapas de ativação de saída passam por uma função de ativação, cujo papel, é introduzir uma componente não linear à rede neural, permitindo que a rede possa resolver problemas mais complexos do que apenas relações lineares. A função de ativação mais utilizada em CNN's é a unidade linear retificada (do inglês: *Rectified Linear Units*, abreviado por ReLU). A ReLU é aplicada a cada unidade do mapa de ativação de maneira a retificar os valores negativos e é descrita pela seguinte equação:

$$\text{ReLU}(x) = \max(0, x) \quad (2.1)$$

Devido ao seu baixo custo computacional e excelentes resultados, a ReLU tem sido amplamente utilizada para o desenvolvimento de sistemas baseados em aprendizado profundo. Todavia, diversas outras funções de ativação têm sido propostas na literatura como, *leaky ReLU* (MAAS; HANNUN; NG, 2013), *randomized ReLU* (XU et al., 2015), *parametric ReLU* (HE et al., 2015) e *exponential linear units* (abreviado por ELU) (CLEVERT; UNTERTHINER; HOCHREITER, 2015). Neste trabalho todos os experimentos limitaram-se ao uso da função ReLU.

2.1.2 Pooling

O objetivo principal da operação de *pooling* é transformar um conjunto de características em um novo conjunto menos susceptível à transformações da imagem, de tal forma que informações importantes sejam preservadas enquanto descarta detalhes irrelevantes (BOUREAU; PONCE; LECUN, 2010). Conseqüentemente, a operação de *pooling* diminui a carga computacional reduzindo o número de conexões entre as redes neurais convolucionais. A operação de *pooling* age similar a de convolução, com a diferença de que possui um *stride* com o mesmo tamanho do filtro. Os tipos de *pooling* mais comuns e utilizados neste trabalho são: *max pooling* e *average pooling*. O *max pooling* propaga o maior valor dentro do campo receptivo e o *average pooling* propaga a média dos valores. Na Figura 4, é apresentado um exemplo comparativo entre o *max pooling* e o *average pooling*. Uma entrada de tamanho 4×4 é submetida a operação de *max pooling* e *average pooling* com um filtro 2×2 . Como resultado, temos um mapa de ativação com a metade da largura e altura do mapa de entrada.

2.1.3 Upsampling

Long, Shelhamer e Darrell (2015) introduziram o uso das camadas de *upsampling* para segmentação semântica no qual eles comparam esta operação como sendo uma convo-

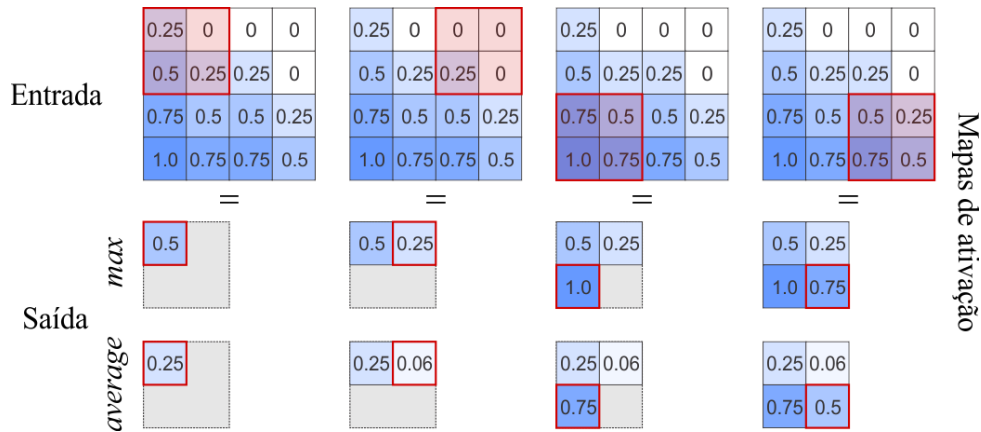


Figura 4 – Exemplo de aplicação das operações de *pooling*.

lução transposta. Esta operação visa transformar mapas de ativação de baixa dimensão em mapas com maior dimensão. Os filtros da operação de *upsampling* podem ser fixos, por exemplo um filtro que produza uma interpolação bilinear, ou podem ser adaptáveis. Neste caso, os filtros são tratados como pesos da mesma maneira que os filtros da camada de convolução e serão ajustados durante a etapa de treinamento da rede. Na Figura 5 é apresentado um exemplo de aplicação da operação de *upsampling* onde cada elemento do mapa de características de entrada é multiplicado pelo filtro de convolução transposta produzindo uma saída de maior dimensionalidade.

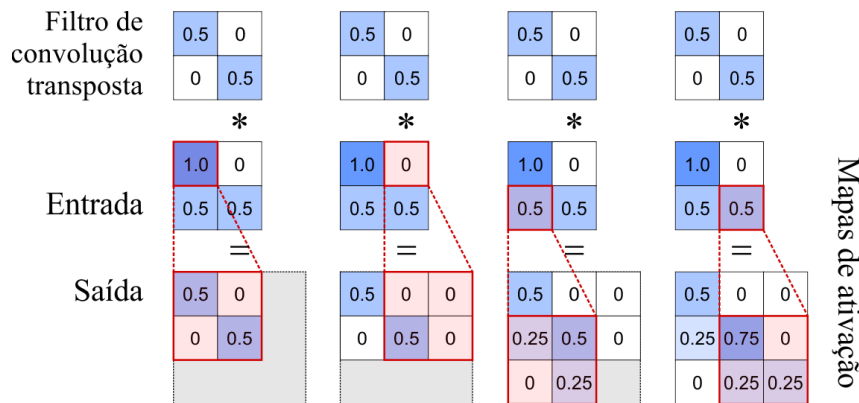


Figura 5 – Exemplo de aplicação da operação de *upsampling*.

2.1.4 Camada totalmente conectada

Esta camada consiste em neurônios totalmente conectados aos neurônios da camada anterior, com os neurônios organizados em uma única dimensão (ALOYSIUS; GEETHA, 2017). Atua como uma rede neural multicamadas comum e contém aproximadamente 90% dos parâmetros de uma CNN (GUO et al., 2016). O papel da camada totalmente conectada é interpretar as características não lineares e de alto nível extraídas pelas camadas de convolução e *pooling*.

2.1.5 Técnicas de regularização

Uma generalização ruim ocorre quando a informação disponível não corresponde a complexidade do modelo (KROGH; HERTZ, 1991). Logo, modelos complexos tendem a sofrer *overfitting* quando treinados com bases de dados pequenas. Para superar este problema, são utilizadas técnicas de regularização. Neste trabalho foram utilizadas três técnicas comuns e que atuam em diferentes partes da rede durante seu treinamento. Detalhes das principais técnicas são descritas a seguir.

2.1.5.1 Decaimento do peso

A técnica de regularização por decaimento de peso, também conhecida por regularização L2, consiste em limitar o crescimento dos pesos da rede pela adição de uma penalização à função de custo. Como consequência, provoca uma restrição a rede reduzindo a sua complexidade (KROGH; HERTZ, 1991). A penalização é calculada pela seguinte equação:

$$E(\mathbf{w}) = E_0(\mathbf{w}) + \frac{1}{2}\lambda \sum_i w_i^2 \quad (2.2)$$

onde E_0 representa a função de perda, λ é o fator de decaimento do peso e \mathbf{w} é o vetor contendo os parâmetros treináveis da rede, ou seja, o conjunto de pesos.

2.1.5.2 Dropout

A técnica de *dropout* atua na modificação das conexões da própria rede, proporcionando uma maneira eficiente de se treinar uma quantidade exponencial de redes neurais menores que compartilham os pesos. O seu funcionamento consiste em omitir temporariamente neurônios aleatórios da rede com uma dada probabilidade p durante a etapa de treinamento. Já a etapa de teste é realizada sem o *dropout*, sendo assim, as unidades que foram retidas durante o treinamento têm agora seus pesos multiplicados pelo fator p . A capacidade de generalização desta técnica tem se mostrado superior a outras técnicas de regularização da literatura (SRIVASTAVA et al., 2014).

2.1.5.3 Data augmentation

A técnica *data augmentation* busca aumentar a capacidade de generalização do modelo através da geração de exemplares sintéticos dos dados de treinamento. Neste trabalho foram empregadas técnicas padrões de *data augmentation* e foi explorado o uso de uma nova abordagem que combina duas imagens selecionadas aleatoriamente gerando uma nova. Detalhes de cada abordagem utilizada são apresentados a seguir.

Padrão: As técnicas padrões de *data augmentation* são aquelas de uso comum na maioria dos trabalhos e apresentam bons ganhos de performance nos modelos. Embora existam várias maneiras de se gerar novas imagens a partir das amostras de treino, nós focamos apenas nas mais comuns, sendo elas: espelhamento (horizontal e vertical), rotação e variação de cor (brilho, contraste e saturação). Os aumentos padrões foram aplicados a todos os experimentos.

Mixup: Esta abordagem foi proposta por [Zhang et al. \(2017\)](#) e consiste de uma simples combinação linear de duas imagens e de seus rótulos. Portanto, uma nova imagem é gerada por $\{\tilde{x}, \tilde{y}\} = \lambda\{x_i, y_i\} + (1 - \lambda)\{x_j, y_j\}$ onde $\lambda \in [0, 1]$ é um valor aleatório gerado pela distribuição de probabilidade Beta, x representa o conjunto de imagens e y o conjunto de rótulos codificados no formato *one-hot*. A codificação *one-hot* consiste em atribuir valores binários às variáveis categóricas, de tal forma que o número de bits é igual a número de classes e cada bit representa uma classe. Por exemplo, um sistema que reconhece gatos e cachorros, poderia atribuir os seguintes valores aos rótulos: gato = 01 e cachorro = 10.

2.2 Transfer learning

De acordo com [Pan e Yang \(2010\)](#), a técnica de *transfer learning* consiste em utilizar o conhecimento a priori obtido na resolução de um determinado problema (tarefa fonte) para resolver outro problema (tarefa destino).

O uso da técnica de *transfer learning* em aprendizado de máquina consiste em reaproveitar modelos treinados anteriormente em outras tarefas para facilitar o aprendizado de tarefas semelhantes ([KAYA et al., 2019](#)). O reuso de modelos pré-treinados torna o aprendizado mais rápido e pode melhorar o desempenho do sistema, em especial para problemas que possuem poucos dados de treinamento.

A abordagem mais comum de *transfer learning* aplicada em modelos de aprendizado profundo consiste em realizar um ajuste fino dos pesos de modelos pré-treinados. Os modelos pré-treinados em tarefas fonte são utilizados como ponto de partida para a tarefa destino, de tal forma que o seu treinamento pode ser parcial, ou seja, parte dos pesos da rede são congelados e apenas uma parte é retreinada, ou pode ser total onde todos os pesos passam a serem adaptados para a nova tarefa. Em ambos os casos é comum a troca da última camada para que o número de classes seja adequado a tarefa destino, neste caso, esta camada deverá obrigatoriamente ser treinada junto com as outras.

2.3 Visão computacional

As redes convolucionais foram inicialmente propostas para classificação de imagens, no entanto, sua aplicação foi expandida para outros problemas, mostrando que são capazes de resolver diferentes tarefas de visão computacional. A partir dos modelos originais, novos *frameworks* foram derivados para resolver tarefas como a detecção de objetos e a segmentação semântica (GUO et al., 2016). Na Figura 6 são comparadas as três tarefas mais comuns de visão computacional e uma breve descrição de cada uma delas é apresentada a seguir.

- Classificação de imagens: a rede rotula as imagens de entrada com a probabilidade da presença de cada classe de objetos para o qual foi treinada. Logo, a imagem é atribuída à classe de maior probabilidade.
- Detecção de objetos: Combina as tarefas de localização e classificação em um único *framework*. A localização consiste em identificar um ou mais objetos em uma imagem e desenhar uma caixa delimitadora em torno deles. Após a detecção dos objetos é atribuído um rótulo para cada um individualmente.
- Segmentação semântica: Diferentemente da classificação de imagens e detecção de objetos, a segmentação semântica realiza previsões a nível de pixel, isto é, agrupa pixels da imagem que pertencem a mesma classe do objeto.

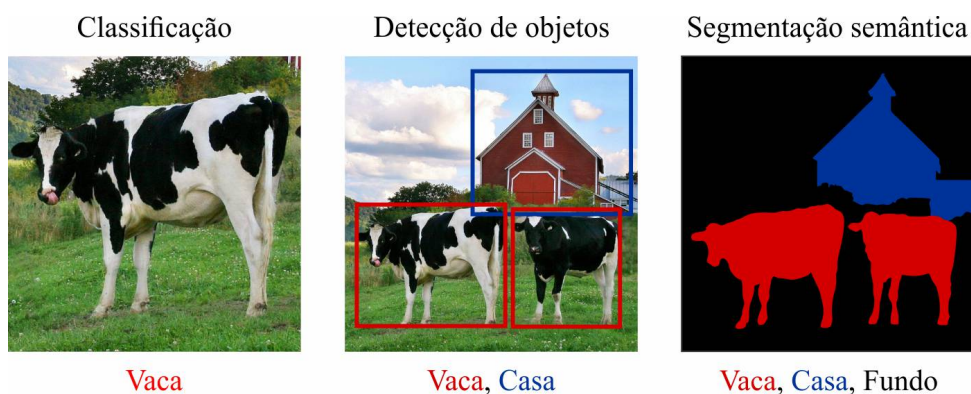


Figura 6 – Comparação entre as principais tarefas de visão computacional.

2.4 Arquiteturas para classificação

Certas arquiteturas de redes convolucionais possibilitaram contribuições significativas na área de classificação de imagens, tornando o seu uso padrão em diversos trabalhos. É o caso da AlexNet, VGG, GoogLeNet e a ResNet (GARCIA-GARCIA et al., 2017). Esta sessão é devotada a revisar as características de cada uma delas.

2.4.1 AlexNet

O trabalho desenvolvido por [Krizhevsky, Sutskever e Hinton \(2012\)](#) é um dos mais influentes a respeito de arquiteturas de CNN. A arquitetura proposta, AlexNet, foi uma das primeiras a obter resultados realmente significativos em classificação de imagens. Foi vencedora da competição ILSVRC-2012, com uma taxa de erro top-5 de 15.3%. O objetivo da competição era classificar milhões de imagens dentro de um grupo de 1000 classes. O modelo proposto contém 5 camadas de convolução e 3 camadas totalmente conectadas. A função de ativação ReLU foi utilizada na saída de todas as camadas da rede. Além disso, foram utilizadas as técnicas de *max pooling* e *dropout* entre determinadas camadas. Detalhes da arquitetura podem ser visualizados na Figura 7.

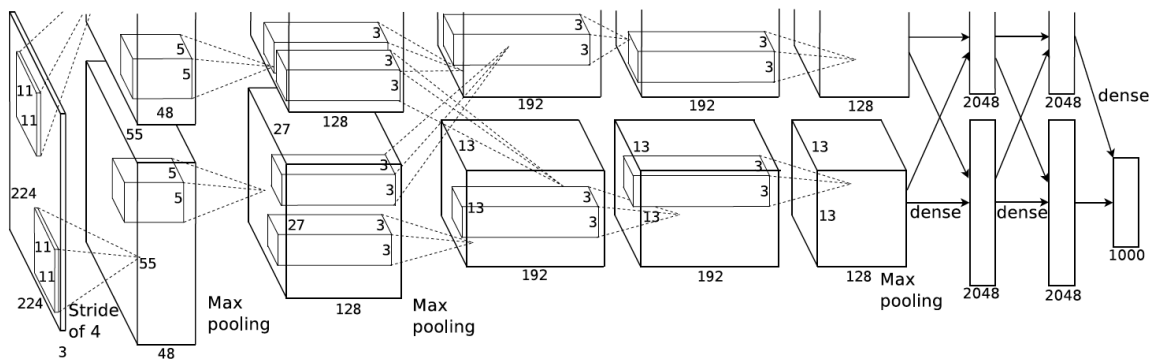


Figura 7 – Arquitetura da rede neural convolucional AlexNet. Fonte: ([KRIZHEVSKY; SUTSKEVER; HINTON, 2012](#)).

2.4.2 VGG

A arquitetura VGG foi proposta por [Simonyan e Zisserman \(2014\)](#) do grupo de geometria visual da universidade de Oxford (do inglês: *Visual Geometry Group*, abreviado por VGG). Foi um dos destaques no ILSVRC-2014 com um erro top-5 de 7.32%. A VGG utiliza pequenos filtros de convolução em suas camadas de tamanho 3×3 . As suas configurações variam de 11 a 19 camadas de profundidade. A Figura 8 apresenta os detalhes da arquitetura VGG16.

2.4.3 GoogLeNet

Também conhecida como Inception V1, foi a vencedora do ILSVRC-2014, obteve um erro top-5 de 6.67%. A GoogLeNet foi proposta por [Szegedy et al. \(2015\)](#) e a maior característica desta rede está na melhor utilização dos recursos computacionais. A sua arquitetura possui um total de 22 camadas e aproximadamente 12 vezes menos parâmetros

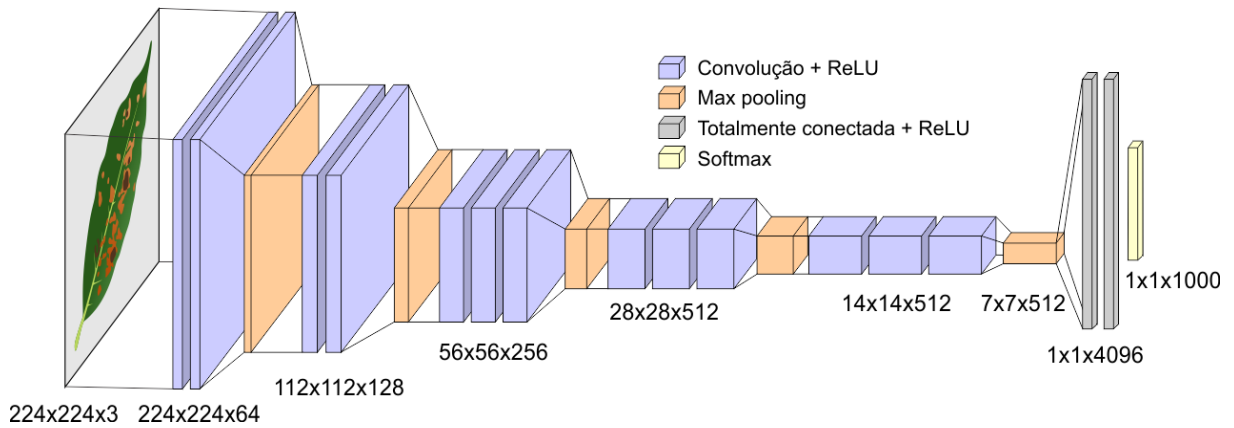


Figura 8 – Arquitetura da rede neural convolucional VGG16.

que a AlexNet, por exemplo. A novidade apresentada pela GoogLeNet são os módulos *inception* que consiste em filtros de múltiplos tamanhos operando no mesmo nível da rede. A Figura 9 apresenta um módulo *inception*.

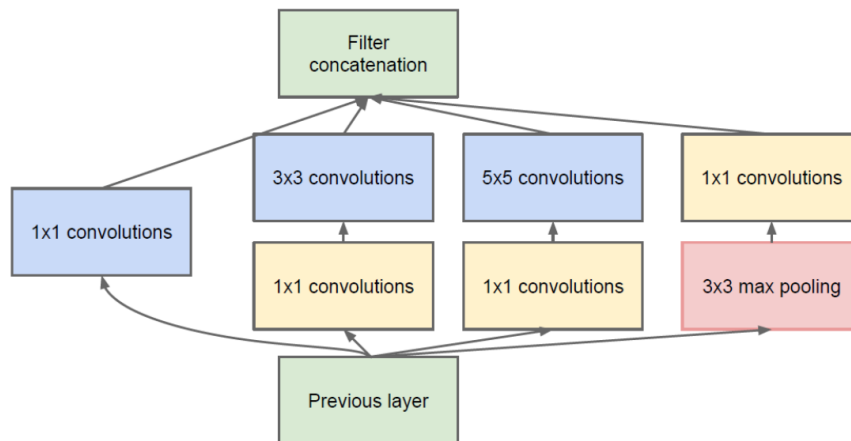


Figura 9 – Módulo *inception*. Fonte: (SZEGEDY et al., 2015).

A entrada do módulo *inception* é processada por diferentes filtros que são concatenados na sua saída. Os filtros de tamanho 1×1 possuem o papel de reduzir o número de canais de entrada diminuindo ainda mais o custo computacional. Basicamente, uma rede GoogLeNet consiste em blocos *inception* empilhados e com ocasionais camadas de *max pooling*.

2.4.4 ResNet

A ResNet é uma arquitetura desenvolvida por pesquisadores da *Microsoft* (HE et al., 2016) e foi vencedora do ILSVRC-2015 com um erro top-5 de 3.57%. Os autores propuseram esta arquitetura com o intuito de resolver o problema de degradação que surge quando as redes profundas começam a convergir. De acordo com o aumento da

profundidade dos modelos de aprendizado profundo, a precisão chega a um ponto que se torna saturada e a partir daí começa a degradar rapidamente os resultados.

Para minimizar este problema e possibilitar o aprendizado de redes mais profundas, foi utilizado o conceito de aprendizado residual para a construção da ResNet. A ResNet é composta por blocos residuais que possuem uma entrada x que passa por uma série de camadas de convolução produzindo uma saída $F(x)$. A saída $F(x)$, produzida pelas camadas de convolução, é posteriormente adicionada a entrada x do bloco, de tal forma que a saída do bloco é $H(x) = F(x) + x$. A Figura 10 exemplifica como é realizada esta conexão.

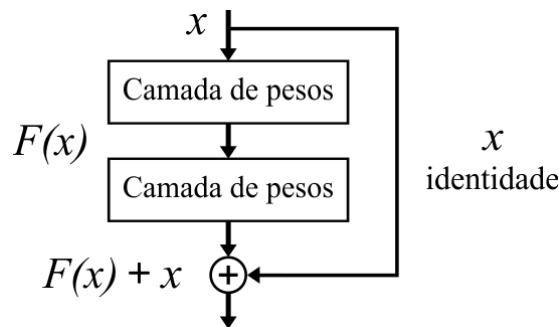


Figura 10 – Bloco residual da ResNet. Fonte: Adaptado de (HE et al., 2016).

2.5 Arquiteturas para segmentação semântica

O sucesso no uso de CNN para problemas de classificação de imagens e detecção de objetos motivou pesquisadores a explorarem a capacidade destas redes na resolução de problemas de segmentação semântica (GARCIA-GARCIA et al., 2017). Desde então, diversas arquiteturas têm sido propostas na literatura com resultados promissores nos mais diversos *benchmarks*. Neste trabalho foram utilizadas duas arquiteturas padrões: UNet e PSPNet.

2.5.1 UNet

A UNet foi proposta por Ronneberger, Fischer e Brox (2015) com o foco na segmentação de imagens biomédicas. A arquitetura consiste em uma rede totalmente convolucional e pode ser dividida em duas etapas principais: contração e expansão, como mostrado na Figura 11. A etapa de contração segue uma típica arquitetura de rede neural convolucional com exceção das camadas totalmente conectadas, ou seja, a imagem de entrada tem sua dimensionalidade reduzida e sua profundidade aumentada através de operações de convolução, ReLU e *max pooling*. Após a contração, vem a etapa de expansão, que faz o processo inverso. A expansão é composta por operações de *upsampling* e

convolução dos mapas de características, que são concatenados com mapas de características correspondentes da etapa de contração.

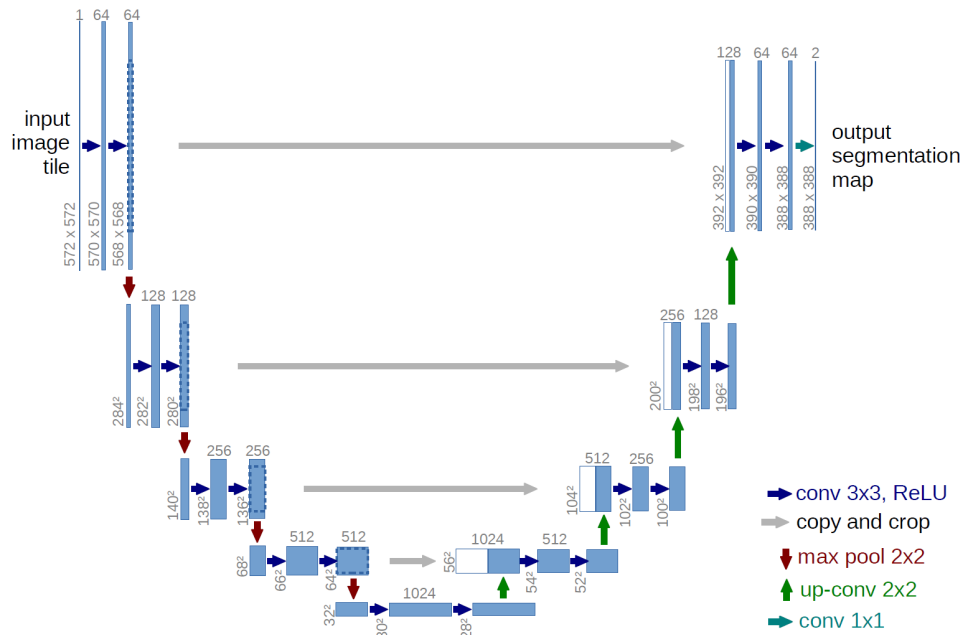


Figura 11 – Arquitetura de rede neural convolucional UNet. Fonte: (RONNEBERGER; FISCHER; BROX, 2015).

2.5.2 PSPNet

A rede de análise de cenas com módulos de agrupamento piramidal (ZHAO et al., 2017) (do inglês *Pyramid Scene Parsing Network*, abreviado por PSPNet) é uma rede de segmentação semântica desenvolvida para a segmentação de cenas complexas, em que a informação de contexto global é importante para diferenciar objetos semelhantes. Por exemplo, uma rede que classifica um barco como um carro, se esta rede pudesse analisar as informações de contexto da cena poderia perceber que o barco está sobre um rio, o que diminui as chances de uma classificação errada. A estrutura da PSPNet pode ser visualizada na Figura 12.

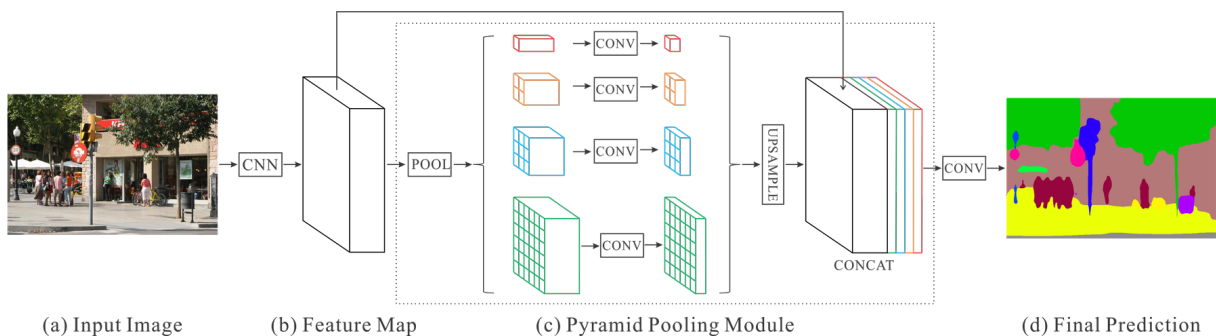


Figura 12 – Arquitetura de rede neural convolucional PSPNet. Fonte: (ZHAO et al., 2017).

Inicialmente, a imagem passa por uma ResNet com a técnica de convolução dilatada (YU; KOLTUN; FUNKHOUSER, 2017) para a extração dos mapas de características. Em seguida, o módulo *pyramid pooling* é aplicado para colher diferentes representações de sub-regiões, ou seja, o mapa de ativação é particionado em diferentes sub-regiões e para cada região é aplicada a operação de *average pooling*. As saídas nos diferentes níveis do módulo *pyramid pooling* contém mapas de características de tamanhos variados, carregando informações contextuais em diferentes escalas. Em seguida, as saídas passam pela operação de *upsampling*, são concatenadas com o mapa de ativação original produzido pela ResNet e passam por uma última etapa de convolução produzindo a predição final pixel-a-pixel.

3 Métodos baseados em Rede Neural Convolutional para Classificação e Segmentação

Neste capítulo é apresentado inicialmente uma visão geral do sistema proposto, em seguida, são detalhadas as duas abordagens baseadas em redes neurais convolucionais investigadas neste trabalho, cujo propósito é resolver o problema de classificação e quantificação de doenças e pragas em folhas de café.

A primeira abordagem consiste em um sistema de etapa única treinado para classificar a base de imagens de folhas com os rótulos de estresse biótico e severidade. A metodologia utilizada é baseada na maioria dos trabalhos da literatura que se limitam apenas a classificar as imagens sem qualquer *feedback* de localização ou área da doença. Como diferencial, nós investigamos a adição da tarefa de classificação de severidade, de tal forma que o sistema, além de reconhecer o sintoma predominante, também estima o intervalo no qual a severidade dos sintomas se enquadra.

Já a segunda abordagem é formada por um sistema de duas etapas, a etapa de segmentação semântica, treinada com a base de máscaras de segmentação, e a etapa de classificação, treinada com a base de imagens de sintomas. As duas etapas são independentes, porém complementares. Os sintomas que são segmentados pela primeira rede são tratados de forma independente e classificados pela segunda rede. Desta maneira, é possível reconhecer múltiplos sintomas e retornar o valor exato da severidade de cada um individualmente.

Mais detalhes dos métodos utilizados são descritos nas subseções seguintes.

3.1 Visão geral da abordagem proposta

Sistemas de classificação baseados em aprendizado profundo seguem um conjunto de etapas bem definido e utilizado na maioria dos trabalhos. De acordo com [Brahimi et al. \(2018\)](#), um sistema para a classificação de doenças de plantas consiste em três fases principais: preparação dos dados, treinamento e implantação. A Figura 13 apresenta uma visão geral do sistema proposto ([ESGARIO; KROHLING; VENTURA, 2019](#)).

3.1.1 Preparação dos dados

Esta é a fase principal de qualquer algoritmo de aprendizado. A construção de uma base bem representativa é um ponto chave no desempenho do modelo. Esta fase consiste em três etapas principais, são elas:

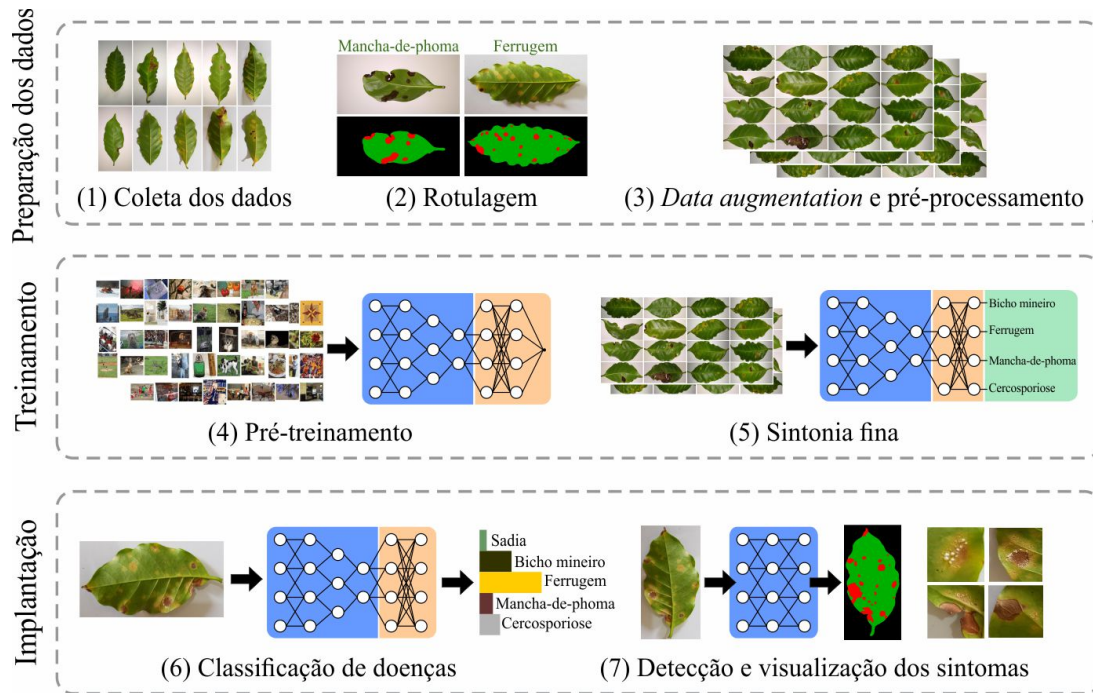


Figura 13 – Visão geral de um sistema baseado em aprendizado profundo para classificação e segmentação de doenças de plantas.

- **Coleta dos dados:** Diferentes problemas possuem diferentes dificuldades de aquisição de imagens. A coleta de imagens de câncer de pele por exemplo é mais difícil se comparado à imagens de folhas. No entanto, coletar imagens de doenças de plantas depende de fatores externos, como época do ano e localização. Além disso, a construção de qualquer uma dessas bases exige um esforço manual muito grande para registrar as imagens e salva-las.
- **Rotulagem:** O processo de rotulagem consiste em anotar as imagens coletadas, geralmente este processo é auxiliado ou realizado por um especialista. A complexidade da rotulagem pode ser separada em rotulagem fraca e rotulagem forte. Na rotulagem fraca, o especialista indica apenas a doença presente naquela folha sem nenhuma informação adicional. Já na rotulagem forte, além dos rótulos das doenças, o especialista demarca a região em que ela se encontra na imagem, essa demarcação pode ser por caixas delimitadoras ou a nível de pixel, logo este processo é bem mais trabalhoso e carece do uso de softwares para auxiliar o especialista. Por esta razão, a maioria das bases disponíveis na literatura são fracamente rotuladas.
- ***Data augmentation* e pré-processamento:** Modelos de aprendizado profundo precisam de um grande volume de dados rotulados para atingirem uma boa capacidade de generalização. Por essa razão, técnicas de *data augmentation* são utilizadas para expandir a base e incluir mais variações aos dados, por exemplo através de transformações geométricas (rotação, espelhamento, recorte, etc.) e transformações de intensidade (contraste, brilho e saturação). O pré-processamento dos dados

consiste em normalizar as imagens de entrada na rede. As técnicas mais comuns são redimensionamento e subtração das cores pela sua média.

3.1.2 Treinamento

De posse dos dados já pré-processados e preparados para serem utilizados, a próxima fase consiste em treinar o modelo com os dados disponíveis. O algoritmo utilizado no treinamento de redes CNN é chamado de *backpropagation* (RUMELHART et al., 1988). Este algoritmo busca minimizar a função de erro através do ajuste dos pesos da rede pelo método do gradiente descendente. A combinação de pesos que minimiza a função de erro é considerada a solução do problema de aprendizado (ROJAS, 1996). Como ilustrado na Figura 13 o treinamento consiste em dois estágios.

- **Pré-treinamento:** Consiste em treinar inicialmente a rede CNN com alguma grande base de dados, por exemplo a ImageNet (1.2 milhões de imagens). A técnica que reutiliza modelos pré-treinados em uma tarefa em outra é chamada *transfer learning*. A ideia se apoia na intuição de que um modelo treinado em uma base suficientemente grande é capaz de aprender a extrair características visuais que servem para a maioria dos problemas.
- **Sintonia fina:** O modelo pré-treinado é, então, reutilizado como ponto de partida para o treinamento do sistema de classificação de doenças. A última camada do modelo é substituída por uma camada compatível com o número de classes do problema. O modelo em questão é retreinado com a base de imagens usando o algoritmo *backpropagation* até que ele se ajuste aos dados.

3.1.3 Implantação

Esta fase visa tornar os modelos treinados acessíveis ao usuário através de interfaces amigáveis, seja um computador ou um *smartphone*. Os modelos treinados podem ser utilizados de dois modos.

- **Classificação de doenças:** A imagem capturada pelo usuário é inserida no modelo e a saída da rede retorna as probabilidades da presença de cada doença naquela folha ou, então, retorna a severidade da doença, tudo depende da tarefa para a qual a rede foi treinada.
- **Detecção e visualização de sintomas:** O modelo é capaz de retornar não apenas a doença, mas também onde ela se encontra, tornando muito mais informativo para o usuário. Este tipo de abordagem é muito útil em especial para usuários inexperientes.

Além disso, o sistema é capaz, até mesmo, de detectar múltiplas doenças em uma mesma folha.

3.2 Classificação de folhas

A primeira abordagem desenvolvida segue a maioria dos trabalhos cujas bases de dados são fracamente rotuladas. Nesta abordagem, o modelo é treinado com a base de folhas apenas para dizer a qual classe pertence determinada imagem, sem fornecer qualquer informação adicional, como a localização da doença.

A Classificação de Folhas (CF) consiste em duas tarefas, classificação de estresse biótico e estimação de severidade a partir de imagens de folhas inteiras. Uma vez que estas tarefas estão intimamente relacionadas (problemas de mesmo domínio). Neste trabalho é proposto a adaptação das redes CNN, tornando-as sistemas com aprendizado multi-tarefa. Portanto, as arquiteturas foram modificadas pela adição de uma nova camada totalmente conectada em paralelo com a existente. Desta forma, a classificação é individualizada, mas as camadas de convolução são compartilhadas, ou seja, o modelo irá aprender características conjuntas úteis na classificação de ambos os problemas. O aprendizado multi-tarefa aproveita a maior parte da arquitetura da rede. Essa simplificação torna o aprendizado substancialmente mais rápido, uma vez que apenas um único modelo precisa ser treinado. A Figura 14 apresenta a estrutura básica de sistemas com tarefas únicas e sistemas multi-tarefa.

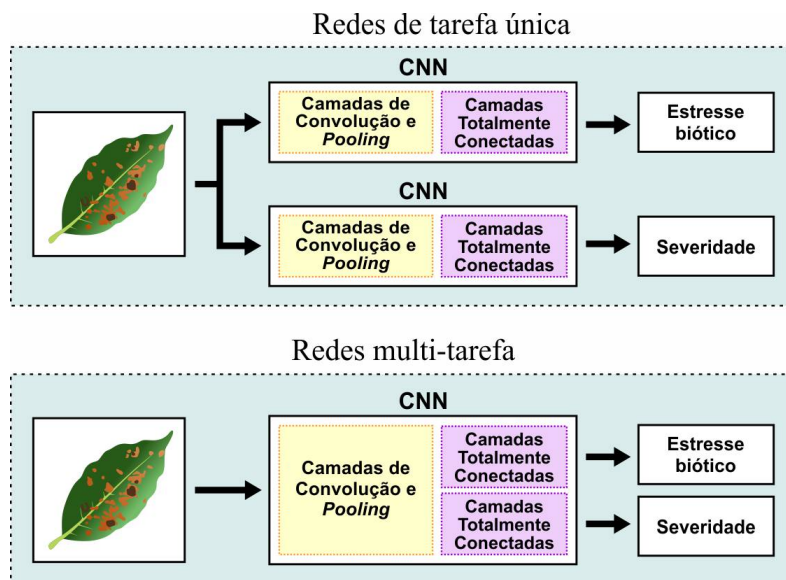


Figura 14 – *Framework* utilizado no aprendizado com tarefa única e multi-tarefa.

As redes CNN podem apresentar dificuldade em capturar características relevantes em imagens da base de folhas que apresentem sintomas pouco visíveis. Para aliviar este problema, um método de segmentação tradicional baseado em um limiar fixo foi utilizado

no canal *S* do espaço de cores *HSV*, de tal forma que a segmentação da folha nos permita definir uma caixa delimitadora para a remoção de excessos da imagem que não envolvam partes da folha. Desta maneira, as imagens foram recortadas e redimensionadas, permitindo que a folha cobrisse maior região da imagem. A Fig 15 apresenta os passos realizados no recorte e redimensionamento das folhas.

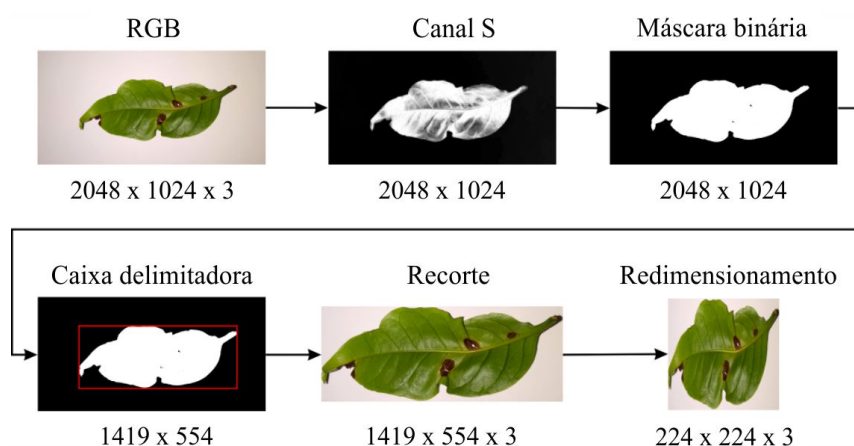


Figura 15 – Etapas para recorte e redimensionamento de imagens de folhas.

3.3 Segmentação e classificação de sintomas

A principal diferença desta abordagem para a anterior está na adição de um estágio de segmentação ao sistema. O modelo agora deve ser capaz de isolar cada sintoma da folha e entregar para uma segunda rede classificar cada um individualmente. A Figura 16 apresenta uma visão geral das etapas da abordagem de Segmentação e Classificação de Sintomas (SCS).

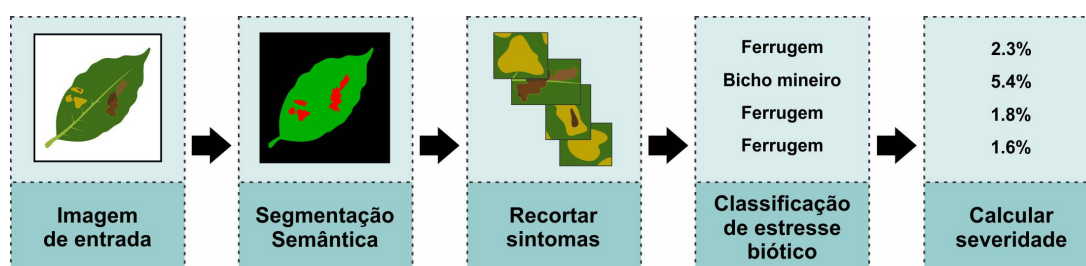


Figura 16 – Principais etapas que compõem a abordagem de segmentação e classificação de sintomas.

A primeira etapa consiste na aquisição da imagem da folha que é redimensionada e normalizada antes de ser inserida na rede de segmentação. A etapa de segmentação semântica é formada por uma rede CNN capaz de isolar o *background*, a folha e os sintomas, atribuindo um rótulo diferente para cada pixel da imagem pertencente a uma dessas classes. A máscara de sintomas é particionada, identificando individualmente cada

componente conexa, ou seja, cada sintoma. Uma caixa delimitadora é posicionada sobre cada componente conexa da máscara de sintomas e as coordenadas dessas caixas são utilizadas para extrair *patches* da imagem original. Estes *patches* formam um *batch* de imagens que são enviadas para o classificador. O classificador consiste em uma rede CNN treinada para reconhecer os principais sintomas de doenças e pragas do café. Após a classificação de cada imagem de sintoma, é possível, com o uso das máscaras geradas pela rede de segmentação, calcular a severidade de cada sintoma individualmente.

Uma vez que os blocos segmentação e classificação são totalmente independentes, o treinamento e validação de ambos foram feitos de forma isolada e só posteriormente combinados para formar um sistema único.

4 Resultados experimentais

A primeira parte deste capítulo consiste em uma descrição detalhada de todas as bases de dados criadas durante o desenvolvimento deste trabalho. São apresentados os resultados experimentais de classificação e segmentação semântica usando as bases de dados de imagens de folhas de café. Os experimentos foram conduzidos a fim de avaliar o quão apropriado os modelos de CNN são para os problemas de classificação de estresse biótico e quantificação da severidade. Detalhes da configuração experimental e dos resultados obtidos são apresentados nas subseções a seguir.

4.1 Bases de dados

A base de dados desenvolvida para este trabalho contém imagens de folhas de café afetadas pelas principais pragas e doenças que atingem a cultura do café. As imagens foram obtidas utilizando-se diferentes *smartphones* (ASUS Zenfone 2, Xiaomi Redmi 5A, Xiaomi S2, Galaxy S8 e Iphone 6S). Além disso, as folhas foram coletadas em diferentes épocas do ano em Marechal Floriano, Espírito Santo. As fotos foram tiradas da face abaxial (inferior) das folhas em condições parcialmente controladas e posicionadas sobre um fundo branco. A aquisição das imagens foi realizada sem muito critério a fim de tornar a base mais heterogênea.

Foram coletadas um total de 1747 imagens de folhas de café, dentre elas folhas saudáveis e folhas doentes, atingidas por um ou mais tipos de estresses bióticos em diversos estágios. O processo de reconhecimento dos estresses bióticos para a rotulagem das imagens foi acompanhado por um especialista. A base desenvolvida é composta pelos quatro tipos de estresses mais comuns que atingem esta cultura, são eles: bicho mineiro, ferrugem, mancha-de-phoma e cercosporiose. Maiores detalhes sobre as pragas e doenças que atingem o cafeeiro podem ser consultados no Apêndice A.

A partir das fotos obtidas, foram geradas três diferentes bases de dados: 1) base com as imagens originais das folhas, 2) base com recortes de cada sintoma individual e 3) base de máscaras de segmentação anotadas a nível de pixel para três classes: fundo, folha e sintoma. Detalhes de cada uma das bases são descritos a seguir.

4.1.1 Base de folhas

É composta pelas imagens originais das folhas inteiras. As imagens foram rotuladas com relação ao estresse biótico predominante em cada folha e a sua severidade. Um total

de 372 imagens apresentaram folhas com mais de um tipo de estresse, dentre elas, 62 folhas apresentaram estresses com severidade semelhante, impossibilitando uma distinção visual de qual estresse era predominante. Portanto, estas 62 imagens não foram utilizadas nessa base.

A severidade dos sintomas foram calculadas por meio da segmentação das lesões e folhas, utilizando métodos de processamento de imagem automáticos apresentados em Manso et al. (2019). Os resultados de segmentação foram validados visualmente para todas as imagens. O valor de severidade para aquelas que apresentaram segmentação ruim foi descartado e estas imagens foram analisadas separadamente por um especialista através de estimativa visual. Para determinados intervalos de severidade, foram atribuídos rótulos da seguinte maneira: sadia ($< 0.1\%$), muito baixa ($0.1\% - 5\%$), baixa ($5.1\% - 10\%$), alta ($10.1\% - 15\%$) e muito alta ($> 15\%$). A Figura 17 apresenta alguns exemplos de imagens contidas na base.

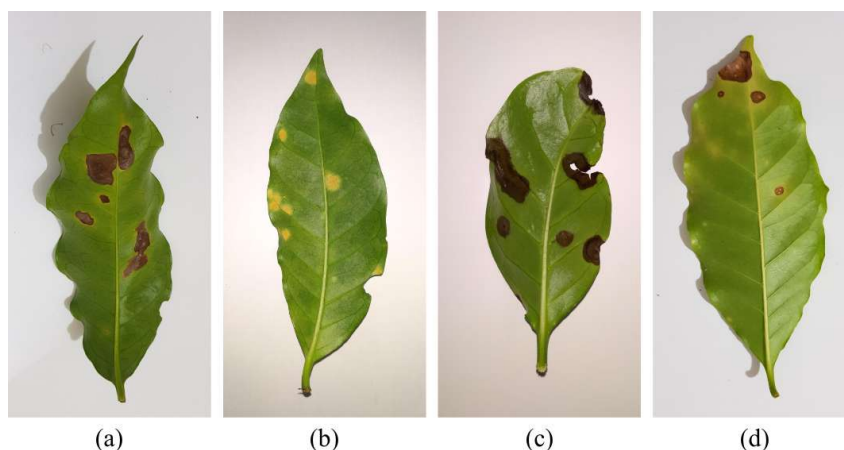


Figura 17 – Exemplos de folhas de café afetadas por diferentes estresses bióticos: bicho mineiro (a), ferrugem (b), mancha-de-phoma (c) e cercosporiose (d).

A Tabela 1 apresenta os detalhes da divisão da base de folhas em termos do estresse biótico e da severidade.

Tabela 1 – Quantidade de amostras por classe da base de folhas.

Estresse biótico	# Imagens	Severidade	# Imagens
Sadia	272	Sadia	272
Bicho mineiro	387	Muito baixa	924
Ferrugem	531	Baixa	332
Mancha-de-phoma	348	Alta	101
Cercosporiose	147	Muito alta	56
Total	1685	Total	1685

4.1.2 Base de sintomas

Esta base de dados foi criada a partir de recortes dos sintomas aparentes nas imagens originais, de maneira a garantir que apenas um único estresse estivesse presente em cada imagem. Foram recortadas um total de 2147 imagens de sintomas. Além das nossas imagens, foram adicionadas à base 575 imagens disponibilizadas por [Barbedo \(2019\)](#) em seu trabalho, totalizando 2722 imagens de sintomas. A Figura 18 apresenta alguns exemplos dos sintomas extraídos das imagens originais.



Figura 18 – Exemplos de imagens de sintomas.

A Tabela 2 especifica o número de imagens para cada estresse da base de sintomas.

Tabela 2 – Quantidade de amostras por classe da base de sintomas.

Estresse biótico	# Imagens
Sadia	256
Bicho mineiro	593
Ferrugem	991
Mancha-de-phoma	504
Cercosporiose	378
Total	2722

4.1.3 Base de segmentação

A construção da base de segmentação consiste em um lento e laborioso processo de anotação das máscaras. Um total de 500 imagens da base de folhas foram selecionadas e os pixels destas imagens foram anotados manualmente com um de três possíveis rótulos: fundo, folha e sintoma. Tais máscaras foram consideradas como padrão-ouro e utilizadas no treinamento e avaliação do modelo de segmentação. A Figura 19 apresenta um exemplo de padrão-ouro gerado a partir de uma foto de folha doente.

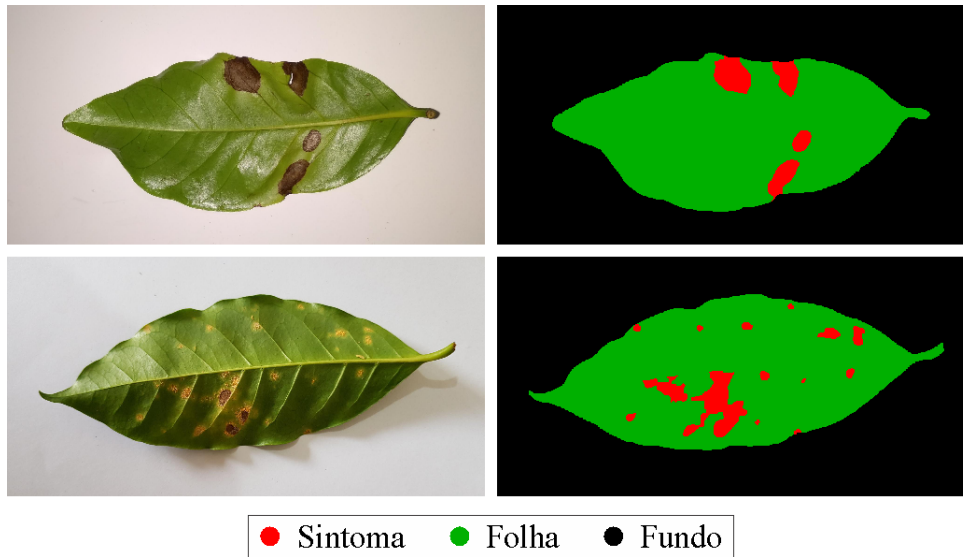


Figura 19 – Exemplos de máscaras de segmentação.

A Tabela 3 apresenta os detalhes de divisão de amostras por classe da base de segmentação. Das 500 imagens selecionadas, 100 consistem de imagens de folhas saudáveis e 400 de folhas doentes.

Tabela 3 – Quantidade de amostras por classe da base de segmentação.

Estresse biótico	# Imagens
Sadia	100
Bicho mineiro	117
Ferrugem	119
Mancha-de-phoma	121
Cercosporiose	33
Total	500

4.2 Configuração experimental

Algumas arquiteturas de CNN não são invariantes ao tamanho das imagens de entrada, portanto os requisitos de dimensão de entrada da rede devem ser obedecidos. Para atender aos requisitos das redes, as imagens de entrada foram redimensionadas conforme a base de dados utilizada. Para as bases de folha e de sintomas, padronizou-se as imagens com largura e altura de 224 pixels e 3 canais de cores que condiz com as dimensões das redes pré-treinadas na base ImageNet. Já as imagens da base de segmentação, foram redimensionadas para uma largura de 512 pixels, altura de 256 pixels e 3 canais de cores. Uma vez que as redes de segmentação são invariantes ao tamanho de entrada, foi escolhido o tamanho 512x256x3 arbitrariamente, buscando equilibrar custo computacional e qualidade de definição das máscaras.

Para a realização de todos os experimentos, foram utilizadas as proporções 70-15-15 para as bases de treino, validação e teste, respectivamente. As técnicas de *data augmentation* foram aplicadas *online* durante o treinamento, de tal forma que a cada novo *batch* de imagens são produzidas novas imagens, que são propagadas pela rede CNN e os pesos da rede são ajustados até que se aprenda as características discriminativas mais relevantes daquele conjunto de imagens. Para tornar o treinamento mais eficiente, foi aplicado a técnica de *transfer learning*. O treinamento foi realizado pelo ajuste fino de redes pré-treinadas com a base ImageNet. Os modelos foram retreinados de ponta a ponta, sem limitar a correção de pesos de qualquer camada. Os hiper-parâmetros utilizados em todos os experimentos no treinamento dos modelos, são listados na Tabela 4.

Tabela 4 – Hiper-parâmetros de treinamento da CNN.

Parâmetro	Valor
Otimizador	Gradiente descendente estocástico
Função de perda	Entropia cruzada
Épocas	100
Tamanho do batch	32 (224x224x3) e 4 (512x256x3)
Taxa de aprendizado*	0.01
Momento	0.9
Decaimento do peso	0.0005

*Diminui a um fator de 1/2 ou 1/5 a cada 20 épocas, alternadamente.

Durante o treinamento das redes, foram salvos os estados (conjunto de pesos) em que as funções de perda dos modelos apresentaram o menor valor para o conjunto de validação. Os modelos salvos, foram então avaliados com a base de teste e os resultados foram calculados usando métricas comuns na literatura. Todos os experimentos foram programados utilizando o PyTorch, biblioteca de código aberto específica para aprendizado de máquina (PASZKE et al., 2017), e conduzidos em uma GPU NVIDIA GeForce GTX 1060 com o CUDA 10.0.

4.3 Métricas

Diferentes tarefas de visão computacional utilizam diferentes métricas que se adequam melhor na avaliação do desempenho dos algoritmos. Para a tarefa de classificação, foram utilizadas as métricas: acurácia (AC), que mede o total de predições corretas sobre o número total de amostras; precisão (PR), que representa a capacidade do sistema em evitar falsos positivos; e revocação (RE), mede a fração de padrões positivos que são classificados corretamente. Uma vez que se está avaliando um modelo capaz de classificar mais de duas classes, as métricas precisão e revocação são calculadas para cada classe individualmente e em seguida é calculado a sua média macro, ou seja, as classes são tratadas igualmente de

tal forma que possuem o mesmo peso no valor final. As equações das métricas de avaliação dos classificadores são descritas a seguir:

$$AC = \frac{\sum_{i=1}^k p_{ii}}{\sum_{i=1}^k \sum_{j=1}^k p_{ij}} \quad (4.1)$$

$$PR = \frac{1}{k} \sum_{i=1}^k \frac{p_{ii}}{\sum_{j=1}^k p_{ji}} \quad (4.2)$$

$$RE = \frac{1}{k} \sum_{i=1}^k \frac{p_{ii}}{\sum_{j=1}^k p_{ij}} \quad (4.3)$$

onde p_{ij} é a quantidade de amostras pertencentes a classe i preditas como sendo da classe j , ou seja, p_{ij} é relativo aos valores da matriz de confusão gerada a partir dos resultados de classificação, e k é igual ao número de classes do problema.

As métricas mais comuns utilizadas na tarefa de segmentação semântica são: acurácia (AC) e interseção sobre união média (do inglês: *mean intersection over union*, abreviado por MIOU) (GARCIA-GARCIA et al., 2017). A acurácia em segmentação semântica é comumente chamada na literatura de acurácia de pixel, no entanto o seu cálculo é o mesmo para o problema de classificação, com a diferença de que é feita pixel a pixel. Já o cálculo da métrica MIOU é descrito pela seguinte equação:

$$MIOU = \frac{1}{k} \sum_{i=1}^k \frac{p_{ii}}{\sum_{j=1}^k p_{ij} + \sum_{j=1}^k p_{ji} - p_{ii}} \quad (4.4)$$

A principal diferença das métricas de segmentação semântica para as de classificação é que p_{ij} agora é tratado como pixels e não como amostras.

Além das métricas apresentadas, é possível, a partir das máscaras de segmentação, calcular a severidade das doenças. A métrica de severidade, mede a porcentagem da área foliar lesionada e pode ser calculada por meio da seguinte equação:

$$Severidade = \frac{A_{lesionada}}{A_{folha}} \times 100 \quad (4.5)$$

4.4 Classificação

Esta seção apresenta os resultados de classificação obtidos para as bases de folhas e de sintomas separadamente. Em seguida, há uma subseção de visualização em que são ilustradas as características extraídas por cada rede em uma representação de baixa dimensão.

4.4.1 Folhas

Os resultados obtidos para a base de dados de Folhas são apresentados na Tabela 5. Os resultados foram divididos pelo problema de classificação (estresse biótico e severidade), arquitetura de rede (AlexNet, GoogLeNet, VGG16 e ResNet50) e técnica de aprendizado (tarefa única e multi-tarefa). Os melhores resultados são apresentados em negrito.

Tabela 5 – Resultados de teste obtidos com diferentes arquiteturas para a base de folhas.

Problema	Arquitetura	Rede de tarefa única			Rede multi-tarefa		
		AC(%)	PR(%)	RE(%)	AC(%)	PR(%)	RE(%)
Estresse Biótico	AlexNet	92.46	89.37	90.02	91.67	88.14	88.67
	GoogLeNet	91.67	88.33	89.38	94.05	92.56	90.87
	VGG16	95.47	96.73	91.94	94.44	94.62	89.55
	ResNet50	95.63	94.12	92.70	95.24	95.29	91.14
	ResNet50*	95.63	95.79	92.32	94.84	92.80	91.48
Severidade	AlexNet	84.13	74.23	72.86	86.9	80.88	77.57
	GoogLeNet	82.94	74.27	73.76	82.94	75.39	73.76
	VGG16	86.51	82.49	80.89	86.51	79.50	76.31
	ResNet50	84.13	81.66	78.90	86.51	82.38	80.90
	ResNet50*	78.57	69.33	67.42	80.95	78.17	68.97

*Modelo treinado com ambas as técnicas de *data augmentation*, padrão e *mixup*.

Os resultados alcançados com a arquitetura ResNet50 mostram uma acurácia de 95.24% para classificação de estresse biótico e de 86.51% para a estimação de severidade usando o aprendizado multi-tarefa. Já com o aprendizado de tarefa única, os resultados obtidos para a mesma arquitetura foram de 95.63% e 84.13%. Em termos de desempenho a VGG16 foi a que mais se aproximou da ResNet50, em contrapartida de todas as arquiteturas experimentadas a VGG16 é que apresenta o maior número de parâmetros, o que pode impactar diretamente em um maior custo computacional. Comparando os resultados individuais de cada arquitetura, vemos que a AlexNet, GoogLeNet e ResNet50 foram beneficiadas com o aprendizado *multi-task*. A adição do *mixup* no treinamento da ResNet50 apresentou resultados semelhantes para o problema de classificação de estresse biótico, porém piorou quando aplicado a estimação de severidade. O desempenho do *mixup* é discutido nas subseções seguintes. Além dos resultados de desempenho dos modelos, foram medidos os tempos médios por época em segundos para o treinamento de cada arquitetura e também os tempos médios de execução por imagem em milissegundos (Tabela 6).

Analisando o tempo das arquiteturas investigadas, apresentados na Tabela 6, vemos que não existe uma relação linear direta com os parâmetros da rede. O tempo necessário para o treinamento dos modelos é influenciado por diversas variáveis, como número de parâmetros, número de camadas, ordem das operações de convolução e *pooling*, cálculo das métricas etc. Já o tempo de execução é tempo médio que um determinado modelo leva

Tabela 6 – Quantidade de parâmetros e tempo médio de treinamento e execução das arquiteturas de classificação investigadas.

Arquitetura	Parâmetros (M)	Treinamento (seg./época)	Execução (miliseg.)
AlexNet	57	7.32	0.78
VGG16	134.3	36.48	1.26
GoogLeNet	5.6	11.76	5.56
ResNet50	23.5	21.90	4.75

para realizar a inferência por imagem. Até mesmo a GoogLeNet apresentou um tempo de inferência suficientemente rápido (5.56ms) para a maioria das aplicações.

Dentre as arquiteturas utilizadas, a ResNet50 apresentou um *tradeoff* entre custo computacional, performance e confiabilidade superior às outras. A Figura 20 apresenta as matrizes de confusão associadas aos resultados de predição obtidos com a ResNet50. Os resultados de classificação de estresse biótico foram consistentes para a maioria das lesões, exceto a cercosporiose que apresentou uma quantidade considerável de erros de classificação. Este resultado corrobora com os experimentos realizados em Barbedo (2019), cuja classe com maior erro também foi a cercosporiose. Estes erros podem estar associados a semelhança com as outras doenças e também com o menor número de amostras disponíveis em relação às outras classes. Além disso, a cercosporiose é uma doença que apresenta uma variedade muito grande nas características visuais dos sintomas, que são influenciadas pelo seu estágio de desenvolvimento e pelo meio ambiente.

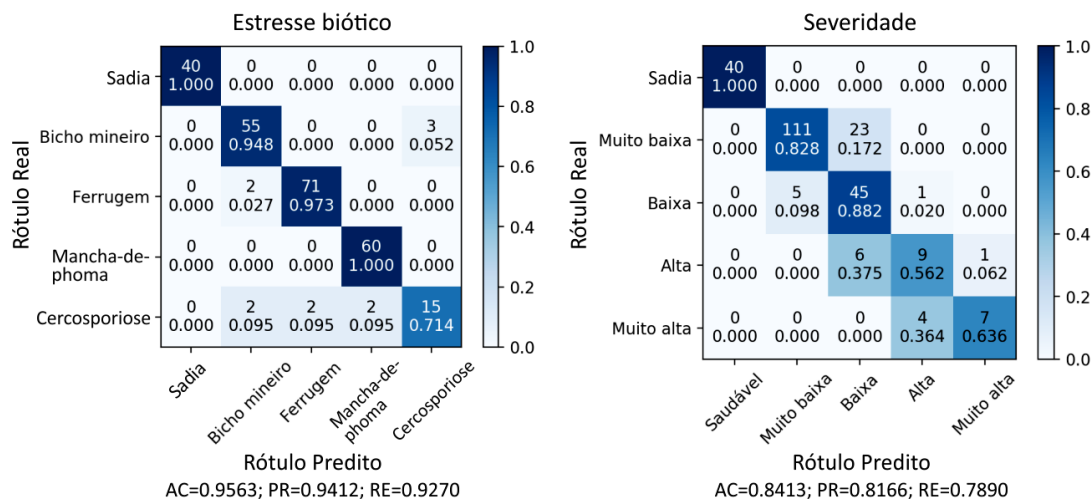


Figura 20 – Matriz de confusão com os resultados de predição usando a arquitetura ResNet50 para a base de folhas.

A estimação de severidade com a ResNet50 obteve uma acurácia de 84.13%. Este problema é claramente mais desafiador se comparado à classificação de estresse biótico. O modelo não apresentou dificuldade em separar folhas saudáveis de folhas doentes, porém, devido ao grande desbalanceamento da base de dados, a classificação de folhas com alta e muito alta severidade apresentou menor porcentagem de acertos. Apesar do modelo ter

cometido uma quantidade considerável de erros, percebemos pela matriz de confusão que os erros estão situados próximos a diagonal principal. Devido à natureza ordinal inerente dos rótulos de severidade, estimar a severidade de um estresse como alta em vez de muito alta, por exemplo, pode ser considerado um erro pequeno para esse problema em questão.

4.4.2 Sintomas

Os resultados experimentais obtidos com a base de sintomas, apresentados na Tabela 7, alcançaram uma acurácia de 97.07% com a arquitetura ResNet50. Como esperado, os resultados de classificação apenas dos sintomas são superiores quando comparados a classificação usando imagens de folhas inteiras. Isto ocorre devido às imagens de sintomas possuírem uma região bem delimitada da área de interesse, o que leva a eliminação de elementos espúrios. Logo, a rede pode focar e visualizar um nível maior de detalhes dos sintomas. A arquitetura ResNet50, mais uma vez, apresentou os melhores resultados, diferentemente da VGG16 que apresentou bons resultados para a base de folhas, mas um resultado inferior para a base de sintomas.

Tabela 7 – Resultados de teste obtidos com diferentes arquiteturas para a base de sintomas.

Arquitetura	Estresse biótico		
	AC(%)	PR(%)	RE(%)
AlexNet	96.58	96.12	96.59
GoogLeNet	96.82	96.56	96.64
VGG16	95.60	95.02	95.31
ResNet50	97.07	96.85	96.69
ResNet50*	97.07	96.85	96.99

*Modelo treinado com o aumento padrão e *mixup*.

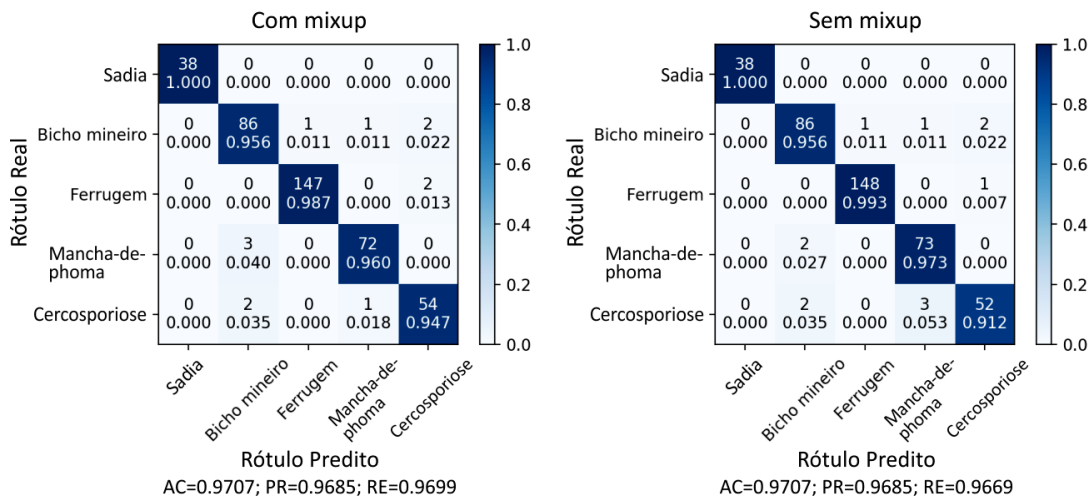


Figura 21 – Matriz de confusão com os resultados de predição usando a arquitetura ResNet50 para a base de sintomas.

A adição do *mixup* ao treinamento da ResNet50 apresentou um pequeno ganho de RE se comparado ao modelo treinado apenas com as técnicas de *data augmentation* padrão. Embora o desempenho seja considerado o mesmo, foi percebido durante o treinamento que o uso dessa técnica dificulta o modelo a sofrer *overfitting* durante a etapa de treinamento, devido à grande diversidade de imagens novas que são geradas a cada iteração. A Figura 21 apresenta as matrizes de confusão com os resultados de predição obtidos pela ResNet50 treinada com e sem o *mixup*. O valor de RE superior é reflexo de resultados mais bem distribuídos entre as classes.

4.4.3 Visualização

Por fim, foi utilizada a técnica *t-Distributed Stochastic Neighbor Embedding* (t-SNE) (MAATEN; HINTON, 2008) na visualização das características de alta dimensão extraídas pela CNN. Ao todo, foram extraídas 2048 características de cada amostra de teste logo após a última camada de convolução da ResNet50. Estas características foram reduzidas a duas dimensões usando o t-SNE.

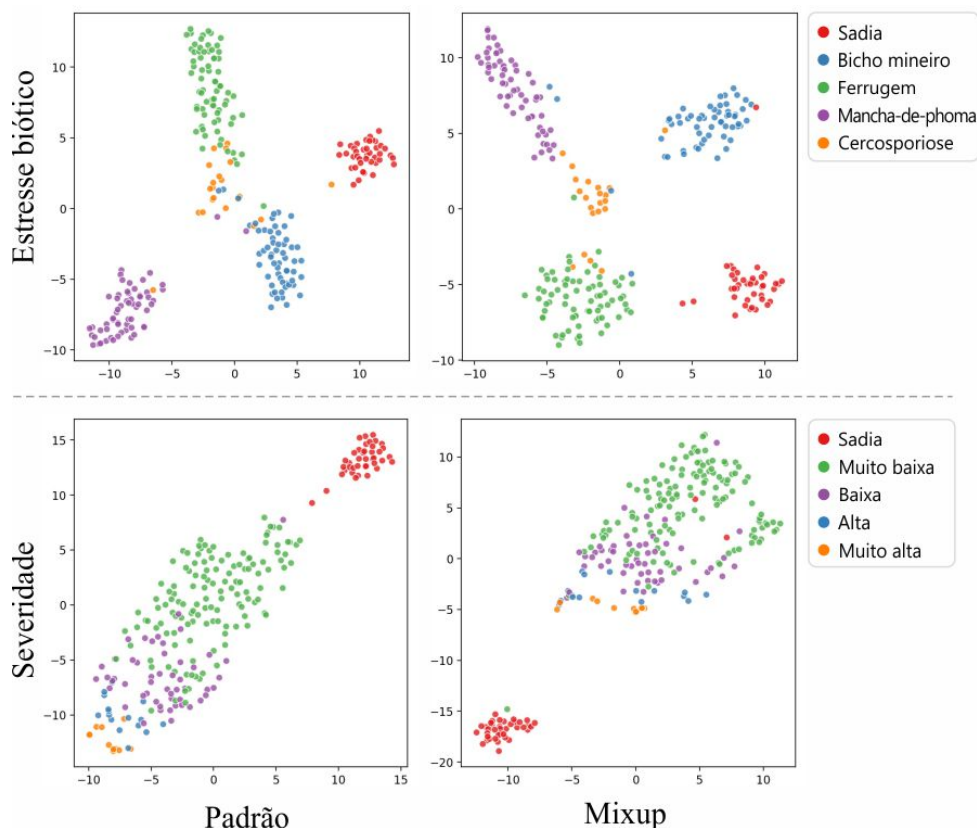


Figura 22 – Visualização da disposição espacial das características extraídas pela ResNet50 para as amostras de teste da base de folhas usando o t-SNE.

A Figura 22 apresenta gráficos de dispersão com as características extraídas das amostras de teste da base de folhas. Os gráficos foram divididos em termos da tarefa de

classificação (estresse biótico e severidade) e em relação a técnica de *data augmentation* (padrão e *mixup*) utilizada.

Visualizando as características extraídas para o problema de estresse biótico, vemos que a cercosporiose apresentou a maior sobreposição com as outras classes, mostrando a dificuldade do modelo em separar essa classe. Em contrapartida, o modelo apresentou *clusters* bem definidos de cada classe, sugerindo que o modelo possui uma alta capacidade de generalização. Além disso, a estimação de severidade com o *data augmentation* padrão mostrou uma relação bem linear entre os rótulos e sua faixa de severidade.

As características aprendidas com o *mixup* apresentaram uma variância intra-classe menor quando comparado ao *data augmentation* padrão. Em especial para a base de sintomas mostrado na Figura 23, fica claro como as características aprendidas usando os dados gerados pelo *mixup* apresentam uma distribuição mais esférica. Embora o *mixup* tenha apresentado um comportamento mais adequado para a separação das características, para o problema de estresse biótico a técnica piorou a classificação quando aplicada a estimação de severidade. A análise dos resultados sugere que, devido à natureza ordinal dos rótulos de severidade, a combinação das imagens com o *mixup* produziu imagens com características de severidade mediana, o que tendenciou a rede a aprender mais rótulos intermediários do que rótulos extremos, ocasionando em uma maior sobreposição das características.

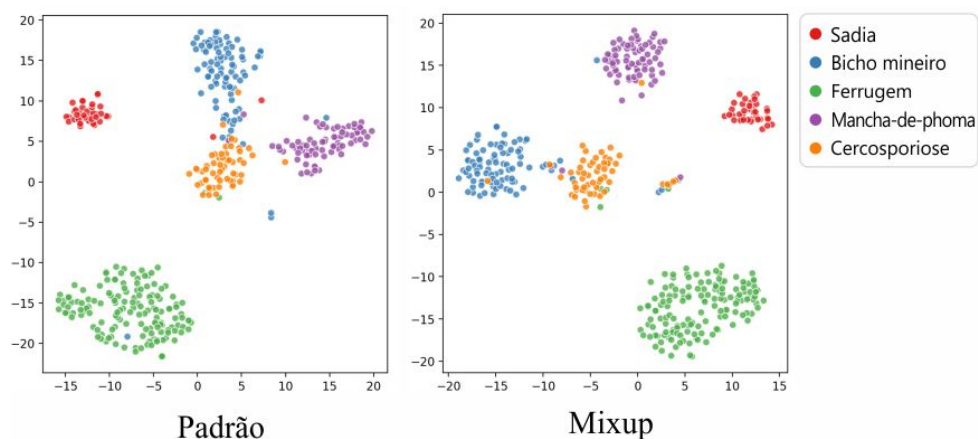


Figura 23 – Visualização da disposição espacial das características extraídas pela ResNet50 para as amostras de teste da base de sintomas usando o t-SNE.

4.5 Segmentação semântica

Os modelos obtiveram resultados bastante satisfatórios para as 50 imagens de testes da base de segmentação. A partir da Tabela 8, podemos visualizar os resultados obtidos com as métricas MIOU e AC para as arquiteturas UNet e PSPNet. A UNet apresentou desempenho consideravelmente superior a PSPNet, em especial se comparadas pela métrica

MIoU, com uma diferença superior a 1%. Todavia, ambos os resultados foram considerados muito bons, de maneira que até mesmo o padrão-ouro pode apresentar inconsistências nas demarcações dos sintomas no processo de anotação manual das máscaras. Estas inconsistências, surgem devido à subjetividade existente ao se definir os limites entre sintoma e folha, além disso há também a imprecisão no manuseio das ferramentas de anotação de máscaras utilizadas.

Tabela 8 – Resultados de teste obtidos com UNet e PSPNet para a base de segmentação.

Arquitetura	MIoU(%)	AC(%)
UNet	94.85	99.53
PSPNet	93.69	99.31
Média	94.27	99.42

Uma vez que as máscaras de segmentação permitem um cálculo preciso da severidade dos sintomas, foi calculada a severidade para cada amostra de teste a partir das máscaras padrão-ouro (Severidade real) e das máscaras geradas pelas redes CNN (Severidade predita). A partir dos valores de severidade, gráficos de dispersão foram gerados e os coeficientes de determinação R^2 calculados, a fim de verificar o quanto os dados se aproximam da reta de regressão. Os gráficos podem ser visualizados na Figura 24.

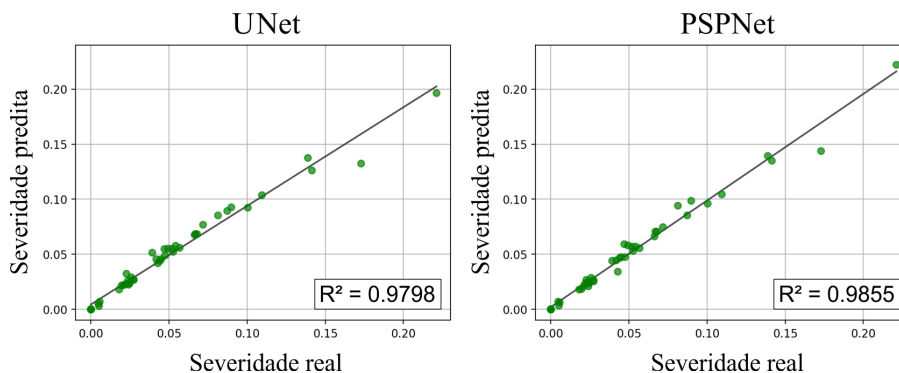


Figura 24 – Gráfico de dispersão da severidade real versus severidade predita.

Ambas as arquiteturas apresentaram alta correlação entre a severidade predita com a severidade real, com valores de R^2 iguais a 0.9798 e 0.9855 para as redes UNet e PSPNet, respectivamente. É interessante notar que, embora a PSPNet tenha apresentado resultados inferiores de MIoU e AC, os valores de severidade apresentaram maior proximidade com aqueles obtidos a partir das máscaras padrão-ouro. A fim de verificar o comportamento das redes na geração das máscaras, foram selecionadas alguns resultados da base de teste visando cobrir desde o pior e o melhor caso de segmentação. Os resultados de segmentação obtidos para cada uma das imagens selecionadas podem ser visualizados na Figura 25.

Analisando visualmente os resultados de segmentação obtidos, é possível notar que a rede PSPNet apresenta máscaras mais suaves e arredondadas se comparada a rede UNet. Esta tendência de suavização das máscaras leva a valores de MIoU inferiores






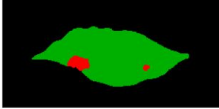
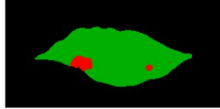
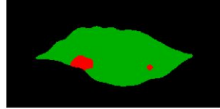


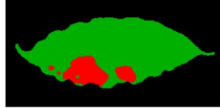
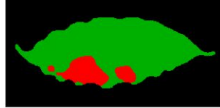

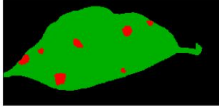
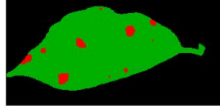
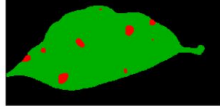



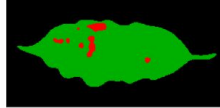

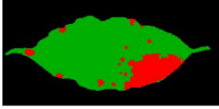
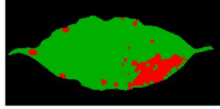

Foto original	Padrão ouro	UNet	PSPNet	
				↑ Melhor caso
Severidade MIoU	0.0% 1.000	0.0% 0.995	0.0% 0.992	
				
Severidade MIoU	4.8% 1.000	4.9% 0.982	4.9% 0.972	
				
Severidade MIoU	13.9% 1.000	13.8% 0.969	14.4% 0.954	
				↓ Pior caso
Severidade MIoU	4.3% 1.000	4.2% 0.929	3.5% 0.907	
				
Severidade MIoU	3.9% 1.000	5.1% 0.900	4.6% 0.866	
				
Severidade MIoU	17.3% 1.000	13.3% 0.893	14.9% 0.909	

Figura 25 – Resultados visuais de segmentação de algumas imagens da base de teste.

quando comparados aos obtidos pela UNet. As imagens cujas redes apresentaram maiores dificuldades de segmentação foram aquelas de folhas infectadas por ferrugem, sendo este um comportamento esperado, já que os sintomas de ferrugem por vezes apresentam tons de amarelo muito parecidos com a cor do tecido foliar.

Por fim, foram medidos para cada arquitetura de segmentação, os tempos médios de treinamento por época em segundos e os tempos médios de execução por imagem em milissegundos. De acordo com os resultados apresentados na Tabela 9, pode-se perceber que a UNet é consideravelmente mais rápida durante a etapa de treinamento quando comparada a PSPNet. Em contrapartida, as arquiteturas não apresentaram diferenças significativas em tempo de execução. Além disso, o número de parâmetros da UNet é bastante superior. Se uma aplicação possui restrições de capacidade de armazenamento em disco, talvez o armazenamento dos pesos da UNet pode se tornar um problema.

Tabela 9 – Quantidade de parâmetros e tempo médio de treinamento e execução época das arquiteturas de segmentação investigadas.

Arquitetura	Parâmetros (M)	Treinamento (seg./época)	Execução (miliseg.)
UNet	148.1	206	7.30
PSPNet	53.6	422	8.02

4.6 Comparativo entre abordagens

Tendo em vista as abordagens utilizadas, classificação de folhas (CF) e segmentação e classificação de sintomas (SCS), um resumo com os melhores resultados experimentais obtidos é apresentado na Tabela 10. Analisando os resultados, é fácil notar que o desempenho obtido com a abordagem SCS é bastante superior quando comparado ao desempenho da abordagem CF. Essa diferença em desempenho vem com um custo. A abordagem SCS depende de dois modelos de CNN, sendo que os modelos de segmentação, em geral, são mais complexos e possuem um número maior de parâmetros do que os modelos de classificação. Isto impacta em maior custo computacional em termos de processamento, memória e disco. Além do custo computacional, duas redes precisam ser treinadas e cada rede exige uma base de dados diferente. A construção das bases de dados para a abordagem SCS consiste em um processo manual demorado e trabalhoso, o que pode ser uma barreira na hora de implementar um sistema deste tipo. Em contra partida, o CF é substancialmente mais simples e a base de dados exige pouco ou nenhum pré-processamento.

Tabela 10 – Resumo com os melhores resultados experimentais para ambas as abordagens propostas.

Abordagem	Arquitetura	Tarefa	Problema	AC(%)	PR(%)	RE(%)	MIoU(%)
CF	ResNet50 (Tarefa única)	Classificação	Estresse biótico	95.63	94.12	92.70	—
			Severidade	84.13	81.66	78.90	—
	ResNet50 (Multi-tarefa)	Classificação	Estresse biótico	95.24	95.29	91.14	—
			Severidade	86.51	82.38	80.90	—
SCS	ResNet50*	Classificação	Estresse biótico	97.07	96.85	96.99	—
	UNet	Segmentação	—	99.53	—	—	94.85

*Modelo treinado com ambas as técnicas de *data augmentation*, padrão e *mixup*.

A escolha de uma abordagem ou outra depende principalmente dos dados e recursos computacionais disponíveis e do nível de detalhes que o usuário necessita a respeito do diagnóstico das doenças. Uma aplicação que deva ser embarcada totalmente em um dispositivo móvel, por exemplo, e que só necessite do rótulo da doença predominante, pode utilizar a abordagem CF. Por outro lado, se houver uma maior disponibilidade de recursos computacionais e se for necessário uma maior precisão na estimativa da severidade e identificar múltiplos estresses presentes em uma determinada folha, o SCS se torna a opção mais adequada. Nesse contexto foi desenvolvido e implementado um aplicativo para *smartphone* que está descrito no Apêndice B.

5 Conclusão

Os desafios que permeiam o problema de classificação de doenças de plantas, como variações das características dos sintomas, múltiplas desordens em uma mesma folha e a similaridade de diferentes sintomas, fazem deste um problema bastante desafiador. Um agravante do problema que foi percebido no desenrolar deste trabalho é a carência de bases públicas de imagens de folhas, em especial as fortemente rotuladas. Os trabalhos que desenvolveram sistemas baseados em detecção de objetos ou segmentação semântica não tornaram disponíveis seus dados, limitando novos trabalhos nesta área. Portanto, para a realização dos experimentos e objetivando contribuir para trabalhos futuros na área, foram desenvolvidas três novas bases de imagens de folhas de café, sendo elas base de folhas, base de sintomas e uma base com máscaras de segmentação.

A metodologia proposta consistiu em avaliar diferentes abordagens baseadas em aprendizado profundo para o problema de classificação e quantificação de estresse biótico. Os experimentos conduzidos para classificação de folhas e sintomas mostraram uma clara superioridade da arquitetura ResNet50 em relação as outras. Os resultados de classificação de estresse biótico obtidos com a ResNet50 a partir de fotos de folhas inteiras apresentaram uma acurácia de 95.65% e os resultados com imagens de recortes de sintomas aumentaram a acurácia para 97.07%. Foi verificado durante os experimentos com a abordagem de classificação de folhas que o aprendizado multi-tarefa pode tornar o modelo mais efetivo na resolução de múltiplos problemas de mesmo domínio. Das técnicas de *data augmentation* investigadas, o *mixup* apresentou uma boa capacidade de regularização e também uma melhor separação das características para o problema de classificação de estresse biótico. Já a abordagem de segmentação e classificação de sintomas permitiu que o sistema classificasse múltiplos estresses e quantificasse de forma bastante precisa a severidade dos sintomas. Os resultados de segmentação semântica alcançaram um valor de 94.85% para a métrica intersecção sobre união média com a rede UNet, este resultado confirma que o modelo consegue reproduzir máscaras muito fieis àquelas geradas manualmente.

A principal limitação deste trabalho está relacionada a baixa representatividade das bases de dados, pois estas cobrem apenas os principais estresses bióticos que atingem a cultura do café e com um número de imagens relativamente baixo. Mesmo assim, as abordagens propostas usando redes convolucionais se mostraram consistentes e com bons resultados. O aumento do número de amostras das bases de dados e a adição de novos tipos de estresses bióticos podem ajudar a aliviar este problema e tornar o sistema ainda mais abrangente e confiável. Outra limitação está relacionada a abordagem de segmentação e classificação de sintomas. Esta abordagem não consegue identificar sintomas de duas doenças diferentes em uma única folha quando há uma sobreposição das áreas infectadas,

ou seja, se o modelo gerar uma máscara totalmente conexa que cobre a mesma área de sintomas de múltiplas doenças, a rede de classificação identificará aquela máscara como pertencente a apenas uma classe.

Embora o trabalho tenha focado em um estudo de caso utilizando folhas de café, toda a metodologia aplicada é facilmente escalável a qualquer outra cultura apenas pela adição de novas imagens e pela adaptação das saídas das redes para o número de classes desejadas. Para trabalhos futuros, pode ser avaliado a possibilidade de aplicação do sistema em fotos tiradas em campo dos pés de café ou de outras culturas. A aquisição dessas imagens pode ainda ser totalmente automatizada por meio de veículos não tripulados. Embora o sistema tenha apresentado bons resultados, ainda há de ser melhorado. A aplicação de outras arquiteturas CNN e técnicas de *data augmentation* devem ser investigadas para melhorar o desempenho do sistema.

Referências

- ALOYSIUS, N.; GEETHA, M. A review on deep convolutional neural networks. In: *Proceedings of the International Conference on Communication and Signal Processing (ICCSP)*. Chennai, India: IEEE, 2017. p. 588–592.
- BARBEDO, J. G. Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, v. 172, p. 84–91, 2018.
- BARBEDO, J. G. A. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, v. 144, p. 52–60, 2016.
- BARBEDO, J. G. A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, v. 153, p. 46–53, 2018.
- BARBEDO, J. G. A. Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, v. 180, p. 96–107, 2019.
- BARBEDO, J. G. A.; KOENIGKAN, L. V.; SANTOS, T. T. Identifying multiple plant diseases using digital image processing. *Biosystems Engineering*, v. 147, p. 104–116, 2016.
- BOCK, C. et al. Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Critical Reviews in Plant Sciences*, v. 29, n. 2, p. 59–107, 2010.
- BOUREAU, Y.-L.; PONCE, J.; LECUN, Y. A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Haifa, Israel: Omnipress, 2010. p. 111–118.
- BRAHIMI, M. et al. Deep learning for plant diseases: detection and saliency map visualisation. In: *Human and Machine Learning*. [S.l.]: Springer, 2018. p. 93–117.
- CLEVERT, D.-A.; UNTERTHINER, T.; HOCHREITER, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv e-prints*, arXiv:1511.07289, 2015.
- DEHNEN-SCHMUTZ, K. et al. Exploring the role of smartphone technology for citizen science in agriculture. *Agronomy for Sustainable Development*, v. 36, n. 2, p. 25, Apr 2016.
- ESGARIO, J. G. M.; KROHLING, R. A.; VENTURA, J. A. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*, 2019. Submetido à publicação.
- FORNAZIER, M. J. et al. Manejo de pragas do café conilon. In: FERRÃO, R. G. et al. (Ed.). *Café Conilon*. 2. ed. Vitória, ES: Incaper, 2017. p. 398–433.
- FUENTES, A. et al. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, v. 17, n. 9, p. 2022, 2017.
- GARCIA-GARCIA, A. et al. A review on deep learning techniques applied to semantic

- segmentation. *arXiv e-prints*, arXiv:1704.06857, 2017.
- GHOSAL, S. et al. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences*, v. 115, n. 18, p. 4613–4618, 2018.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- GUO, Y. et al. Deep learning for visual understanding: A review. *Neurocomputing*, v. 187, p. 27–48, 2016.
- HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015. p. 1026–1034.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, 2016. p. 770–778.
- HU, G. et al. A low shot learning method for tea leaf's disease identification. *Computers and Electronics in Agriculture*, v. 163, p. 104852, 2019.
- HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, v. 195, n. 1, p. 215–243, 1968.
- HUGHES, D. P.; SALATHE, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv e-prints*, arXiv:1511.08060, 2015.
- International Coffee Organization. *ICO Annual Review*. 2019. Acessado em 28-08-2019. Disponível em: <<http://www.ico.org/documents/cy2018-19/annual-review-2017-18-e.pdf>>.
- JOHANNES, A. et al. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. *Computers and Electronics in Agriculture*, v. 138, p. 200–209, 2017.
- KAYA, A. et al. Analysis of transfer learning for deep neural network based plant classification models. *Computers and Electronics in Agriculture*, v. 158, p. 20–29, 2019.
- KRANZ, J. Measuring plant disease. In: *Experimental techniques in plant disease epidemiology*. Berlin, Heidelberg: Springer, 1988. p. 35–50.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*. [S.l.]: Curran Associates, Inc., 2012. p. 1097–1105.
- KROGH, A.; HERTZ, J. A. A simple weight decay can improve generalization. In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991. (NIPS'91), p. 950–957.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, v. 521, n. 7553, p. 436–444, 2015.

- LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, Nov 1998.
- LIANG, Q. et al. PD2SE-Net: Computer-assisted plant disease diagnosis and severity estimation network. *Computers and Electronics in Agriculture*, v. 157, p. 518–529, 2019.
- LIN, K. et al. Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Frontiers in Plant Science*, Frontiers, v. 10, p. 155, 2019.
- LIU, B. et al. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, v. 10, n. 1, p. 11, 2017.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 3431–3440.
- MA, J. et al. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, v. 154, p. 18–24, 2018.
- MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: *Proceedings of the 30th International Conference on Machine Learning (ICML)*. Atlanta, USA: [s.n.], 2013. v. 30, n. 1, p. 3.
- MAATEN, L. v. d.; HINTON, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008.
- MANSO, G. L. et al. A smartphone application to detection and classification of coffee leaf miner and coffee leaf rust. *arXiv e-prints*, arXiv:1904.00742, 2019.
- MOHANTY, S. P.; HUGHES, D. P.; SALATHÉ, M. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, v. 7, p. 1419, 2016.
- OLIVEIRA, C. et al. Crop losses and the economic impact of insect pests on Brazilian agriculture. *Crop Protection*, v. 56, p. 50–54, 2014.
- Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, v. 22, n. 10, p. 1345–1359, 2010.
- PASZKE, A. et al. Pytorch. *Computer Software. Vers. 0.3*, v. 1, 2017.
- PRETTY, J. Agricultural sustainability: concepts, principles and evidence. *Philosophical Transactions of the Royal Society B: Biological Sciences*, v. 363, n. 1491, p. 447–465, 2007.
- RAWAT, W.; WANG, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, MIT Press, v. 29, n. 9, p. 2352–2449, 2017.
- ROJAS, R. The backpropagation algorithm. In: *Neural Networks*. Berlin, Heidelberg: Springer, 1996. p. 149–182.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional networks for biomedical image segmentation. In: *Proceedings of the International Conference on Medical image computing and computer-assisted intervention*. Munich, Germany: Springer, 2015. p. 234–241.

- RUDER, S. An overview of multi-task learning in deep neural networks. *arXiv e-prints*, arXiv:1706.05098, 2017.
- RUMELHART, D. E. et al. Learning representations by back-propagating errors. *Cognitive Modeling*, v. 5, n. 3, p. 1, 1988.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv e-prints*, arXiv:1409.1556, 2014.
- SINGH, A. K. et al. Deep learning for plant stress phenotyping: Trends and future perspectives. *Trends in Plant Science*, v. 23, n. 10, p. 883–898, 2018.
- SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research (JMLR)*, v. 15, n. 1, p. 1929–1958, 2014.
- SUZUKI, N. et al. Abiotic and biotic stress combinations. *New Phytologist*, v. 203, n. 1, p. 32–43, 2014.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, 2015. p. 1–9.
- TOO, E. C. et al. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, v. 161, p. 272–279, 2019.
- VENTURA, J. A.; COSTA, H.; LIMA, I. d. M. Manejo de pragas do café conilon. In: FERRÃO, R. G. et al. (Ed.). *Café Conilon*. 2. ed. Vitória, ES: Incaper, 2017. p. 434–479.
- WANG, G.; SUN, Y.; WANG, J. Automatic image-based plant disease severity estimation using deep learning. *Computational Intelligence and Neuroscience*, Hindawi, v. 2017, 2017.
- XU, B. et al. Empirical evaluation of rectified activations in convolutional network. *arXiv e-prints*, arXiv:1505.00853, 2015.
- YU, F.; KOLTUN, V.; FUNKHOUSER, T. Dilated residual networks. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Havaí, EUA: IEEE, 2017. p. 472–480.
- ZHANG, H. et al. mixup: Beyond empirical risk minimization. *arXiv e-prints*, arXiv:1710.09412, 2017.
- ZHAO, H. et al. Pyramid scene parsing network. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Havaí, EUA: IEEE, 2017. p. 2881–2890.

Apêndices

APÊNDICE A

Pragas e doenças do cafeeiro

Este apêndice apresenta um resumo dos capítulos 17 e 18 do livro *Café Conilon* (FORNAZIER *et al.*, 2017; VENTURA; COSTA; LIMA, 2017), que apresentam detalhes sobre as principais pragas e doenças que atingem o cafeeiro.

A.1 Pragas

As pragas que ocorrem de forma sistemática, chamadas de pragas-chave, normalmente causam danos quantitativos e/ou qualitativos toda vez que uma lavoura é implantada. Usualmente é recomendado o uso de produtos químicos com o objetivo de reduzir rapidamente a sua população. No entanto, o uso de forma incorreta e frequente deste tipo de produto induz ao aumento da resistência de tais pragas aos químicos usados. Além disso, pode provocar intoxicação direta dos trabalhadores e reduzir a qualidade dos produtos vendidos. A severidade com que ocorre os ataques das pragas é favorecida ou atenuada devido aos fatores climáticos e por meio da aplicação de boas práticas.

Neste trabalho, apenas folhas atingidas pela praga bicho mineiro foram inseridas a base de dados. No entanto, existem outras pragas de grande importância econômica para o cafeeiro. As subseções a seguir apresentam uma descrição mais detalhada das pragas estudadas.

A.1.1 Bicho mineiro

O bicho mineiro é uma mariposa de origem africana e hoje está presente em todas as regiões de cultivo do café. Possui uma envergadura de aproximadamente 6,5mm e asas cor branco-prateada. Se escondem na parte inferior da folha durante o dia e ao escurecer abandonam o esconderijo e iniciam as atividades. Em média, colocam 7 ovos por noite em diferentes partes da folha. Após um período de 5 a 21 dias, dependendo de fatores climáticos, ocorre o processo de eclosão das lagartas que penetram na folha, minando-a ao se alimentarem. O período larval pode durar até 40 dias. Após esta fase, as lagartas abandonam as folhas pela face inferior e constroem um casulo em forma de X. Após um período de 5 a 26 dias, emerge o adulto. Daí, surgem novas mariposas que repetem o ciclo de reprodução. A Figura 26 apresenta as etapas de desenvolvimento do bicho mineiro.



Figura 26 – Mariposa (A); danos causados pelo bicho mineiro (B); e casulo em forma de X (C). Fonte: Adaptado de (FORNAZIER *et al.*, 2017).

A.1.2 Outras pragas

Outras pragas de importância econômica que atingem as folhas de café são os ácaros e as formigas. A reprodução dos ácaros afeta o brilho e desenvolvimento foliar, provocando até o desfolhamento do cafeeiro. As formigas cortam as folhas das plantas, atrasando o seu desenvolvimento ou mesmo provocando sua morte.

Além das pragas que atingem diretamente as folhas do cafeeiro, existem aquelas que atacam os frutos e as raízes da planta. A principal praga que atinge o cafeeiro conilon é a broca-do-café. Esta praga ataca os frutos em todos os estágios de maturação, sendo responsável por grandes perdas na produtividade. A cochonilha é uma praga que se fixa nos ramos e folhas sugando a seiva das plantas. Há também aquelas que são consideradas pragas secundárias, com ocorrência esporádica, são elas: largarta-dos-cafezais, carneirinhos, broca-das-hastes, caruncho-das-tulhas e traças-do-café.

A.2 Doenças

As doenças que atingem o café se apresentam como um fator limitante para a sua produção. As perdas causadas por doenças podem chegar a inviabilizar o cultivo dessa cultura. As doenças que atingem o cafeeiro podem ser de origem biótica (fungos, bacterias, nematoide e vírus) ou abiótica (relacionado a fatores ambientais, deficiências nutricionais ou manejo inadequado das lavouras). Afetam todas as partes da plantas levando a perdas significativas.

Apenas as principais doenças de origem biótica foram exploradas neste trabalho. Detalhes de cada uma delas são apresentados nas subseções a seguir.

A.2.1 Ferrugem

A ferrugem é a principal doença do cafeeiro. Constatada pela primeira vez em 1970 no Brasil, essa doença já atinge hoje toda a América latina. É causada pelo fungo biotrófico (sobrevive apenas em tecido vivo) *H. vastatrix*. A sua ocorrência se dá em maior ou menor severidade em função das condições climáticas, adubações desequilibradas e da resistência dos cultivares plantados. A ferrugem causa principalmente desfolha precoce e seca dos ramos produtivos, levando a perdas de até 45% em condições favoráveis à doença.

Os sintomas se manifestam principalmente na face abaxial da folha como manchas amareladas translúcidas de 1 a 2mm de diâmetro. As manchas evoluem rapidamente e em pouco tempo formam pústulas circulares, pulverulentas, de cor amarela/alaranjada, com aspecto de pó amarelado. Com o aumento excessivo das lesões, o centro adquire aspecto necrótico. Na Figura 27, pode-se visualizar como é o aspecto de uma pústula de ferrugem na face abaxial da folha.



Figura 27 – Pústula de ferrugem na face abaxial da folha de café. Fonte: Adaptado de (VENTURA; COSTA; LIMA, 2017).

A.2.2 Mancha-de-phoma

A mancha-de-phoma ocorre na maioria das lavouras, mas com intensidade variável, em função das condições climáticas e altitude, provocando desfolha e a morte das mudas. A mancha-de-phoma tem como agente etiológico o fungo *Phoma tarda*.

As folhas infectadas apresentam manchas de cor escura que normalmente se iniciam pelas bordas da folha, como observado na Figura 28. A ocorrência da doença é favorecida por temperaturas entre 16 e 20 °C, ventos frios e chuvas finas.



Figura 28 – Sintomas de mancha-de-phoma em mudas de café. Fonte: (VENTURA; COSTA; LIMA, 2017).

A.2.3 Cercosporiose

Também conhecida como mancha-de-olho-pardo. Ocorre principalmente nas condições de viveiro e na fase de transplântio no campo. Fatores abióticos, como desequilíbrio nutricional, déficit hídrico e até solos com textura inadequada para o cultivo do café, favorecem o aumento da incidência e severidade da doença. Além disso, temperaturas entre 20 e 25 °C, associada com alta umidade do ar, aceleram seu crescimento.

O agente causador da doença é o fungo *Cercospora coffeicola*. Os sintomas se apresentam como manchas circulares de coloração marrom-escuro ou pardo-clara que normalmente possuem o centro acinzentado. Em torno da lesão é comum a presença de um halo amarelado. As folhas infectadas caem, causando desfolha das mudas e plantas.



Figura 29 – Sintomas de cercosporiose em plantas adultas de café conilon. Fonte: Adaptado de (VENTURA; COSTA; LIMA, 2017).

A.2.4 Outras doenças

Além das doenças descritas, há diversas outras que atingem as folhas e frutos do cafeeiro em maior ou menor frequência. Dentre aquelas com maior impacto econômico, podemos citar: mancha-manteigosa, mancha-de-corynespora, queima-do-fio e mancha-aureolada.

A mancha-manteigosa ocorre de forma esporádica, atualmente com baixa frequência devido à seleção de clones resistentes. Essa doença é capaz de provocar a morte das plantas com o passar do tempo. Os sintomas consistem de pequenas manchas (1 a 3mm) com aspecto oleoso e de bordas bem definidas. A mancha manteigosa também afeta os frutos, ocasionando lesões deprimidas, responsáveis pela queda desses frutos de maneira prematura.

A mancha-de-corynespora ou mancha-alvo é causada pelo fungo *Corynespora cassiicola*. Essa doença infecta folhas, flores e frutos de algumas variedades clonais do café conilon, causando desfolha em plantas adultas. Apresenta como sintoma manchas foliares de cor castanha com halo amarelado. Além de manchas amarronzadas nos ramos e nervuras.

A doença queima-do-fio, causada pelo fungo do gênero *Ceratobasidium*, geralmente ocorre em locais rodeados por florestas densas. Os sintomas aparecem nas folhas e ramos, que ficam cobertos pelo micélio do fungo, semelhante ao aspecto de teia de aranha.

Mancha-aureolada é uma doença causada pela bactéria *Pseudomonas syringae*. É mais frequente em viveiros com muita sombra e excesso de irrigação, principalmente se localizado em regiões frias. As manchas ocorrem nas folhas com coloração pardo-escura, muitas vezes os sintomas são confundidos com o de outras doenças, como aqueles causados por mancha-de-phoma.

APÊNDICE B

Aplicativo

Como produto deste trabalho, foi gerado um aplicativo para *smartphone*, multi-plataforma. O aplicativo desenvolvido, visa integrar toda a metodologia de segmentação e classificação de sintomas apresentada no escopo deste trabalho. De tal forma que o sistema permite à qualquer usuário com acesso a um *smartphone* e internet, usufruir de um *feedback* completo de quais doenças atingem a folha fotografada e também a severidade dos sintomas.

O desenvolvimento deste sistema baseou-se em tecnologias já utilizadas em trabalhos anteriores do laboratório, a experiência a priori com estas tecnologias facilitou a implementação da aplicação móvel (*frontend*) e também do servidor (*backend*). Devido à complexidade dos modelos de CNN utilizados, tornou-se inviável embarca-los totalmente em *smartphones*. Por isso, toda a etapa de processamento e classificação das imagens é realizada do lado do servidor. A Figura 30 apresenta os principais processos que compõem esta aplicação e como os dados trafegam entre eles.

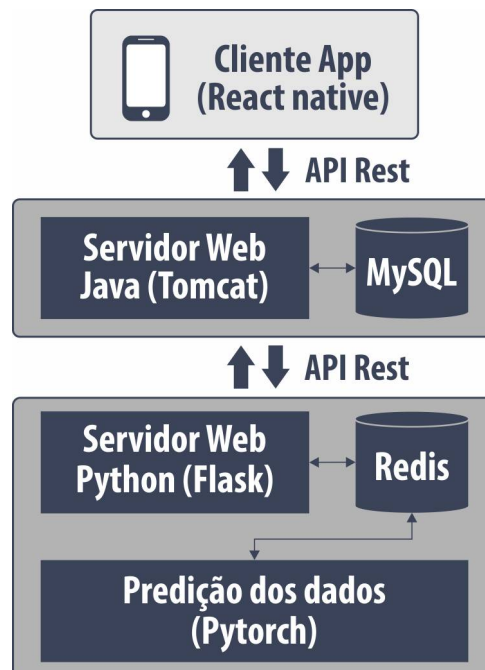


Figura 30 – Etapas de processamento do sistema que permite o acesso às predições dos modelos de *machine learning* pelo aplicativo.

Do lado do cliente temos uma aplicação desenvolvida utilizando o *framework* React Native, criado pelo Facebook, rodando sobre o Expo SDK. O uso do Expo facilita o acesso à recursos nativos dos celulares com um código javascript puro que pode rodar

em qualquer plataforma, seja Android ou IOS. A aplicação envia a foto para o servidor e periodicamente faz chamadas requisitando os resultados. Quando o processamento for finalizado a aplicação receberá uma resposta positiva do servidor e poderá consultar os resultados para a foto enviada.

A primeira camada do *backend* consiste em um servidor web java (Tomcat) que implementa uma API Rest para ser consumida pelo usuário em forma de serviço. Todos os dados, como informações do usuário, imagens etc, são armazenados no banco MySQL.

A segunda camada, consiste em um micro-framework de aplicações web chamado Flask. O Flask é totalmente baseado na linguagem python, o que permite uma execução direta dos modelos de *machine learning* desenvolvidos em python. Toda requisição para processamento de uma nova imagem que chega ao Flask é enfileirada no Redis (banco de dados NoSQL do tipo chave-valor). Se nenhuma imagem está sendo processada então o primeiro dado disponível na fila do Redis é lido e enviado a um modelo previamente treinado com a biblioteca Pytorch. O resultado do modelo é salvo no Redis que posteriormente poderá ser consultado pelo usuário.

B.1 Funcionalidades

Ao iniciar o aplicativo é exibida a sua tela inicial (Figura 31A). Nesta tela é onde o usuário deverá efetuar o *login* ou então realizar o seu cadastro. Após efetuado o *login*, o usuário é redirecionado para o menu principal do sistema mostrado na Figura 31B. Ao clicar na opção *biblioteca* surge uma tela contendo todo o histórico de fotos tiradas pelo usuário (Figura 31C), onde é possível consultar os resultados de processamentos anteriores.

No menu principal, ao clicar no ícone de uma câmera o usuário é redirecionado para a tela exibida na Figura 32A. Nesta tela o usuário captura a foto da folha e corta os excessos de imagem, evidenciando a região de interesse. Após a captura da foto o usuário é levado à tela seguinte onde deverá verificar se a foto obtida está adequada, em caso positivo, basta o usuário enviar a foto para o servidor clicando no botão *upload* (Figura 32B). Ao enviar a foto o usuário deverá aguardar o processamento realizado pelo servidor. Assim que o resultado estiver disponível ele será apresentado ao usuário conforme exibido na Figura 32C.

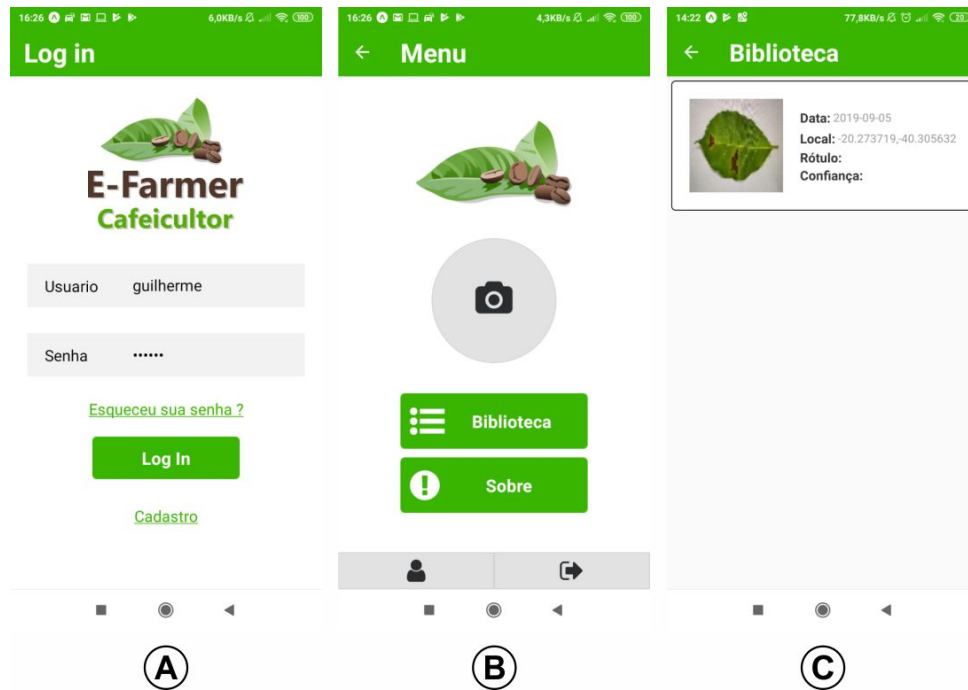


Figura 31 – Telas capturadas durante o uso da aplicação: login (A); menu (B); e biblioteca (C).

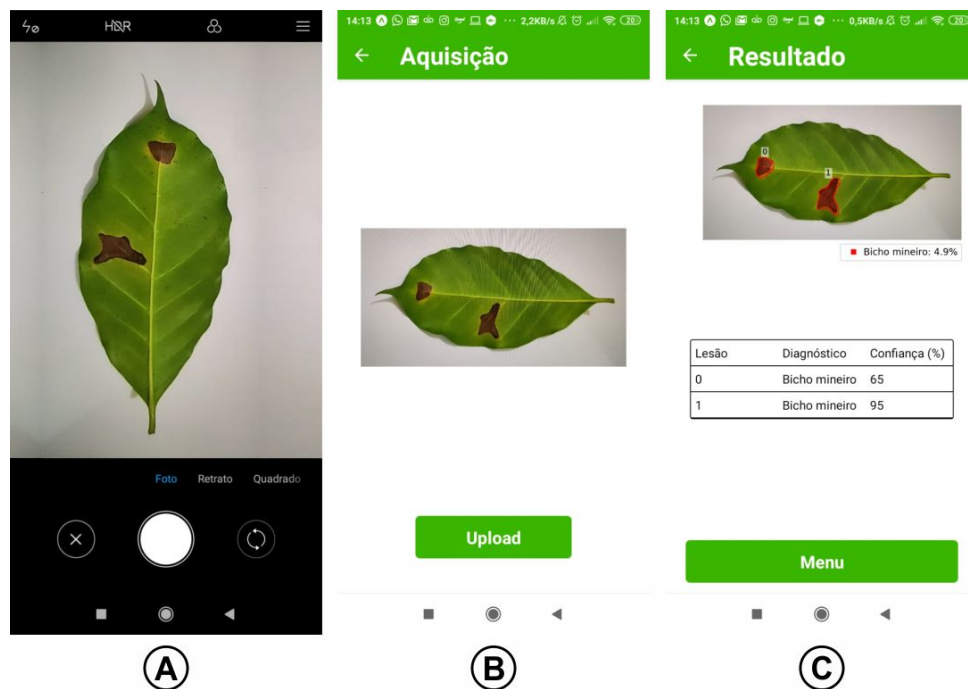


Figura 32 – Telas capturadas durante o uso da aplicação: câmera (A); verificação e *upload* (B); e resultado (C).