

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO**

**FELIPE CARVALHO FUNCK**

**DETECTANDO A FERRUGEM ASIÁTICA NA FOLHA DA SOJA  
UTILIZANDO REDES NEURASIS CONVOLUCIONAIS**

**TRABALHO DE CONCLUSÃO DE CURSO**

**Santa Helena, Paraná**

**2019**

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO**

FELIPE CARVALHO FUNCK

**DETECTANDO A FERRUGEM ASIÁTICA NA FOLHA DA SOJA  
UTILIZANDO REDES NEURAIS CONVOLUCIONAIS**

Trabalho de Conclusão apresentado ao Curso de Ciência da Computação da Universidade Tecnológica Federal do Paraná, Câmpus Santa Helena, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Giuvane Conti

Santa Helena, Paraná

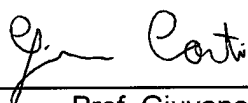
2019

## TERMO DE APROVAÇÃO

### “DETECTANDO A FERRUGEM ASIÁTICA NA FOLHA DA SOJA UTILIZANDO REDES NEURAIAS CONVOLUCIONAIS”

por  
“Felipe Carvalho Funck”

Este Trabalho de Conclusão de Curso foi apresentado às 14 horas do dia 27 de junho de 2019 na sala K-10 como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação na Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Santa Helena. O(a) aluno(a) foi arguido pela Banca de Avaliação abaixo assinados. Após deliberação, a Banca de Avaliação considerou o trabalho aprovado.



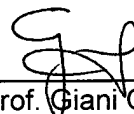
Prof. Giuvane Conti  
(Presidente - UTFPR/Santa Helena)




Prof. Davi Marcondes Rocha  
(Avaliador 1 - UTFPR/Santa Helena)



Prof. Suzan Kelly Borges Piovezan  
(Avaliador 2 - UTFPR/Santa Helena)



Prof. Giani Carla Ito  
(Professor responsável pelo TCC - UTFPR/Santa Helena)



Prof. Arlete Teresinha Beuren  
(Coordenador do curso de Bacharelado em  
Ciência da Computação - UTFPR/Santa Helena)

Dedico este trabalho à minha família, pelos  
momentos de ausência.

## **AGRADECIMENTOS**

Agradeço aos meus amigos, familiares, professores e todos os demais membros desta instituição de ensino que me ajudaram durante esses anos e na realização deste trabalho.

A inquietude é o estímulo essencial à pesquisa científica.

Anderson Vailati Ritzmann

## RESUMO

FUNCK, Felipe Carvalho. **Detectando a ferrugem asiática na folha da soja utilizando redes neurais convolucionais**. 2019. 66f.. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Tecnológica Federal do Paraná. Santa Helena.

A cultura da soja é destacadamente o principal cultivo agrícola do Brasil. A principal doença responsável por atingir esta cultura e diminuir o potencial de produção da planta no Brasil é a ferrugem asiática da soja. Este trabalho apresenta uma solução de um aplicativo mobile para a plataforma Android que é capaz de identificar a ferrugem asiática da soja quando ela estiver presente na folha da planta utilizando a câmera de um dispositivo Android, desta forma, auxiliando os produtores rurais a realizarem o tratamento da doença nas lavouras de soja o mais rápido possível. O trabalho foi desenvolvido aplicando técnicas de visão computacional e um modelo de Rede Neural Convolucional para realizar a classificação de imagens. A biblioteca TensorFlow foi utilizada durante todo o processo de desenvolvimento do aplicativo, desde o processo de treinamento da Rede Neural Artificial, até a etapa de identificação da doença em tempo real na folha da soja. Para realizar as classificações de imagens no aplicativo foi utilizado o modelo de Rede Neural Convolucional Inception-V3, um modelo famoso e conhecido por possuir ótimos resultados em problemas de *Deep Learning* e classificações de imagens. Ao total foram reunidas 2770 imagens de folhas distintas de soja para o desenvolvimento deste trabalho, sendo divididas entre imagens para realização do treinamento da rede neural e imagens de testes para validação do aplicativo. Ao final dos testes realizados o aplicativo desenvolvido atingiu uma acurácia de 84,61% na detecção da ferrugem asiática na folha da soja. Ao final dos testes realizados o modelo treinado se demonstrou eficiente na separação de folhas de soja saudáveis e com a ferrugem asiática da soja, também foram apresentadas algumas dificuldades em distinguir outras doenças da cultura da soja, mesmo com os resultados obtidos em testes sendo satisfatórios.

**Palavras-chave:** Visão Computacional. TensorFlow. Redes Neurais Convolucionais. Ferrugem Asiática da Soja. Android.

## ABSTRACT

FUNCK, Felipe Carvalho. **Detecting Asian rust on soybean leaf using convolutional neural network**. 2019. 66p. Work of Conclusion Course (Graduation in Computer Science) – Federal Technology University – Paraná. Santa Helena.

Soybean cultivation is the main agricultural crop in Brazil. The main disease responsible for reaching this crop and decreasing the production potential of the plant in Brazil is the Asian soybean rust. This work presents a solution of a mobile application for the Android platform that is able to identify the Asian soybean rust when it is present in the leaf of the plant using the camera of an Android device, in this way, helping the rural producers to carry out the treatment of the disease in the soybean plantations as soon as possible. The work was developed applying computational vision techniques and a Convolutional Neural Network model to perform image classification. The TensorFlow library was used throughout the entire application development process from the artificial neural network training process to the real-time identification of the disease in the soybean leaf. Inception-V3 Convolutional Neural Network model was used to perform in-app image classifications, a famous model known for having great results in Deep Learning problems and image classifications. In total, 2770 images of different leaves of soybean were collected for the development of the work, being divided between images for performing neural network training and imaging tests for application validation. At the end of the tests, the developed application reached an accuracy of 84.61% in the detection of Asian rust on soybean leaf. At the end of the tests, the trained model was shown to be efficient in the separation of healthy soybean leaflets and Asian soybean rust, some difficulties were also observed in distinguishing other diseases from the soybean crop, even with the results obtained in satisfactory.

**Keywords:** Computer Vision. TensorFlow. Convolutional Neural Network. Asian Soybean Rust. Android.



## LISTA DE ILUSTRAÇÕES

Figura 1: Sintomas iniciais da ferrugem asiática na folha da soja.....	17
Figura 2: Lesões causadas pela ferrugem asiática.....	18
Figura 3: Representação da sequência do processamento de imagens.....	19
Figura 4: Célula neural biológica com a sequência de propagação do sinal.....	22
Figura 5: Modelo artificial de neurônio biológico.....	23
Figura 6: Diagrama em blocos da aprendizagem com um professor.....	24
Figura 7: Arquitetura de uma MLP.....	26
Figura 8: Arquitetura de uma CNN.....	27
Figura 9: Visão geral das atividades desenvolvidas.....	31
Figura 10: Grupo 1 da base de imagens.....	32
Figura 11: Grupo 2 da base de imagens.....	33
Figura 12: Extração de texturas das folhas de soja.....	35
Figura 13: Conversão realizada para escala de cinza.....	35
Figura 14: Arquitetura do Inception-V3.....	36
Figura 15: Evolução da acurácia e entropia cruzada durante a etapa 1 de treinamento.....	41
Figura 16: Evolução da acurácia e entropia cruzada durante a etapa 2 de treinamento.....	41
Figura 17: Evolução da acurácia e entropia cruzada durante a etapa 3 de treinamento.....	41
Figura 18: Evolução da acurácia e entropia cruzada durante a etapa 4 de treinamento.....	42
Figura 19: Evolução da acurácia e entropia cruzada durante a etapa 5 de treinamento.....	42
Figura 20: Evolução da acurácia e entropia cruzada durante a etapa 6 de treinamento.....	42
Figura 21: Diagrama de classes do aplicativo Android.....	44
Figura 22: Fluxo de atividades desenvolvidas para classificação de uma imagem.....	46
Figura 23: Interfaces de interação com usuário presentes no aplicativo.....	46
Figura 24: Intervalo mínimo e máximo nas classificações de folhas contaminadas com ferrugem e saudáveis.....	53
Figura 25: Intervalo mínimo e máximo nas classificações de folhas contaminadas com mancha parda, oídio da soja, olho de rã e manchas causadas por outras doenças.....	54
Figura 26: Intervalo mínimo e máximo nas classificações de folhas contaminadas com a ferrugem e com folhas não contaminadas com a ferrugem.....	54

Figura 27: Classificações de folhas com ferrugem e saudáveis com base na média entre valor mínimo e máximo capturado na análise de cada folha .....	55
Figura 28: Classificações de folhas com mancha parda, oídio da soja, olho de rã e manchas causadas por outras doenças com base na média entre valor mínimo e máximo capturado na análise de cada folha.....	56
Figura 29: Classificações gerais realizadas de folhas com e sem ferrugem com base na média entre valor mínimo e máximo capturado na análise de cada folha.....	56

## LISTA DE TABELAS E QUADROS

Tabela 1: Treinamentos realizados.....	40
Tabela 2: Resultados obtidos nos treinamentos realizados.....	40
Tabela 3: Resultados gerais obtidos em testes divididos por cada classe de imagem.....	57
Tabela 4: Acurácia e taxa de erro obtidos em testes dividido por cada classe de imagem.....	58
Tabela 5: Resultados obtidos sobre as classificações realizadas.....	58
Quadro 1: Tecnologias e ferramentas utilizadas para o treinamento do Inception-V3.....	38
Quadro 2: Parâmetros estabelecidos ao Inception-V3 durante os treinamentos realizados...	39
Quadro 3: Ferramentas e tecnologias utilizadas no desenvolvimento do aplicativo.....	43
Quadro 4: Especificações do dispositivo em que a aplicação foi desenvolvida.....	44
Quadro 5: Classes de folhas utilizadas no processo de testes.....	50
Quadro 6: Termos atribuídos aos resultados das classificações.....	50

## **LISTA DE ABREVIATURAS E SIGLAS**

CNN	Convolutional Neural Network
RNA	Redes Neurais Artificiais
CPU	Central Processing Unit
API	Application Programming Interface
MLP	Multilayer Perceptron
SDK	Software Development Kit
JDK	Java Development Kit
VANT	Veículo Aéreo Não Tripulado
RGB	Red Green Blue
GHz	Gigahertz
MB	Megabytes
MP	Megapixels
PB	Protocol Buffers
OO	Orientação a Objetos
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	12
1.1 OBJETIVOS .....	13
1.1.1 Geral .....	13
1.1.2 Específicos .....	13
1.2 CONTRIBUIÇÕES DO TRABALHO .....	14
1.3 JUSTIFICATIVA .....	14
1.4 DELIMITAÇÕES DO TRABALHO .....	15
<b>2 REVISÃO DA LITERATURA</b> .....	16
2.1 CULTURA DA SOJA NO BRASIL .....	16
2.1.1 Ferrugem Asiática da soja .....	16
2.2 VISÃO COMPUTACIONAL .....	18
2.2.1 Aquisição de Imagem e Técnicas de Pré-processamento .....	19
2.2.2 Segmentação .....	20
2.2.3 Extração de atributos ou características .....	21
2.2.4 Reconhecimento de Padrões .....	21
2.3 REDES NEURAIAS ARTIFICIAIS .....	22
2.3.1 Perceptron Multicamadas .....	25
2.3.2 Redes Neurais Convolucionais .....	26
2.4 TENSORFLOW .....	28
2.5 ESTADO DA ARTE .....	28
<b>3 METODOLOGIA</b> .....	31
3.1 BASE DE IMAGENS .....	31
3.2 PRÉ-PROCESSAMENTO DIGITAL DAS IMAGENS .....	33
3.3 ARQUITETURA E TREINAMENTO DA RNA .....	36
3.4 DESENVOLVIMENTO DO APLICATIVO .....	43
3.4.1 Interfaces e Funcionalidades do Aplicativo .....	45
3.4.2 Aquisição de Imagem .....	47
3.4.3 Classificação das Imagens .....	48
3.5 APLICAÇÃO DOS TESTES E MÉTRICAS DE AVALIAÇÃO .....	49
<b>4 ANÁLISE DE RESULTADOS</b> .....	53
<b>5 CONCLUSÃO</b> .....	60
<b>REFERÊNCIAS</b> .....	62

# 1 INTRODUÇÃO

Atualmente, a cultura agrícola vem adotando diversas tecnologias em algumas de suas subáreas, tendo como propósito a maximização do potencial de produção das lavouras e consequentemente a obtenção de lucros maiores. Entre as tecnologias atuais presentes atualmente na agricultura, pode-se destacar as soluções computacionais, irrigação automatizada do solo, além de utilização de fertilizantes e técnicas que estimulam uma maior produção nas lavouras. Uma das tecnologias computacionais presentes neste meio é a utilização de ferramentas computacionais para detecção de doenças e pragas fúngicas, nas quais são aplicadas técnicas como visão computacional, aprendizado de máquina e reconhecimento de padrões.

A cultura agrícola de maior crescimento econômico no Brasil é o plantio da soja, o que torna o Brasil o segundo maior produtor mundial de soja, atrás somente dos Estados Unidos (USDA, 2018). A produção da soja passa por diversos desafios e problemas durante o seu período de cultivo. De dias após o seu plantio, até algumas semanas antes de sua colheita, a planta está sujeita a contaminação por diversas pragas e doenças que podem levar à perda completa de uma lavoura ou a perda do potencial de produção das plantas, caso tais pragas e doenças não sejam tratadas corretamente a tempo.

Uma das principais doenças responsáveis por levar prejuízos as lavouras de soja é a ferrugem asiática, doença causada pelo fungo *Phakopsora Pachyrhizi*, sendo a rápida propagação um dos fatores mais preocupantes sobre essa doença. O principal dano de tal doença é a desfolha precoce, o que impede a formação completa dos grãos, consequentemente a redução na produtividade da soja.

Erradicar a ferrugem asiática dentro da cultura da soja é algo que não é possível, entretanto é possível evitar que a doença atinja altos índices de incidência e ocasione grandes perdas no rendimento da cultura. Um dos principais meios de controlar a doença é a aplicação de fungicidas no início do aparecimento dos sintomas. Para isto, o quanto antes a doença for identificada, maior será a chance de sucesso no combate à doença. O problema pode se agravar quando alguns agricultores podem ter a dificuldade de identificar a doença, necessitando da ajuda de algum profissional qualificado da área. A principal dificuldade encontrada em identificar a ferrugem asiática na folha da soja é o fato de que os danos visuais causados pela

doença serem muitos semelhantes aos danos causados por outras doenças fúngicas da cultura da soja (YORINORI; NUNES JUNIOR e LAZZAROTTO, 2004).

A partir disso, foi construído uma aplicação móvel para a plataforma Android, utilizando Redes Neurais Convolucionais (Convolutional Neural Network - CNN), para auxiliar os agricultores a identificarem a ferrugem asiática quando ela estiver presente na folha da soja. Tal aplicativo foi construído utilizando uma interface amigável de fácil uso ao usuário. A identificação da doença na folha da soja ocorre utilizando a câmera do celular, através do aplicativo desenvolvido, onde o aplicativo extrai características da folha da soja, em tempo real, e expõe os dados a uma Rede Neural Artificial (RNA) para que seja realizada a classificação e detecção da ferrugem asiática, caso exista a doença naquela folha. Para o desenvolvimento do aplicativo foi utilizado a ferramenta TensorFlow, uma biblioteca de código aberto para aprendizado de máquina. Ela foi utilizada para criação e treinamentos do modelo de RNA utilizado neste projeto para detectar e correlacionar padrões.

## 1.1 OBJETIVOS

Expõem-se a seguir os objetivos geral e específicos que este trabalho buscou atingir.

### 1.1.1 Geral

Construir uma aplicação para a plataforma Android, utilizando CNN, a qual identifica e classifica folhas de soja que estão contaminadas com a ferrugem asiática da soja.

### 1.1.2 Específicos

- 1) Criar uma base de dados contendo imagens de folhas de sojas saudáveis, contaminadas pela ferrugem asiática da soja e folhas contaminadas com outras doenças da cultura da soja;
- 2) Treinar um modelo de CNN com TensorFlow e integrá-lo com o aplicativo Android desenvolvido;

- 3) Construir um classificador capaz de classificar se uma folha de soja está contaminada com a ferrugem asiática;
- 4) Criar um aplicativo Android com aquisição de imagem em tempo real e integrá-lo com o classificador desenvolvido;
- 5) Aplicar metodologias de testes e métricas de avaliação para avaliar o classificador de imagens desenvolvido;

## 1.2 CONTRIBUIÇÕES DO TRABALHO

No ponto de vista computacional, espera-se que os recursos teóricos e práticos que serão realizados no desenvolvimento deste aplicativo, como o estudo da biblioteca TensorFlow, a utilização de visão computacional para reconhecimento de uma doença fúngica, RNAs e classificação de imagens possam auxiliar a comunidade acadêmica e científica na construção de novas aplicações deste gênero.

Espera-se com a implementação deste aplicativo ajudar os produtores rurais que cultivam soja a detectarem a doença ferrugem asiática quando ela estiver presente na folha da soja. Assim, com este aplicativo, deseja-se reduzir a perda da produção da soja devido ao retardo na identificação da doença, o que é um dos principais motivos que faz esta doença causar tantas perdas na cultura da soja no Brasil.

## 1.3 JUSTIFICATIVA

A construção deste trabalho implicará diretamente na utilização de recursos e técnicas computacionais aplicadas em uma das áreas de maior importância econômica no Brasil e no mundo, a agricultura. O TensorFlow por se tratar de uma biblioteca recente, ainda não foi aplicado em muitos trabalhos científicos desta área, principalmente na região em que esse estudo foi realizado. Partindo deste princípio, a justificativa para a utilização de tal ferramenta na construção deste trabalho é mostrar uma nova tecnologia e possibilidades de desenvolvimento de aplicações nesta área.

Os dispositivos móveis estão cada vez mais presentes na vida das pessoas, hoje em dia é comum quase todas as pessoas terem um dispositivo móvel ao seu lado. Atualmente o sistema



operacional para dispositivos móveis mais utilizado no mundo é o Android, cerca de 1.3 bilhões de aparelho em todo o mundo (GARTNER, 2018). Tal número de aparelhos executando o sistema operacional Android, assim como a grande proximidade que existe atualmente entre os celulares e as pessoas influenciaram a escolha de construir uma aplicação móvel e utilizar a plataforma Android para o desenvolvimento deste projeto.

#### 1.4 DELIMITAÇÕES DO TRABALHO

O aplicativo desenvolvido está limitado a analisar se uma folha de soja está contaminada com a ferrugem asiática da soja ou não e exibir ao usuário graus de confiança sobre o resultado da classificação realizada. Devido ao ambiente de utilização do aplicativo desenvolvido possuir uma alta variação de iluminação e o fator de variação da qualidade da câmera utilizada pelo usuário, os resultados obtidos por este projeto serão inferiores a outros projetos que se propõem a identificar a ferrugem asiática na folha da soja, porém em um ambiente controlado, como imagens fotografadas ou digitalizadas.

O período em que o desenvolvimento do aplicativo foi finalizado não havia lavouras de soja em produção no Brasil para que os testes pudessem ser realizados em campo. Por este motivo, todos os testes realizados com o aplicativo desenvolvido aconteceram em um ambiente controlado, com imagens coletadas para o desenvolvimento deste trabalho. Os resultados obtidos nos testes realizados para este trabalho poderão sofrer variações quando o aplicativo for testado no campo nas lavouras de soja. O motivo é a diferente variedade de iluminação do local, e a maior dificuldade de realizar capturas de imagens com um maior nível de detalhes nestes locais.

## 2 REVISÃO DA LITERATURA

### 2.1 CULTURA DA SOJA NO BRASIL

A soja, de acordo com Sanches, Michellon e Roessing (2004), é uma leguminosa cultivada pelos chineses há cerca de cinco mil anos. A soja constitui um dos principais cultivos da agricultura mundial. O grão da soja é de extrema importância econômica no mundo, muito pelo seu alto teor de proteína (40%), o que atribui ao grão da soja várias aplicações alimentares importantes tanto para seres humanos quanto para animais. O grão da soja também possui um teor de óleo de 19%, o que o torna a planta uma das principais produtoras de óleo vegetal no mundo (DALL'AGNO et al., 2007). O grão da soja também é matéria-prima para diversos complexos industriais, o que a atribui um importante papel socioeconômico (STRECK et al, 2008).

No ano de 2018 a soja teve uma produção mundial de aproximadamente 336,699 milhões de toneladas, com uma área total plantada de 124,589 milhões de hectares. Os Estados Unidos foi o país que mais produziu soja no ano de 2018 com cerca de 119,518 milhões de toneladas, seguido pelo Brasil, que produziu 116,996 milhões de toneladas neste mesmo ano, sendo que no Brasil, os estados que mais produziram foram o Mato grosso, Paraná e Rio Grande do Sul com 31,887, 19,070 e 16,968 milhões de toneladas respectivamente (EMBRAPA, 2018).

A cultura da soja sofre com ataques de diversas pragas e fungos. Existem cerca de 40 doenças que afetam a cultura, da mais diversificada etiologia, entre as mais conhecidas está a Ferrugem Asiática, doença causada pelo fungo *Phakopsora Pachyrhizi* (FREITAS, 2011).

#### 2.1.1 Ferrugem Asiática da soja

Atualmente a ferrugem asiática da soja é a principal responsável pelos danos causados na cultura da soja no Brasil. Tal doença atingiu lavouras brasileiras pela primeira vez no ano de 2001, firmando-se como o principal problema daquela safra, atingido uma área de 22 milhões de hectares em todo o país (CAMARGOS, 2017).

Os principais sintomas gerados pela ferrugem asiática são identificados na folha da soja, são denominadas “lesões”. Geralmente, a doença se inicia pelas folhas localizadas nas partes mais baixas da planta. Os primeiros sintomas são caracterizados por minúsculos pontos escuros (no máximo com 1mm de diâmetro) localizadas no tecido sadio da folha, possuindo uma coloração esverdeada a cinza-esverdeada (Figura 1). Essas lesões, provenientes da fase inicial da infecção, correspondem à formação de protuberância, que se apresentam como pequenas saliências na lesão (EMBRAPA, 2011).



Figura 1 - Sintomas iniciais da ferrugem asiática na folha da soja

**Fonte:** Agrolink (2016).

Em seu estágio mais avançado, as lesões ocasionadas pela doença atingem de 2mm a 5mm de diâmetro, podendo aparecer nos pecíolos, vagens e caules. Posteriormente, as urédias adquirem uma tonalidade de cor entre castanho claro e castanho escuro (Figura 2). Neste período há a abertura de um minúsculo poro, soltando os chamados uredosporos, que são esporos causados pelo fungo gerador da ferrugem. As lesões geradas pela ferrugem asiática na soja podem também ser facilmente confundida com lesões causadas por outras doenças na cultura da soja, como as lesões geradas pela mancha parda (*Septoria Glycines Hemmi*), doença que forma um halo amarelo ao redor da lesão necrótica, que é angular e castanho. O fato de os danos visuais causados pela ferrugem asiática serem semelhantes aos causados por outras doenças populares da cultura da soja é uma das principais dificuldades no seu reconhecimento (YORINORI; NUNES JUNIOR e LAZZAROTTO, 2004).



Figura 2 - Lesões causadas pela ferrugem asiática

**Fonte:** Agrolink (2016).

## 2.2 VISÃO COMPUTACIONAL

De acordo com Vianna (2009), a visão computacional é uma importante e promissora tecnologia de sensoriamento, aplicada a sistemas robóticos e de automação, uma vez que tenta reproduzir em máquinas um dos sentidos mais complexos, a visão. Uma das principais propostas da visão computacional é auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e habilidade do ser humano em tomar decisões de acordo com as informações contidas em imagens. A principal capacidade da visão computacional é a de extrair informações relevantes a partir de imagens capturadas por câmeras fotográficas, vídeos, sensores, entre outros dispositivos, para automatizar a tomada de decisão em um outro sistema (BORTH et al. Apud SHAPIRO, 2014).

A visão computacional é dividida em algumas subáreas importantes que podem incluir aquisição de imagens, pré-processamento de imagens, segmentação de imagens, extração de atributos ou características e reconhecimento de padrões (FORSYTH e PONCE, 2011). A partir das técnicas de pré-processamento e análise de imagens é possível realizar a construção de sistemas de visão computacional utilizando as características extraídas das imagens digitais (FORSYTH e PONCE, 2011).

Os processos presentes em um sistema de visão computacional são representados como uma estrutura sequencial (Figura 3), sendo que a estrutura segue a seguinte ordem: aquisição, pré-processamento, segmentação, normalização das características de interesse, classificação e

reconhecimento de padrões, e por último, a resolução do problema. Todas essas etapas são ligadas por uma base de conhecimento gerada pelos resultados obtidos nos processos anteriores. Esta estrutura sequencial ainda pode ser dividida em três estágios: visão em baixo nível (aquisição e pré-processamento); visão em nível intermediário (segmentação, extração e normalização de características); visão em alto nível (classificação, reconhecimento e correspondência) (FORSYTH e PONCE, 2011).

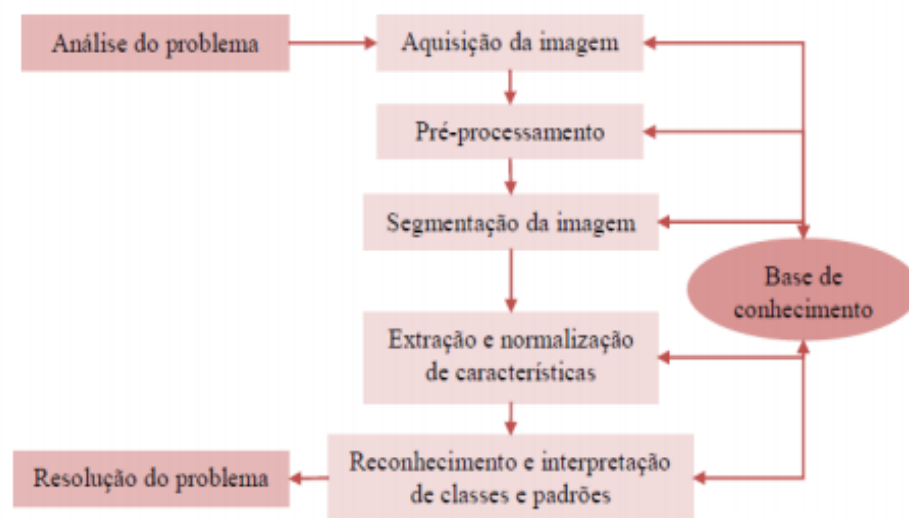


Figura 3 - Representação da sequência do processamento de imagens

**Fonte:** Cássio (2011).

### 2.2.1 Aquisição de Imagem e Técnicas de Pré-processamento

A etapa de aquisição de imagens é o processo inicial em um sistema de visão computacional. O principal objetivo nesta etapa é realizar a captura, armazenamento e transmissão de uma imagem. Para a captura de imagens, diversos dispositivos diferentes podem ser utilizados, por exemplo, câmeras digitais, celulares, *Smartphones*, *Tablets*, infravermelhos, câmeras de segurança normais ou térmicas etc. (BORTH et al., 2014).

As técnicas de pré-processamento têm como principal objetivo melhorar a qualidade da imagem. Tais técnicas são divididas em duas categorias principais, que são os métodos que operam no domínio espacial e os métodos que operam no domínio da frequência. As técnicas

de processamento no domínio espacial baseiam-se em filtros que manipulam o plano da imagem, enquanto que as técnicas de processamento no domínio da frequência se baseiam em filtros que agem sobre o espectro da imagem (ALBUQUERQUE et al., 2012).

### 2.2.2 Segmentação

Segundo Borth et al. (2014), a etapa de segmentação é responsável por separar a imagem em regiões de seu interesse baseando-se no problema a ser resolvido, como por exemplo, a separação de um objeto específico e fundo da imagem. Neste sentido, segmentar uma imagem significa separá-la em regiões com propriedades comuns, as quais correspondem a objetos ou partes de objetos na mesma imagem ou mesmo ao fundo (*Background*) dessa imagem (CHAGAS e RODRIGUES, 2018).

Um das técnicas mais utilizadas para segmentação de imagem, e o chamado modelo de contornos ativos, também conhecida como *Snakes* em Visão Computacional. Este método se caracteriza como um recurso bem-sucedido na solução de problemas relacionados com a segmentação de imagens, pois além de contarem com informações locais da imagem envolvida, pode incorporar outras informações, tais como a topologia e suavidade do objeto a ser segmentado (CHAGAS e RODRIGUES, 2018). Além do modelo de contornos ativos, duas técnicas de segmentação de imagens bem conhecidas que podem ser citadas são as técnicas de Watershed (Bacia hidrográfica) e Graph Cut (Corte de gráfico) (ANANTRASIRICHAI; HANNUNA e CANAGARAJAH, 2017).

De acordo com Pinto (2014) a técnica de segmentação de imagens Watershed é um modelo de segmentação de imagens que se baseia no crescimento de regiões, sendo a transformação Imagem-Floresta uma eficiente forma de implementá-la. Andrade (2011) afirma que a grande vantagem do método Watershed é a sua velocidade no processamento para segmentação de imagens. Andrade (2011) também afirma que uma das principais desvantagens desse método é que existe uma alta sensibilidade a variações de imagens, o que pode prejudicar o resultado nas segmentações de imagens em alguns casos.

Ferreira apud Shi e Malik (2011) define a técnica de corte de grafos como sendo a remoção de um determinado conjunto de arestas de um grafo a qual resultara em subconjuntos independentes. Ferreira (2011) divide a técnica de corte em grafos em duas categorias principais: A técnica baseada no corte mínimo e a técnica de corte normalizado.

### 2.2.3 Extração de atributos ou características

A etapa de extração de características de uma imagem busca encontrar um conjunto de características do objeto de interesse que seja consonante com o propósito da classificação desejada. Existem diversas técnicas que são utilizadas nesta etapa, no qual podem ser agrupadas em dois tipos: características externas (bordas) e características internas (*Pixels* que compõem o objeto). Esta etapa, na maior parte das vezes, tem como entrada de dados os resultados obtidos na etapa de segmentação (FRANCO, 2017).

Os resultados obtidos na etapa de extração de características são utilizados na etapa seguinte, na qual a interpretação e o reconhecimento buscados são processados, atribuídos rótulos e significados aos conjuntos de objetos identificados ao longo de todo o processo (FRANCO apud MARQUES FILHO; VIEIRA NETO, 1999).

### 2.2.4 Reconhecimento de Padrões

A tarefa de reconhecer objetos, faces, cenas, defeitos, estruturas ou qualquer tipo de coisa que possa ser classificada de alguma forma em uma imagem é uma das principais tarefas da visão computacional e está diretamente relacionado com o reconhecimento de padrões (BORTH et al., 2014). Segundo Cavani (2007), as técnicas de reconhecimento de padrões são usadas para extrair informações que tenham maior valor semântico, segundo critérios humanos. Tais informações, com maior valor semântico, podem ser empregadas em diversos sistemas automáticos que devem abstrair a informação (aprender) com o objetivo de viabilizar uma interação sofisticada da máquina com o meio (MARENGONI, 2009).

De acordo com Cássio (2011), as metodologias de reconhecimento de padrões se dividem em duas categorias: a primeira, denominada de decisão teórica, abrange os padrões de descritores quantitativos; a segunda, denominada de decisão estrutural, abrange os padrões de descritores qualitativos. Ambas são usadas para extrair informações semânticas segundo algum critério. A visão computacional utiliza as técnicas de reconhecimento de padrões para identificar imagens, elementos da imagem ou a interação entre os elementos em uma imagem. A interação entre os elementos da imagem deve ser analisada usando uma abordagem descritiva (CAVANI, 2007).

### 2.3 REDES NEURAIS ARTIFICIAIS

Uma Rede Neural Artificial (RNA) é um sistema de processamento de informação que possui algumas características de desempenho em comum com as redes neurais biológicas. Os modelos neurais artificiais têm como principal fonte de inspiração as redes neurais biológicas. A natureza das RNAs faz com que seu estudo seja multidisciplinar, envolvendo pesquisadores de diversas áreas, como neurofisiologia, psicologia, física, computação e engenharia (SILVA, 1998). A Figura 4 representa um modelo de um neurônio biológico com a sequência de propagação dos sinais pela célula.

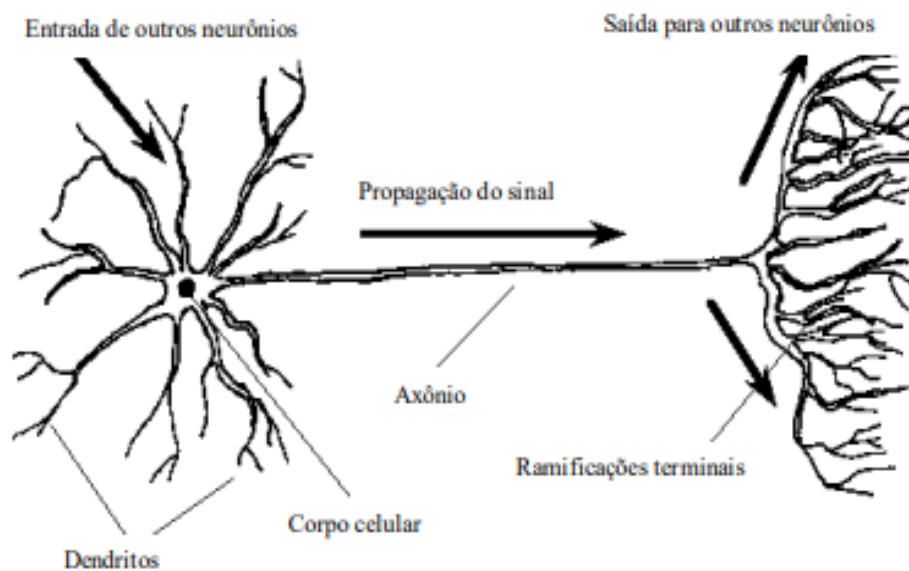


Figura 4. Célula neural biológica com a sequência de propagação do sinal

**Fonte:** Silva (1998).

Segundo Matsunaga (2012), as RNAs são formadas por um conjunto de neurônios artificiais que interagem entre si, semelhantes ao funcionamento dos neurônios biológicos. Basicamente, são sistemas paralelos de computação e suas implementações podem ser em hardware (realiza uma determinada tarefa a partir de componentes eletrônicos) ou em software (simulações por programação de um computador digital). As redes neurais possuem a capacidade de coletar, utilizar e armazenar informações baseadas em experimentos (aprendizagem). O processo de aprendizagem é realizado a partir de algoritmos de



aprendizagem, onde os pesos sinápticos da rede são modificados de uma forma ordenada para alcançar o resultado desejado (MATSUNAGA, 2012).

Todos os tipos de RNAs possuem a mesma unidade de processamento, um neurônio artificial, o qual tem a função de simular o comportamento de um neurônio biológico. O neurônio artificial pode conter diversas entradas, que correspondem as conexões sinápticas com outras unidades similares a ele, e uma saída. O valor de saída de um neurônio dependerá da somatória ponderada de todas as saídas de outros neurônios conectados a esse (MATSUNAGA, 2012). A Figura 5 representa um modelo artificial de neurônio biológico.

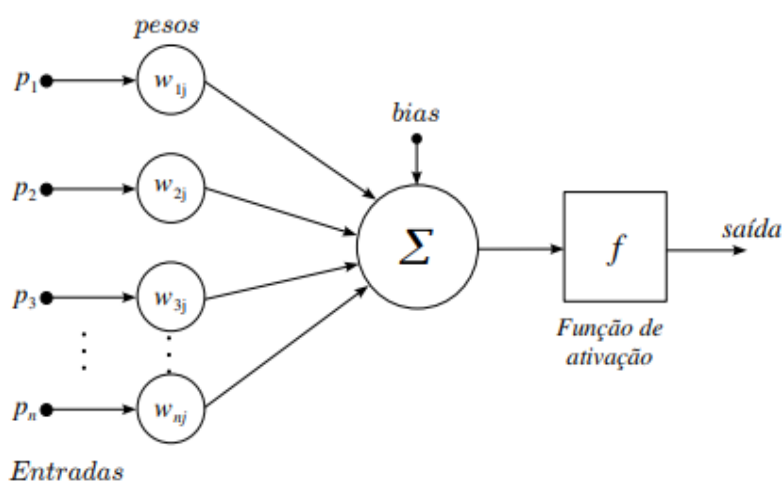


Figura 5. Modelo artificial de neurônio biológico

**Fonte:** Matsunaga (2012).

O modelo artificial de neurônio que é mostrado na Figura 5, inclui um sinal adicional *bias* ( $b$ ) que irá favorecer ou limitar a possibilidade de ativação do neurônio. O processo sináptico é representado pelos pesos ( $w$ ) que amplificam cada um dos sinais recebidos. A função de ativação ( $f$ ) é responsável por modelar a forma como o neurônio responde as entradas recebidas, limitando e definindo a resposta dada pela rede neural (MATSUNAGA, 2012).

As RNAs possuem como principais características a tolerância a falhas, adaptabilidade a novas condições, resoluções de problemas com base no conhecimento passado e reconhecimento de padrões. Existem diversos tipos de redes neurais e diferentes maneiras de classificá-las, sendo que talvez a forma mais importante de as classificar seja através da sua

forma de aprendizado, que pode ser supervisionado ou não supervisionado (BRAGA et al., 2000).

Na forma de aprendizado supervisionado, são sucessivamente apresentados à rede conjuntos de padrões de entrada e seus correspondentes padrões de saída. A aprendizagem supervisionada também é conhecida como aprendizagem com um professor. Em termos conceituais, o professor é considerado uma fonte que contém o conhecimento sobre o ambiente, com este conhecimento sendo representado por um conjunto de entradas e saídas. Entretanto o ambiente é desconhecido pela rede neural de interesse. A figura 6 ilustra um diagrama em blocos para esta forma de aprendizagem (HAYKIN, 2001).

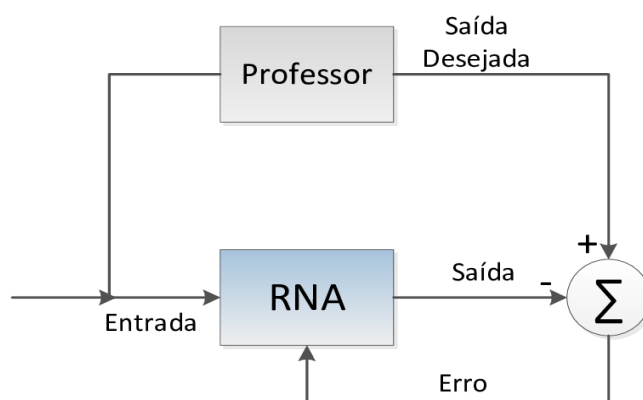


Figura 6. Diagrama em blocos da aprendizagem com um professor

**Fonte:** Haykin (2001).

Neste tipo de aprendizado, supondo que a rede neural e professor sejam expostos a um novo vetor de treinamento retirado do ambiente, em virtude ao seu conhecimento prévio, o professor é capaz de fornecer a rede neural uma resposta desejada para aquele vetor de treinamento. A resposta desejada irá representar a ação ótima a ser realizada pela rede neural. Os parâmetros da rede são ajustados sob a influência combinada do vetor de treinamento e do sinal de erro. O sinal de erro é definido como a diferença entre a resposta desejada e a resposta real da rede. Este ajuste é realizado passo a passo, iterativamente, com o objetivo de fazer a rede neural emular o professor (HAYKIN, 2001).

Segundo Haykin (2001), na aprendizagem não supervisionada, ou aprendizagem sem um professor, não há um professor para supervisionar o processo de aprendizagem. Isto significa que não há exemplos rotulados da função a ser aprendida pela rede neural. Em vez da

utilização de um professor, como ocorre no aprendizado supervisionado, são dadas condições para realizar uma medida independente da tarefa da qualidade da representação que a rede deve aprender, e os parâmetros livre da rede são otimizados em relação a esta medida.

Entre as técnicas de análise de dados que utilizam o conceito de aprendizado não supervisionado, destacam-se as técnicas de agrupamento de dados, isto é, métodos destinados a análise de dados que não possuem informações prévia sobre a classe a que pertencem. Assim, tais métodos têm como objetivo agrupar dados de acordo com algumas medidas de similares, sendo que os grupos encontrados devem refletir a estrutura subjacente dos dados (ASSAO, 2008). De acordo com Sanches (2003), a grande aplicação do aprendizado não supervisionado é no reconhecimento de padrões, agindo em dados a partir de alguma caracterização de regularidade, sendo esses padrões denominados *Clusters*. Os exemplos contidos em cada *Cluster* são mais similares, segundo alguma medida de similaridade, do que aqueles que estão contidos em *Clusters* diferentes.

### 2.3.1 Perceptron Multicamadas

Atualmente, a arquitetura de RNA que vem sendo mais utilizada em problemas de reconhecimento e classificações é a arquitetura de Perceptron multicamadas (Multilayer Perceptron - MLP) (AMAKDOUF et al., 2018). A MLP, de acordo com Gardner e Dorling (1998), consiste em um sistema simples interconectados de neurônios, na qual representam um mapeamento não linear entre um dado vetor de entrada e um vetor de saída. Cada nó ou neurônio de uma MLP estão conectados por pesos e sinais de saída que representam uma função de soma para o neurônio para possibilitar a ativação do mesmo (GARDNER E DORLING, 1998). Arquitetura de uma MLP está dividida entre o vetor de entrada, uma ou várias camadas ocultas de neurônios artificiais, e uma camada com vetor de saída, como representado na Figura 7.

O processo de treinamento ocorre de maneira iterativa, nessa etapa os pesos de entradas e os valores das funções de ativação dos neurônios artificiais são ajustados com o propósito de maximizar os resultados obtidos pela RNA. O algoritmo de treinamento mais conhecido para esse tipo de RNA é o algoritmo *Backpropagation*, o qual consiste em uma técnica em tentar localizar um mínimo absoluto global em uma superfície de erro que satisfaça o erro máximo

estabelecido para a RNA. Inicialmente, os pesos da RNA são definidos como valores baixos, então algoritmo *Backpropagation* irá calcular o gradiente local de uma superfície de erro alterando os pesos, dessa forma, indo em direção ao gradiente mais inclinado, conseqüentemente com uma menor taxa de erro Gardner e Dorling (1998).

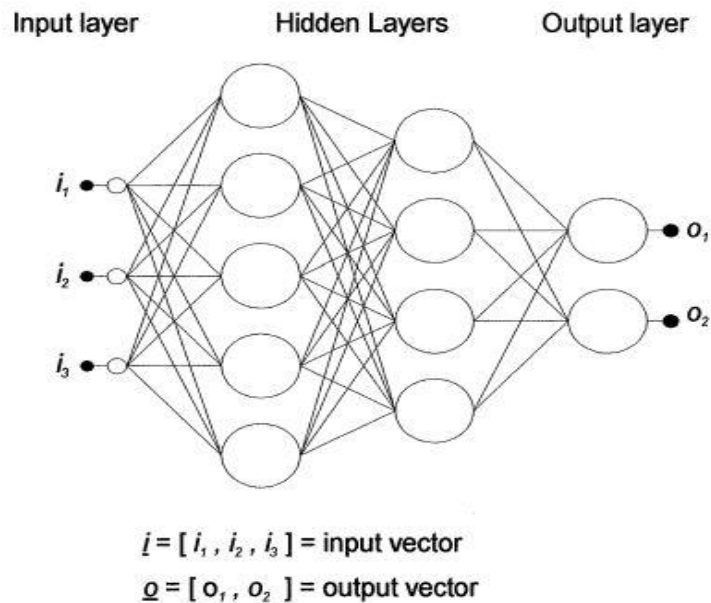


Figura 7. Arquitetura de uma MLP

**Fonte:** Gardner e Dorling (1998).

### 2.3.2 Redes Neurais Convolucionais

De acordo com Seo e Shin (2019) as CNN são uma arquitetura de rede neural utilizadas em classificações de imagens utilizadas principalmente em técnicas de aprendizado profundo (Deep Learning) e visão computacional. Esta arquitetura de RNA foi inspirada e baseada principalmente na estrutura do sistema visual. O que diferencia uma CNN de outras arquiteturas de redes neurais são as três camadas internas que ela possui, camada de convolução, camada de Pooling e por último uma camada totalmente conectada onde todos os neurônios da rede se conectam (GU et al., 2018). A Figura 8 ilustra a arquitetura de uma CNN.

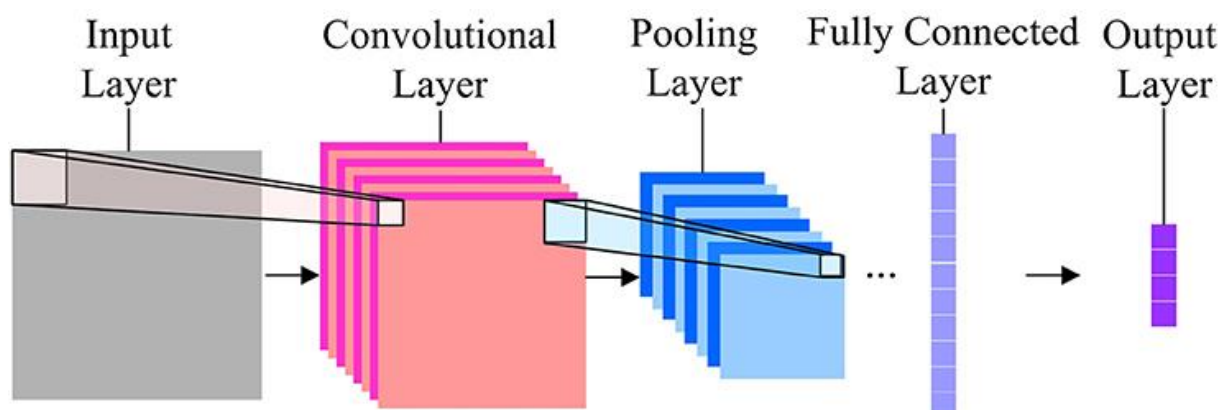


Figura 8. Arquitetura de uma CNN

**Fonte:** Peng et al. (2017).

As primeiras etapas de uma arquitetura de uma CNN são compostas por dois tipos de camadas diferentes, que são as camadas onde são aplicados os operadores de convolução e de *Pooling*. Na etapa onde será aplicado o operador de convolução, a imagem de entrada passará por um filtro, onde o somatório da multiplicação de cada *bit* que representa a imagem juntamente com seu vizinho mais próximo por uma segunda matriz que representará a matriz de convolução. O resultado dessa multiplicação irá gerar uma nova matriz de *bits* que representará uma outra imagem que servirá de entrada para uma camada posterior de uma CNN. Nesta etapa mais de um filtro de convolução pode ser aplicado, resultando em mais de uma matriz com mapas de características da imagem analisada (LECUN; BENGIO; HINTON, 2015).

A camada de *Pooling* em uma CNN tem como entrada os mapas de características gerados pela camada de convolução. Nesta etapa os *bits* que representam as características de uma imagem em sua forma matricial são subdivididos em grupos, sendo que cada grupo é analisado e somente o *bit* de maior valor ou a média de todos os *bits* são coletadas para gerar uma nova matriz que representará as características de uma imagem. O principal objetivo desta etapa é selecionar as regiões da imagem que irão ter um maior valor significativo para a classificação da imagem, descartando os *Pixels* da imagem que terão valor irrelevante para a sua classificação (GU et al., 2018).

A última etapa de uma CNN é composta por camadas que representam uma rede neural densa totalmente conectada, como por exemplo uma RNA que possui a arquitetura de uma MLP. Esta camada de uma CNN possui como entrada os dados gerados pelas camadas

anteriores, camadas de convolução e Pooling. O objetivo desta etapa é gerar o resultado final na classificação que está sendo realizada. O processo de classificação em uma CNN é iterativo, podendo interagir mais de uma vez entre as primeiras camadas de convolução e *Pooling* até a etapa final onde é aplicada a RNA completa. Este processo iterativo é necessário para descobrir qual filtro de convolução obteve o melhor resultado quando foi utilizado mais de um filtro (HIJAZI; KUMAR; ROWEN, 2015).

## 2.4 TENSORFLOW

O TensorFlow é uma biblioteca de software de código aberto para computação numérica que usa gráficos de fluxos de dados. A biblioteca foi desenvolvida por pesquisadores e engenheiros da Google Brain Team, no departamento de pesquisas de inteligência de máquina do Google com a finalidade de realizar pesquisas sobre redes neurais profundas e aprendizado de máquina. O TensorFlow possui uma arquitetura flexível, o que permite que seja implantado em aplicações computacionais que utilizam uma ou mais CPUs, em servidores ou em dispositivos móveis utilizando uma única API (Application Programming Interface).

Um das principais características do TensorFlow é permitir que desenvolvedores experimentem novas otimizações e algoritmos de treinamentos com base em seus interesses. Atualmente, as aplicações que utilizam TensorFlow estão voltadas para executar redes neurais profundas com objetivo de classificar manuscritos de dígitos, reconhecimento de imagens, incorporação de palavras, redes neurais recorrentes, modelos sequência a sequência para tradução automática, processamento de linguagem natural e simulações baseadas em equações diferenciais parciais (TENSORFLOW, 2018).

## 2.5 ESTADO DA ARTE

O setor agrícola brasileiro vem sendo um dos responsáveis por impulsionar a economia brasileira positivamente. O Brasil é o terceiro maior exportador de grãos do mundo, só perdendo para os Estados Unidos e a União Europeia (USDA, 2018). O fato deste setor estar em alta incentiva a criação de novas tecnologias, trabalhos e pesquisas que possam ajudar a alavancar ainda mais esta área. Um dos trabalhos computacionais que vem sendo aplicado nesta área é a

de detecção de doenças e pragas em lavouras utilizando técnicas de processamento de imagens e RNA.

A pesquisa de Ferreira (2017) propôs o desenvolvimento de uma aplicação utilizando visão computacional e redes neurais profundas que era capaz de detectar ervas daninhas (Plantas indesejadas que crescem em culturas agrícolas) quando elas estivessem presentes em lavouras de soja. Para o desenvolvimento deste trabalho foi utilizado uma base de dados com cerca de 15 mil imagens que foram coletadas através de fotografias aéreas capturadas por um Veículo Aéreo Não Tripulado (VANT) em lavouras de sojas. Este trabalho utilizou o modelo de RNA CaffeNet, uma variação do modelo AlexNet, um modelo CNN conhecida por apresentar excelentes resultados e desempenho em problemas de classificação de imagens. O trabalho de Ferreira obteve uma precisão superior a 98%, uma precisão consideravelmente alta. Segundo Ferreira (2017), os excelentes resultados obtidos em seu projeto tiveram como grande influência a base de imagens utilizada no treinamento de seu modelo e a utilização de um modelo eficiente de CNN, que se comportou muito bem para o problema em questão.

Melo (2015) realizou o desenvolvimento de uma aplicação para a plataforma Android que era capaz de realizar o reconhecimento automático dos índices de severidade da ferrugem asiática da soja. Este trabalho teve grande utilidade para auxiliar no diagnóstico precoce da doença e realizar uma tomada de decisão no controle da doença. Para o desenvolvimento deste trabalho foi utilizado técnicas de processamento digital de imagens, RNAs e uma base de imagens com aproximadamente 2 mil folhas de soja contaminadas com a ferrugem asiática. A base de imagens foi dividida em sete classes diferentes representando os níveis de severidade da ferrugem asiática. Ao final do projeto o melhor resultado alcançado foi uma precisão de 78,86%, utilizando o algoritmo Resilient-propagation. A arquitetura de RNA utilizada neste projeto foi um MLP, possuindo uma única camada oculta com 15 neurônios artificiais.

As aplicações para dispositivos móveis estão cada vez mais incluídas no meio agrícola. A plataforma Android por ser a mais utilizada no mundo é a mais comum de ser utilizada nesse meio de aplicações (MELO; MATHIAS; DIAS, 2015). Melo, Mathias e Dias (2015) propuseram uma aplicação móvel para a plataforma Android que era capaz de realizar a classificação automática do oídio da soja. Este trabalho foi desenvolvido utilizando RNAs para realizar a classificação das imagens e a biblioteca OpenCV para realizar o pré-processamento das imagens utilizadas no projeto. Após o estudo os autores confirmaram a viabilidade da

utilização de tais técnicas para solucionar problemas desta área, abrindo a possibilidade de novas pesquisas nesse meio.

Ainda sobre a detecção da ferrugem asiática na folha de soja, Cassol e Agosti (2015) realizaram a construção de um software para detecção da ferrugem asiática na folha de soja utilizando RNAs para realizar a classificação de imagens. Para a execução e treinamento do modelo de RNA foi utilizado o Framework ANNeF, utilizando o algoritmo Backpropagation para realizar o treinamento do modelo. Após o término deste trabalho Cassol e Agosti (2015) concluíram que através da extração de dados de uma imagem e gerando dados de entrada para uma RNA é totalmente possível realizar a detecção da ferrugem asiática na folha da soja utilizando uma RNA de topologia MLP.

A biblioteca TensorFlow, por ser uma biblioteca recente (Lançada em 9 de novembro de 2015), ainda não foi aplicada em muitos trabalhos científicos voltados para a área agrícola no Brasil. A maioria dos trabalhos que utilizaram esta biblioteca são trabalhos estrangeiros, voltados para visão computacional, aprendizado de máquina e criação, treinamento e execução de modelos de RNA. Abadi et al. (2016) aborda a principais funcionalidade e benefícios da utilização do TensorFlow em problema de aprendizado de máquina e execução de RNAs profundas. Segundo Abadi et al. (2016) um dos principais benefícios da utilização de TensorFlow é o alto desempenho, se tratando de precisão e velocidade em problemas com foco em treinamento e inferência. A fácil integração do TensorFlow com RNAs profundas o torna uma forte ferramenta para ser utilizada em problemas de visão computacional, como em problemas que necessitam de reconhecimento de objetos em imagens.



### 3 METODOLOGIA

Os métodos e atividades necessárias para o desenvolvimento deste trabalho foram divididos em quatro etapas principais (Figura 9), que são as etapas de coleta de imagens, treinamento da RNA, desenvolvimento do aplicativo Android e realizar a aplicação de métodos destes e métricas de avaliação para uma avaliação mais precisa do aplicativo.

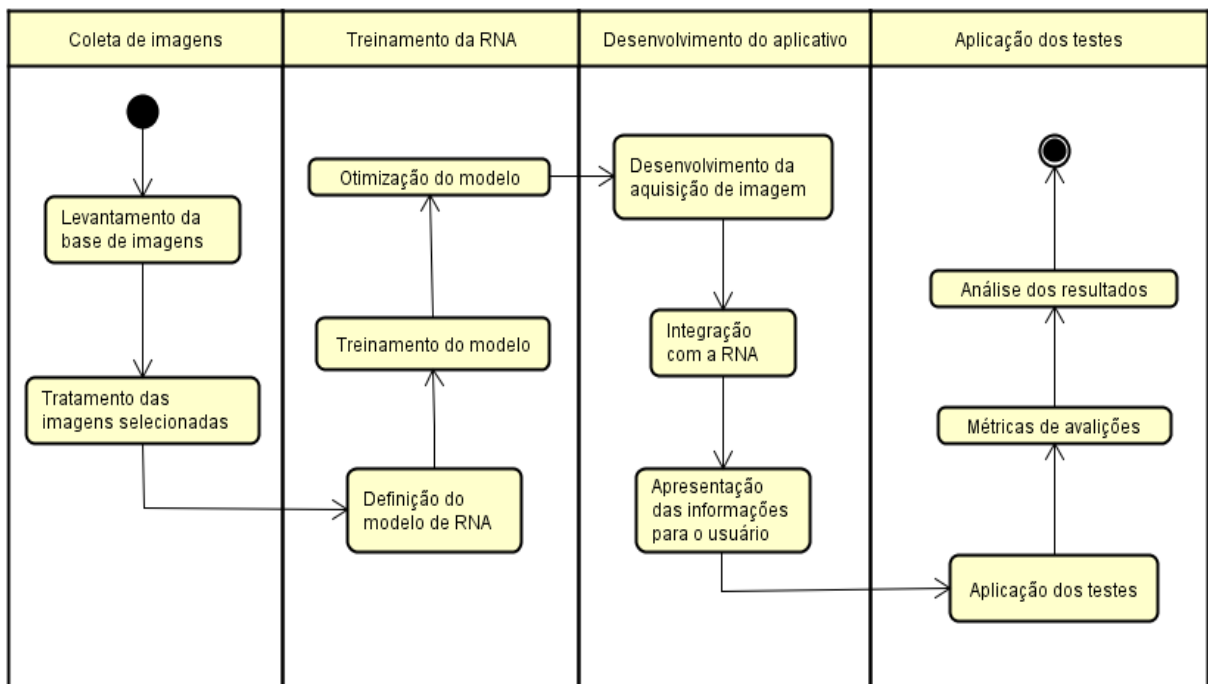


Figura 9. Visão geral das atividades desenvolvidas.

#### 3.1 BASE DE IMAGENS

Para a elaboração deste trabalho foi necessário o levantamento de dois grupos diferentes de imagens de folhas de soja. O primeiro grupo (Grupo 1) foi o de imagens de folhas de soja que possuíam lesões causadas pela ferrugem asiática (Figura 10). As folhas pertencentes a este grupo de imagens representam lesões causadas pela ferrugem asiática desde os índices mais baixos de severidade da doença até os índices mais altos. Ao total foram reunidas 770 imagens de folhas contaminadas com a ferrugem asiática de três bases diferentes, que são as bases de

imagens abertas despolitizadas pelo Invasive (2019), Ipmimages (2019) que ao todo totalizaram 570 imagens, e imagens coletadas em campo que totalizaram 200 imagens.

O segundo grupo de imagens (Grupo 2) que foi utilizado neste projeto, representado na Figura 11 contém imagens variadas de folhas de soja, tanto folhas saudias quando folhas contaminadas com outras doenças da cultura da soja, exceto folhas contaminadas com a ferrugem asiática. Este grupo contém 2000 imagens que foram retiradas da Plant Village (2019), uma plataforma online que se dedica aos estudos em doenças de plantas. A Plant Village possui mais de 8000 imagens de plantas de soja, incluindo imagens de folhas saudias, com doença e também imagens da planta. Todas as imagens presentes no banco de dados da Plant Village foram obtidas em estações de pesquisas associadas ao Land Grant Universities, USA. Projetos similares a esse, como o de Pires (2017), utilizam a Plant Village com base de imagens.

Após a etapa de extração de partes das texturas das folhas do Grupo 1 e Grupo 2 citada na Seção 3.2, o Grupo 1 de imagens continha 981 imagens de folhas de soja com ferrugem asiática, sendo imagens de 770 folhas distintas. O Grupo 2 de imagens passou a ter 2500 imagens, sendo 2000 de folhas distintas contendo folhas saudias e contaminadas com outras doenças sem ser a ferrugem asiática da soja.

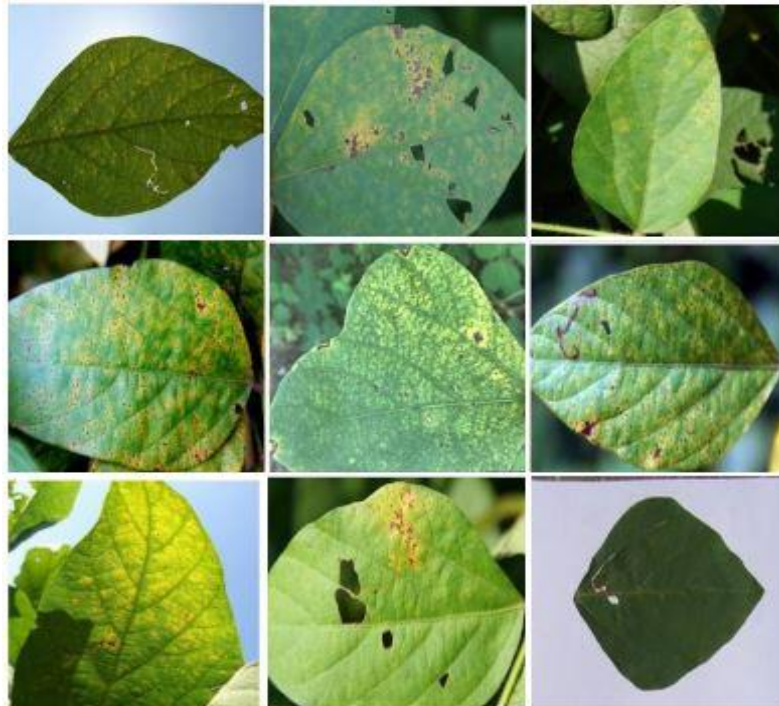


Figura 10. Grupo 1 da base de imagens.

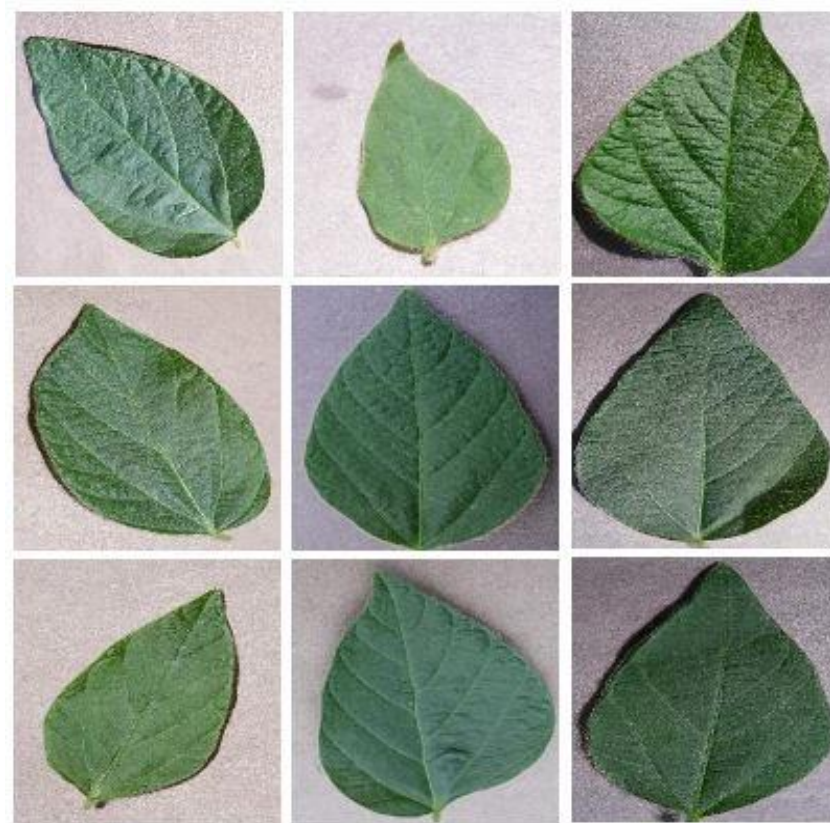


Figura 11. Grupo 2 da base de imagens.

### 3.2 PRÉ-PROCESSAMENTO DIGITAL DAS IMAGENS

Os dois grupos de imagens foram submetidos a quatro etapas iniciais de pré-processamento antes de serem submetidas ao treinamento da RNA deste trabalho. A primeira etapa consistiu em redimensionar todas as imagens que estavam disponíveis na base de dados para a dimensão de 224 *pixels* de altura por 224 *pixels* de largura. Todas as imagens do grupo dois deste trabalho, que foram fornecidas pela Plant Village, já estavam nesta dimensão, por isso não sofreram alteração no tamanho. A escolha de realizar o redimensionamento das imagens para esta medida específica foi pelo fato de que ela não seria uma resolução muito alta que causaria uma entrada de dados muito grande para a RNA utilizada, tornando a classificação das imagens lenta. Como o aplicativo irá funcionar em dispositivos móveis que nem sempre tem uma capacidade muito alta de processamento, esta etapa foi necessária para deixar a classificação das imagens mais rápida. Para o redimensionamento destas imagens foi utilizado o resultado produzido pelo trabalho de Silva (2009), o qual consiste em um software para

redimensionamento de imagens que aplica técnicas para manter a proporção original dos objetos presentes na imagem após a alteração nas medidas da imagem. Desta forma, a proporção da folha de soja continuou o mais próximo do original, mesmo que houvesse um ajuste desproporcional entre a altura e a largura da imagem.

A Segunda etapa de pré-processamento das imagens, antes que fossem submetidas a etapa de treinamento da RNA, consistiu em segmentar todas as imagens da base de dados. Nesta etapa todos os *Pixels* das imagens que não faziam parte da folha de soja receberam os tons de vermelho, verde e azul e igual a zero. Desta forma somente o objeto de interesse, a folha de soja, permaneceu nas imagens. Para realizar a extração das folhas nas imagens foi utilizado o algoritmo implementado por (ANANTRASIRICHAJ; HANNUNA e CANAGARAJAH, 2017) o qual se baseia na técnica de segmentação de imagens Watershed. A técnica de Watershed inicialmente realiza a transformação da imagem a ser segmentada para uma escala de cinza. Em seguida é verificado onde as regiões de interesse na imagem se encontram, ou seja, as bordas dos objetos presentes na imagem. Após a identificação dos objetos da imagem, todos os *pixels* que não pertencem a um objeto são descartados, em seguida a imagem é transformada para a escala de cor original. Para que a extração da folha da imagem seja realizada, a área com maior quantidade de verde é selecionada (o mais provável que seja uma folha) inicialmente, em seguida outras regiões que se conectam a esta área também são selecionadas, neste caso, outros objetos que podem fazer parte da folha.

Com as imagens da base segmentadas e redimensionadas, elas foram submetidas a uma etapa de extração de texturas, que consistiu em extrair da folha de soja partes relevantes nas imagens que pudessem descrever suas principais características em relação a doença presente na folha caso a folha estivesse infectada com alguma doença, ou partes da textura da folha que descrevessem e apresentassem de forma clara as características de uma folha de soja saudável. Toda esta etapa foi realizada de forma manual, e os resultados obtidos após este processo são representados na Figura 12. Todas as texturas extraídas foram reaplicadas a etapa de redimensionamento para a dimensão de 224x224 *Pixels* e foram reintegradas ao seu respectivo grupo de imagens.

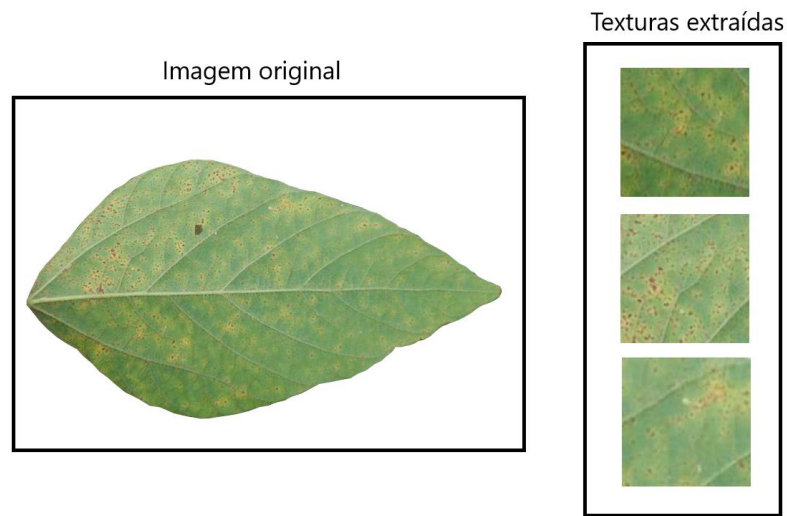


Figura 12. Extração de texturas das folhas de soja.

A última etapa de pré-processamento realizada na base de imagens consistiu em realizar a transformação das imagens disponíveis para escala de cinza, antes de que elas fossem submetidas ao treinamento da RNA. As imagens originais possuem seus tons de vermelho, verde e azul (Red Green Blue - RGB) em uma escala entre 0 e 255. Para que as imagens mudassem para a escala de cinza os três canais de cores de RGB foram unificados para um canal único de cor que varia entre 0 a 255. Unificação dos valores para escala de cinza aconteceu utilizando a seguinte fórmula:  $R * 0.3 + G * 0.59 + B * 0.11$ .

Conversão realizada para escala de cinza

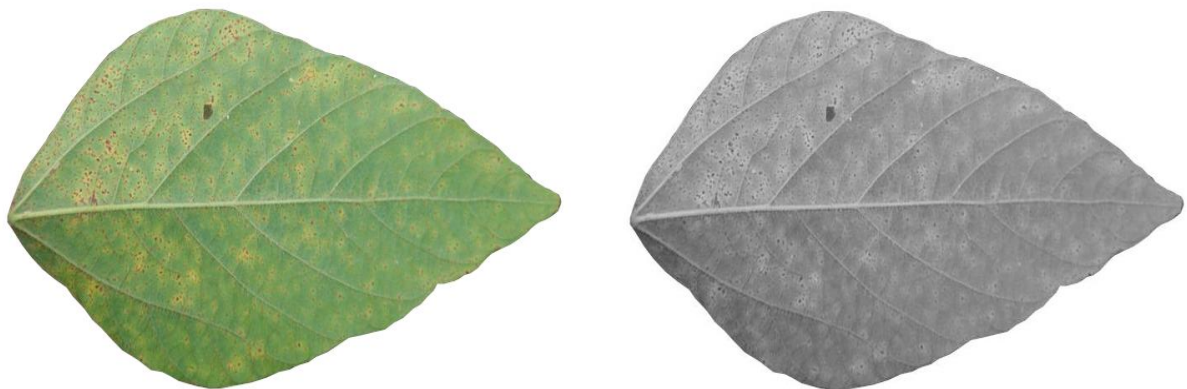


Figura 13. Conversão realizada para escala de cinza.

### 3.3 ARQUITETURA E TREINAMENTO DA RNA

Esta etapa do projeto consistiu em realizar a definição e treinamento do modelo de RNA que foi utilizada no aplicativo Android para realizar as classificações das imagens de folhas de soja. Para realizar o treinamento da RNA foram utilizados os resultados das imagens produzidas na etapa de pré-processamento de imagens. Foram avaliados dois modelos de RNA baseados na arquitetura de CNNs, o modelo Inception-V3 e o modelo Mobilenet 0.50. O modelo Mobilenet 0.50, projetado especificamente para funcionar em dispositivos moveis, apesar de ser um modelo mais leve do que o Inception-V3, apresenta resultados bem inferiores aos resultados produzidos pelo Inception-V3. Pensando na maximização dos resultados para este projeto o modelo escolhido para ser utilizado foi o Inception-V3.

O Inception-V3 é um modelo de RNA profunda utilizado em Deep Learning que se baseia na arquitetura de CNN. O Inception-V3 foi o sucessor do seu modelo anterior, o Inception-V2. O motivo da escolha desse modelo de RNA foram os ótimos resultados obtidos em problemas de classificação de imagens e o fato desse modelo ser extremamente rápido em problemas de classificação de imagens. A facilidade de realizar a integração deste modelo com o TensorFlow e a plataforma Android também influenciaram e escolha do Inception-V3. A Figura 13 ilustra a arquitetura do modelo Inception-V3.

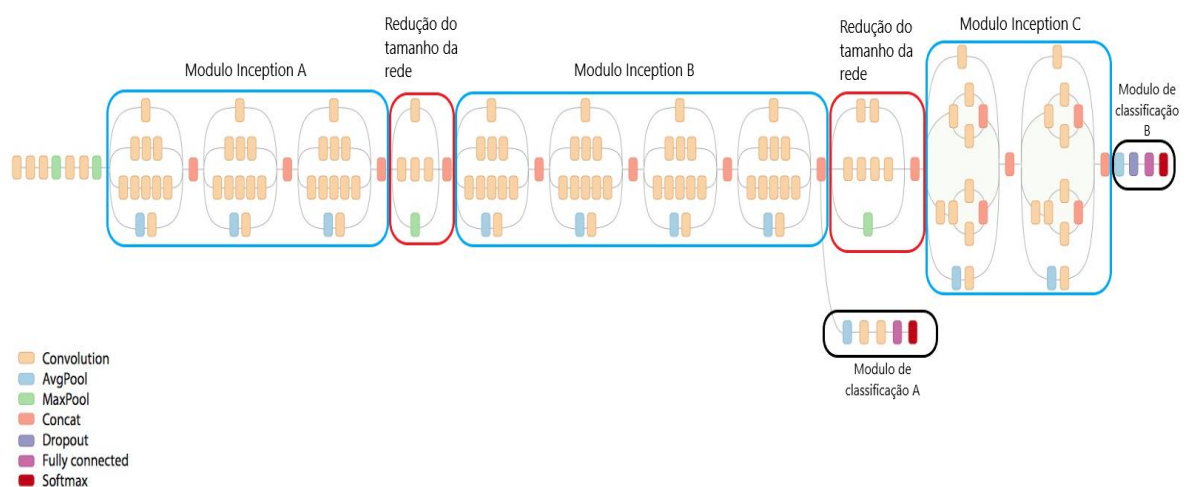


Figura 14. Arquitetura do Inception-V3.

**Fonte:** Adaptado de Islam et al. (2018).

O modelo utilizado possui 42 camadas internas, que estão divididas com as seguintes funcionalidades:

- Convolution: Camada onde é aplicada os filtros de convolução, podendo possuir dimensões de uma matriz 1x1, 3x3 e 5x5.
- AvgPool: Camada onde as matrizes de características de entrada são divididas em grupos, e em cada grupo é feita a média dos valores presentes nele gerando uma nova matriz resultante de características.
- MaxPool: Camada semelhante ao AvgPool, porem em vez de realizar a média dos valores pertencentes a um grupo é selecionado o maior valor de cada grupo.
- Concat: Esta camada irá reunir os resultados das saídas geradas pelas camadas Convolution, AvgPool e MaxPool e fornecer uma nova entrada para o próximo modulo de processamento do Inception.
- Dropout: Camada utilizada para minimizar o Overfitting na RNA.
- Fully Connected: Camada constituída de uma RNA densa, que recebera o resultado gerado pelas camadas anteriores a ela e irá realizar a classificação da imagem.
- Softmax: Camada que irá receber o vetor resultante da camada Fully Connected e irá calcular qual a probabilidade do item que está sendo classificado em cada uma das classes de saída. Neste trabalho, esta camada verificou a probabilidade de uma folha de soja estar contaminada com a ferrugem asiática da soja ou não, dando como saída a probabilidade para cada uma dessas classes.

A arquitetura do Inception-V3 (Figura 14), está dividida em três módulos principais: módulos Inception A, B e C. Estes módulos possuem camadas de Convolution e AvgPool, que são utilizadas com o objetivo de aplicar os filtros de convolução e gerar os mapas de características dos registros de entrada, descartando regiões de não interesse no problema que está sendo tratado e por fim tornando a entrada para camada onde é realizada a classificação no modelo menor, tornando o classificador mais rápido. Entre cada um desses módulos do Inception-V3 existe uma camada intermediária que tem como objetivo obter o resultado obtido em cada modulo, realizar redução e por fim, gerar uma nova entrada para o modulo posterior do Inception.

O modelo Inception-V3, possui dois módulos adicionais para realizar suas classificações. O módulo de classificação A (Figura 14) foi utilizado para realizar classificações durante o período de treinamento do modelo, onde é realizado uma abordagem de aprendizado supervisionado, gerando uma comparação entre a resposta dada pelo classificador e o seu real valor representado na base de dados. Enquanto o módulo de classificação B (Figura 14) foi utilizado somente durante os períodos de testes ou inferências, ou seja, após o treinamento ser realizado, este será o módulo utilizado para realizar as classificações das imagens.

Para a realização do treinamento do modelo Inception-V3 foram utilizadas as tecnologias e ferramentas listadas no Quadro 1.

Quadro 1 – Tecnologias e ferramentas utilizadas para o treinamento do Inception-V3.

<b>Ferramenta/Tecnologia</b>	<b>Descrição</b>
Python	Linguagem de programação interpretada, tipagem dinâmica e de alto nível utilizada para construção de Scripts para processamento e análise de dados.
Docker	Tecnologia de software utilizada para virtualização de componentes baseada em containers. Funciona adicionando uma camada de abstração entre o sistema operacional, Linux ou Windows e o container utilizado.
TensorFlow 1.1.0	Biblioteca utilizada para gerar o modelo do Inception-V3, extrair os atributos da imagem, realizar o treinamento do modelo e realizar classificações.
Numpy	Biblioteca para linguagem de programação Python utilização na computação numérica para realização de operações sobre estruturas de dados como matrizes e vetores.
Windows 10 Pro	Sistema operacional utilizado para criação do ambiente de desenvolvimento e execução dos testes
Asus K46CA	Notebook utilizado para o desenvolvimento para o desenvolvimento do projeto, possuindo as seguintes configurações: Processador Intel Core i5 com quatro núcleos de 1.70 Gigahertz (GHz), Memória Ram de DDR3 de 8 Gigabytes (GBs).

Além dos valores de Learning rate e épocas de treinamentos, que sofreram variação de valores em cada etapa de treinamento (Tabela 1), outras variáveis tiveram valores estabelecidos no modelo sem sofrerem variações de valores entre cada etapa dos treinamentos realizados (Quadro 2). O modelo do Inception-V3 utilizado foi configurado para receber uma entrada de dados correspondentes aos 224x224x3 *Pixels*, com bits que representam os tons de RGB de cada *Pixels* de uma das imagens presentes na base de dados e por fim gerar uma saída com uma



resposta indicando uma probabilidade que varia de 0 a 1 indicando quais as chances da imagem classificada fazer parte de uma das duas classes de saídas possíveis, a classe que representa o Grupo 1 de imagens (Folhas que possuem ferrugem) ou Grupo 2 de imagens (Demais folhas de soja).

Quadro 2 – Parâmetros estabelecidos ao Inception-V3 durante os treinamentos realizados.

<b>Parâmetro</b>	<b>Descrição</b>	<b>Valor utilizado</b>
Percentual de teste	Porcentagem de imagens presentes na utilizadas para realização dos testes de inferência.	30
Percentual de validação	Porcentagem de imagens presentes na utilizadas para realização da validação do modelo.	30
Frequência de avaliação dos resultados	Define com que frequência e intervalo de épocas de treinamento o modelo seria reavaliado.	10
Batch Size	Define quantas imagens serão utilizadas em cada etapa de treinamento.	50
Test Batch Size	Define o número de imagens submetidas para testar o Batch Size	100
Validation Batch Size	Define quantas imagens foram utilizadas para validação do Batch Size	100

A Tabela 1 ilustra todas as etapas de treinamento que foram realizadas com o modelo Inception-V3, ao total foram realizados seis treinamentos. Nos treinamentos foram aplicadas taxas de variação de Learning Rate e números de épocas. As imagens utilizadas nesta etapa foram geradas pela etapa de pré-processamento, aplicadas na etapa anterior do projeto, que continha além das imagens originais segmentadas e em escala de cinza, as imagens geradas pela extração de texturas realizada também na etapa de pré-processamento.

Os treinamentos realizados apresentaram os resultados demonstrados na Tabela 2 após o processo de validação do modelo treinado, onde toda base de imagens presentes no treinamento foi submetida ao novo modelo para gerar o cálculo final da entropia cruzada e acurácia do modelo. Os resultados presentes na Tabela 2 foram apresentados a partir dos resultados obtidos nas classificações realizadas no processo de validação pelo Modulo de classificação B do Inception-V3 (Figura 14).

Tabela 1 - Treinamentos realizados.

<b>Etapa</b>	<b>Imagens G1</b>	<b>Imagens G2</b>	<b>Learning rate</b>	<b>Épocas</b>
1	781	827	0,1	5000
2	781	827	0,01	5000
3	781	827	0,001	5000
4	781	827	0,1	10000
5	781	827	0,01	10000
6	781	827	0,001	10000

Tabela 2 – Resultados obtidos nos treinamentos realizados.

<b>Etapa</b>	<b>Tempo de treinamento</b>	<b>Acurácia</b>	<b>Entropia cruzada</b>
1	01:21:14s	86.0%	2.7%
2	01:19:04s	85.5%	0.47%
3	01:28:00s	84.0%	0.44%
4	02:12:31s	86.7%	2.4%
5	02:15:54s	87.9%	0.72%
6	02:10:11s	87.3%	0.44%

Apesar da pouca variação entre os resultados obtidos na validação em cada etapa de treinamento, a etapa 5 entre os treinamentos realizados foi a que obteve melhores resultados e foi o modelo escolhido para ser utilizado no aplicativo Android e dar continuidade aos testes.

Ao final de cada treinamento realizado os estados das variáveis que representam os valores de cada peso sináptico, valores dos filtros de convolução e outros valores que representam a estrutura do modelo treinado foram exportados para um arquivo binário no formato Protocol Buffers (PB). Além do arquivo PB gerado, um segundo arquivo de texto foi gerado contendo o nome de cada classe presente no problema, neste caso, a classe que nomeava as folhas com ferrugem asiática e a classe que nomeava as folhas que não possuíam ferrugem. Estes dois arquivos gerados ao final dos treinamentos foram necessários para realizar a integração do modelo treinado com a aplicativo Android construído, pois a versão Mobile do TensorFlow exige um arquivo PB com informação do classificador e um arquivo de texto com os nomes das classes as serem classificadas para que se possa realizar classificações no Android. Os gráficos ilustrados nas imagens da Figura 15, Figura 16, Figura 17, Figura 18, Figura 19 e

Figura 20 ilustram a evolução da acurácia e entropia cruzada durante cada geração da RNA nos treinamentos realizados. O modulo de classificação A (Figura 14) foi responsável por realizar os testes durante as evoluções ocorridas em cada geração do modelo.

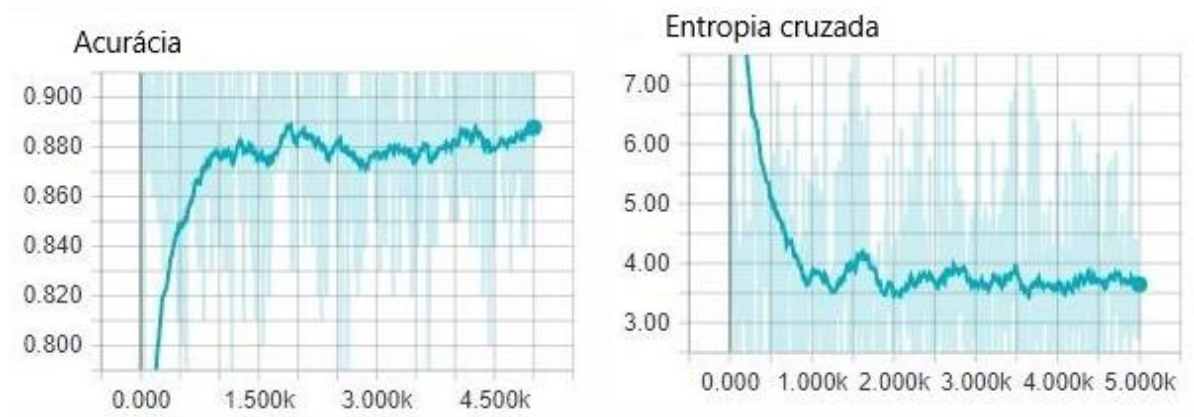


Figura 15. Evolução da acurácia e entropia cruzada durante a etapa 1 de treinamento.

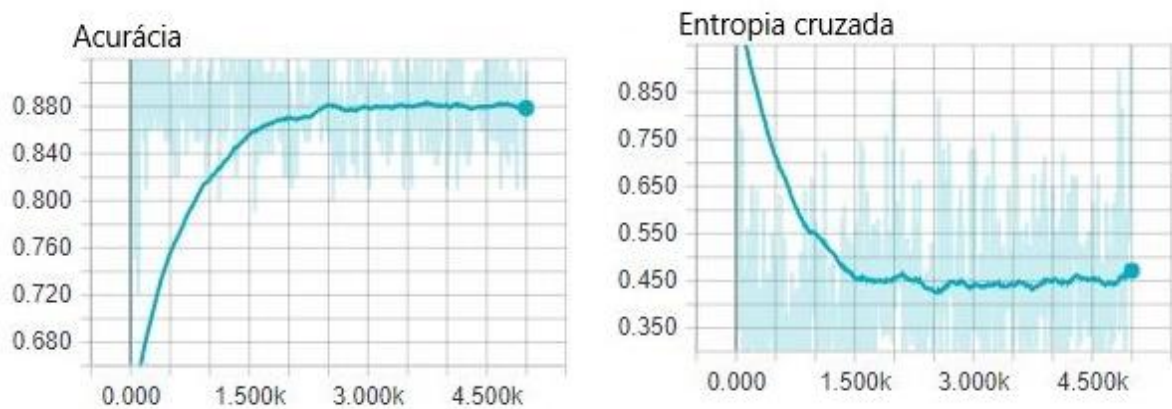


Figura 16. Evolução da acurácia e entropia cruzada durante a etapa 2 de treinamento.

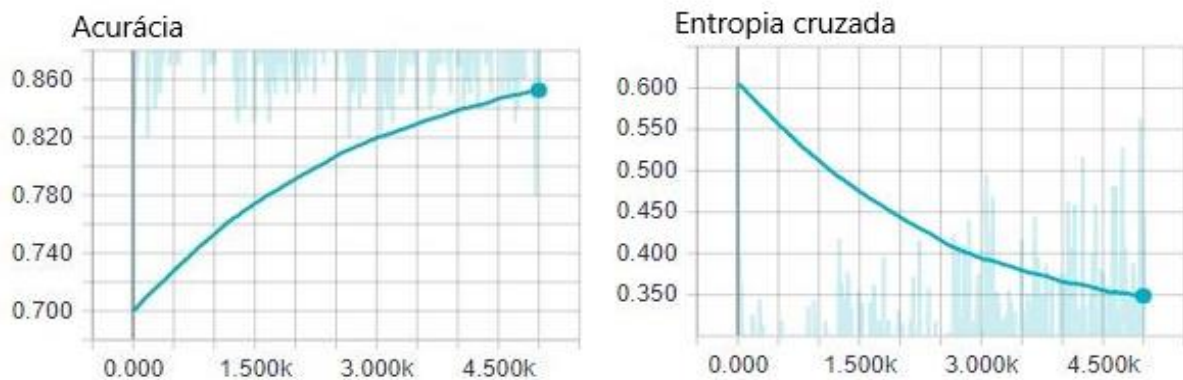


Figura 17. Evolução da acurácia e entropia cruzada durante a etapa 3 de treinamento.

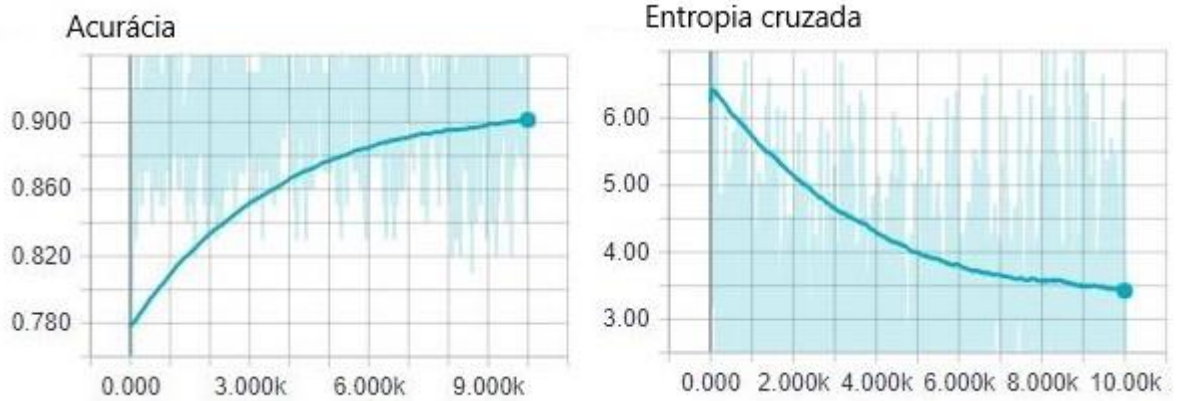


Figura 18. Evolução da acurácia e entropia cruzada durante a etapa 4 de treinamento.

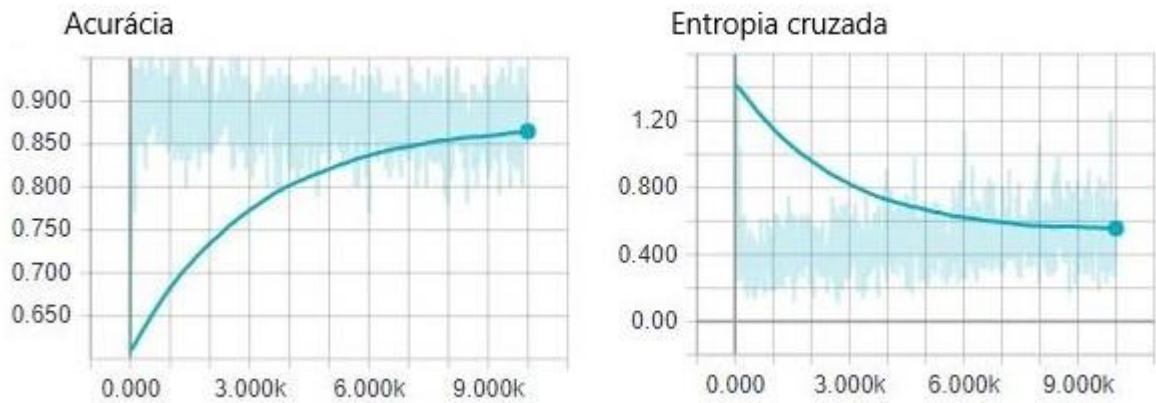


Figura 19. Evolução da acurácia e entropia cruzada durante a etapa 5 de treinamento.

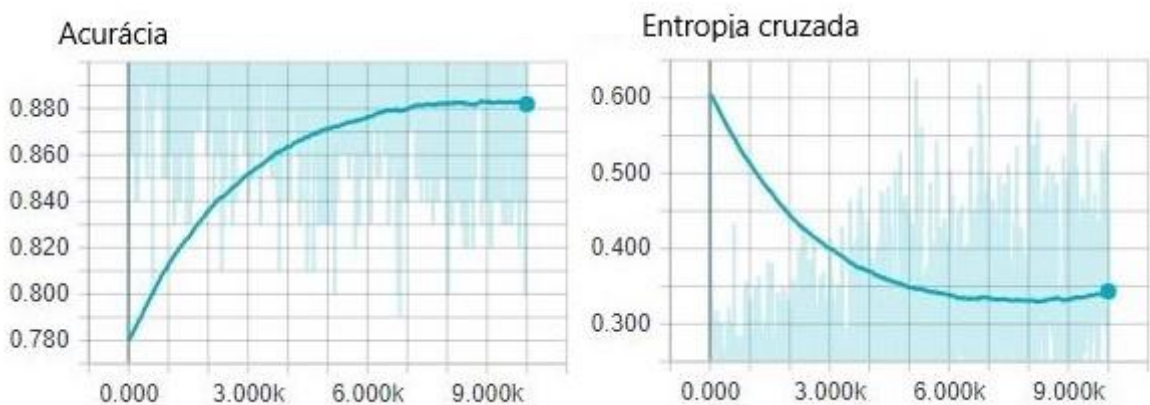


Figura 20. Evolução da acurácia e entropia cruzada durante a etapa 6 de treinamento.

### 3.4 DESENVOLVIMENTO DO APLICATIVO

Após o término dos treinamentos realizados sobre o modelo Inception-V3, deu-se início ao desenvolvimento do aplicativo Android que seria utilizado para realizar a detecção da ferrugem asiática na folha da soja em tempo real utilizando a câmera do dispositivo. O aplicativo final foi integrado com o modelo que obteve os melhores resultados durante os treinamentos realizados. O Quadro 3 aponta as ferramentas e tecnologias que foram utilizadas para o desenvolvimento do aplicativo.

Quadro 3 - Ferramentas e tecnologias utilizadas no desenvolvimento do aplicativo.

<b>Ferramenta</b>	<b>Descrição</b>
Android Studio 3.3.1	Ambiente de desenvolvimento integrado utilizado para o desenvolvimento de aplicações para a plataforma Android.
Android SDK 26.0.2	Conjunto de ferramentas de desenvolvimento de software que fornece funcionalidades para o desenvolvimento de aplicações para a plataforma Android.
Java	Linguagem de programação híbrida e orientada a objetos que pode ser utilizada para o desenvolvimento de softwares Mobile, Web e Desktop.
JDK 1.8.0	JDK (Java Development Kit) é um conjunto de utilitários que permite o desenvolvimento de software utilizando a linguagem de programação Java.
TensorFlow-Android 1.8.0	Biblioteca do TensorFlow para utilizar e realizar classificações em Java utilizando modelos pré-treinados pelo TensorFlow.
Camera2 API	Biblioteca mais atual fornecida pelo Android para manipular a câmera de dispositivos Android. Fornece funcionalidades para capturar ou transmitir imagens de um dispositivo Android com câmera.
Gradle 3.4.1	Sistema de automação e construção de código utilizado para realizar o gerenciamento de dependências e pacotes de aplicações Android.

O aplicativo foi desenvolvido e testado baseado na versão 7.1 (API nível 25) do sistema operacional Android. O TensorFlow-Android exige que o Android esteja no mínimo na versão 5.0 (API nível 21), desta forma, este aplicativo só está disponível para versões iguais ou superiores e essa.

Durante o processo de desenvolvimento do aplicativo todos os testes das funcionalidades implementadas foram realizados utilizando um Smartphone da marca Motorola

e modelo Moto g4 play. As especificações do dispositivo utilizado neste projeto são representadas no Quadro 4.

Quadro 4 – Especificações do dispositivo em que a aplicação foi desenvolvida.

Componente	Especificação
Processador	O dispositivo possui um processador com quatro núcleos de processamento com velocidade de 1.2 GHz cada.
Memória	O dispositivo possui 2 GB de memória Ram disponível.
Câmera	O dispositivo possui uma câmera de 8 Megapixels (MP).
Versão do Android	O dispositivo possui a versão 7.1 do Android (API Level 25).
Tela do dispositivo	O dispositivo possui uma tela de 5 polegadas com resolução de 1280x720 <i>pixels</i> .

Para o desenvolvimento do aplicativo Android foi utilizado o conceito de Orientação a Objetos (OO), aplicando seus pilares, como polimorfismos, herança, encapsulamento e abstração dos dados. O aplicativo implementado segue o seguinte modelo de classes representado no diagrama da Figura 21.

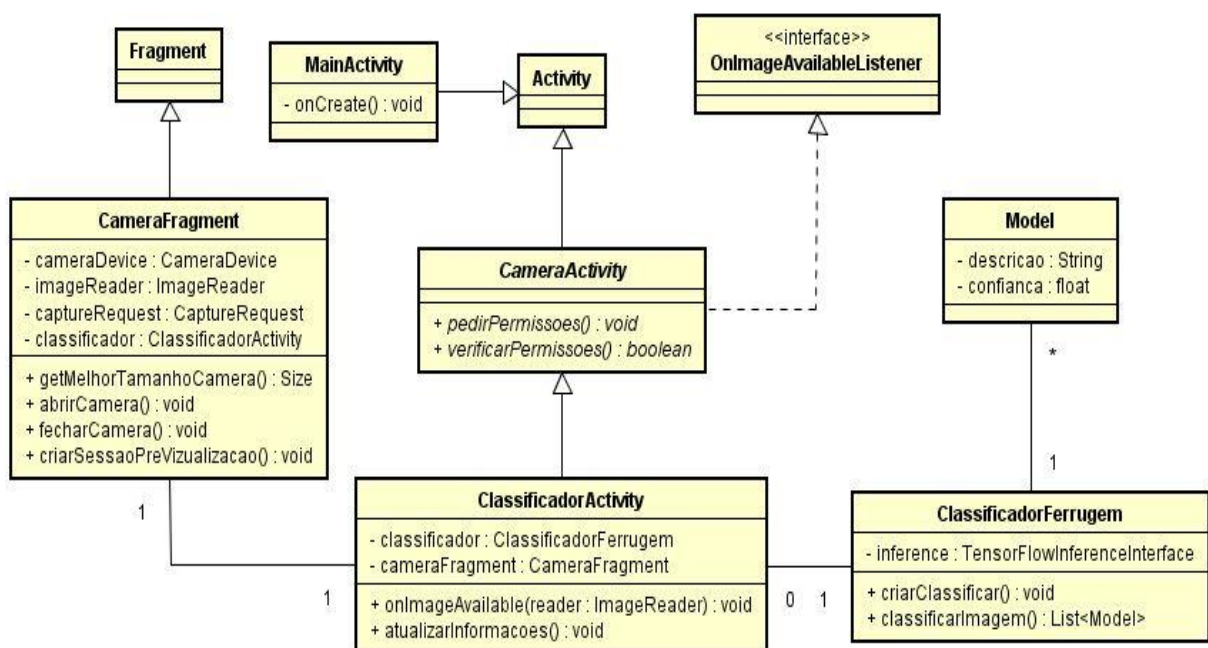


Figura 21. Diagrama de classes do aplicativo Android.

No modelo de classes representado na Figura 21, todas as classes que herdam Activity (classe nativa do Android) são utilizadas para gerenciar a interface de usuário no aplicativo. São nessas classes que são coletadas as informações geradas pela câmera do dispositivo, além de controlar as informações que serão exibidas para o usuário e delegar as imagens coletadas para serem classificadas. A classe CameraFragment, que herda Fragment (classe nativa do Android), foi construída para representar um fragmento de uma interface que localizará a região em que o usuário irá visualizar as imagens que estão sendo coletadas pela câmera do dispositivo.

Na classe CameraFragment estão localizadas funções para manipular a câmera (abrir e fechar), definir o melhor tamanho para o Fragment de pré-visualização de imagens de acordo com o tamanho da tela do dispositivo e também definir qual câmera será utilizada para coleta de imagens (no caso de dispositivos com mais de uma câmera). A classe ClassificarFerrugem é responsável por realizar a classificação das imagens obtidas. Ela utilizara a classe Model, responsável por armazenar informações das classificações realizadas.

### 3.4.1 Interfaces e Funcionalidades do Aplicativo

As interfaces do aplicativo foram projetadas e desenvolvidas para que qualquer usuário pudesse utilizar e se familiarizar com o aplicativo facilmente, mesmo aqueles que não possuem muito contato com tecnologia e a utilização de *Smathphones* (Figura 23). O motivo disto é o objetivo de atingir o maior número de usuários possíveis, tendo em vista que o ambiente em que esperasse que o aplicativo seja utilizado futuramente, as lavouras de soja, na maioria das vezes sendo utilizado por produtores rurais, os quais grande parte são pessoas que não possuem muito contato com tecnologias de difícil usabilidade, e teriam uma difícil adaptação com interfaces e funcionalidades complicadas. Por este motivo o aplicativo possui poucas interfaces e texto, e nenhum tipo de configuração fornecida pelo usuário.

O aplicativo possui uma única funcionalidade específica a qual o usuário precisará interagir com a aplicação, que é a de realizar a identificação da ferrugem asiática na folha da soja quando ela estiver presente. Para esta funcionalidade o usuário irá precisar inicializar a captura de imagens e concentrar a câmera do dispositivo em uma folha de soja. Todas outras etapas e funcionalidades ocorrerão de maneira interna, sem a interferência do usuário. O aplicativo irá analisar as imagens coletadas e fornecer a resposta sobre o resultado da

classificação para o usuário em tempo real. O diagrama de atividades representado na Figura 22 ilustra o fluxo de atividade necessário até realizar uma classificação no aplicativo.

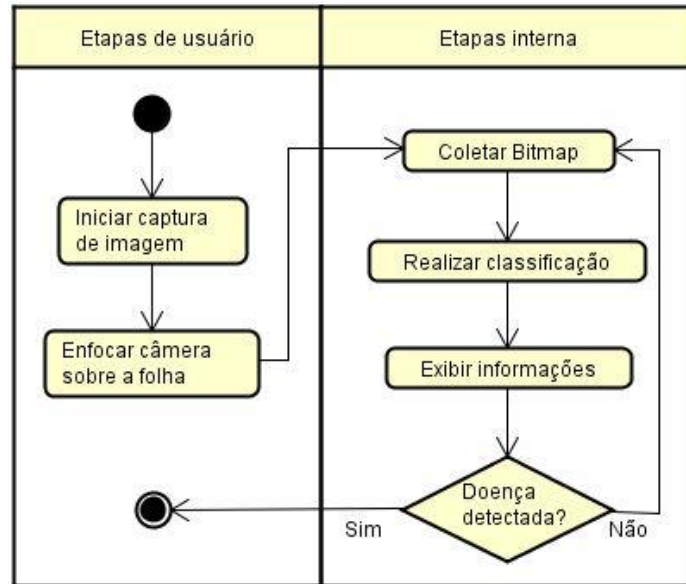


Figura 22. Fluxo de atividades desenvolvidas para classificação de uma imagem.

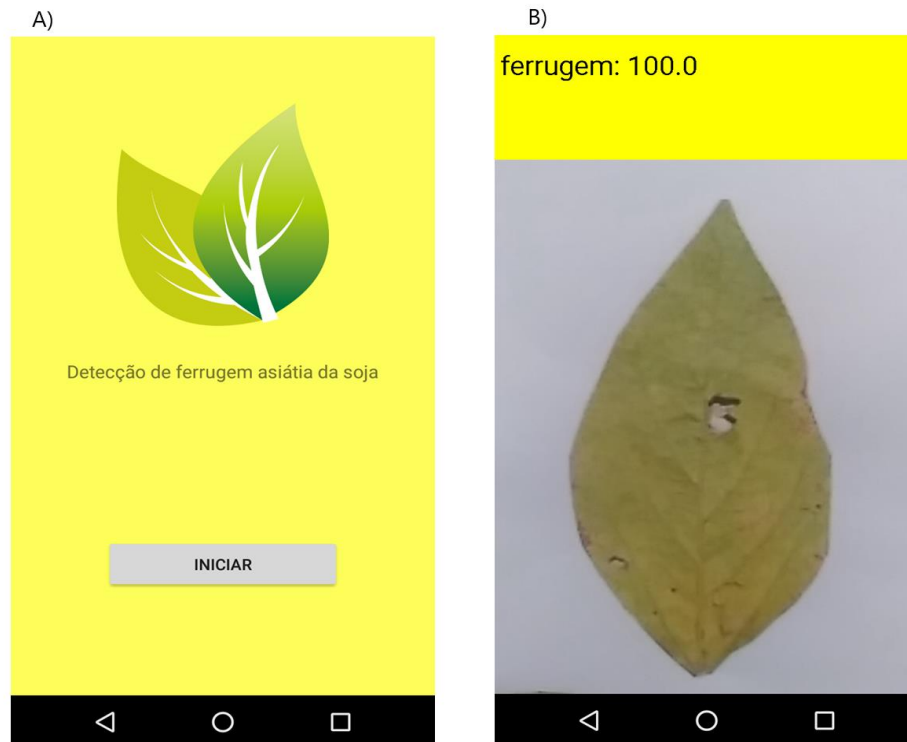


Figura 23. Interfaces de interação com usuário presentes no aplicativo.



### 3.4.2 Aquisição de Imagem

Esta etapa do desenvolvimento do aplicativo Android consistiu em desenvolver o processo responsável por realizar a captura de imagens com a câmera do dispositivo e transmitir essas imagens para o módulo responsável por realizar as classificações de imagens. Após iniciar o aplicativo o usuário irá ser redirecionado para interface representada na Figura 23a, após pressionar o botão de iniciar o aplicativo seria redirecionado para a interface representada na Figura 23b. A aquisição é iniciada de forma automática, a partir deste momento o usuário só precisa apontar a câmera do dispositivo para a folha de soja a ser analisada e todos os outros processos ocorreram internamente na aplicação.

Para o desenvolvimento desta etapa foi utilizado as classes e interfaces disponibilizadas pela *Camera2 API* (Quadro 3). As seguintes classes e interfaces fornecidas pela *Camera2 API* foram utilizadas neste aplicativo:

- *CameraDevice*: Classe responsável por representar e controlar uma única câmera que está conectada a um dispositivo Android.
- *CaptureRequest*: Pacote utilizado para configurações necessárias para realizar a captura de uma imagem da câmera do dispositivo. Este pacote também permite realizar algumas configurações de como sensores, lentes e *flash*.
- *CameraCaptureSession*: Classe responsável por iniciar uma sessão de captura de imagens e configurar o destino das imagens salvas.
- *CameraCharacteristics*: Métodos responsável por listar características de câmeras do dispositivo, como qual das câmeras disponíveis está sendo utilizada, qual a orientação atual da câmera (horizontal ou vertical) etc.
- *CameraManager*: Classe responsável por gerenciar a câmera que está sendo utilizada e executar funções como abrir e fechar a câmera e definir eventos para mudanças de estado na câmera.

A classe no projeto Android responsável por realizar a captura de imagens e implementar as classes e interfaces necessários fornecidas pela *Camera2 API* foi a *CameraFragment* (Figura 21). Nela foi realizada toda a configuração necessária para exibição das imagens capturadas para o usuário, além de definir o local de destino para as imagens capturadas e iniciar as sessões

de capturas de imagens. A classe *ClassificadorActivity* serviu como um ouvinte para a classe *CameraFragment*, nela foi reimplementado o método *onImageAvailable* da interface *OnImageAvailableListener*. Este método é invocado a cada nova imagem disponível que é capturado pela câmera, recebendo esta imagem por parâmetro. Este método recebe um objeto do tipo *ImageReader* por parâmetro, representando a imagem capturada, e após realizada todo o tratamento necessário na imagem para classificá-la.

### 3.4.3 Classificação das Imagens

Esta foi a etapa do projeto responsável por criar o módulo em que seria realizado a classificação das imagens coletadas no aplicativo Android. Esta etapa é executada paralelamente com a aquisição de imagens realizada no aplicativo. Ela funciona em segundo plano, e assim que uma nova imagem está disponível ela é chamada, e sem que a coleta de imagens pare, todo processo necessário para realizar a classificação da imagem coletada é realizado.

A classe responsável por realizar a classificação das imagens coletadas no aplicativo é a *ClassificadorFerrugem* (Figura 21). Dentro desta classe existe uma referência para um objeto do tipo *TensorFlowInterenceInterface*. Este objeto é fornecido pela versão do TensorFlow utilizada no Android. Sua utilidade é executar modelos pré-treinados realizando classificações e previsões de problemas. Este módulo para classificar uma folha é acionado pela classe *ClassificadorActivity*, quando o método Listener *onImageAvailable* é acionado (Quando existe uma nova imagem disponível), a imagem coletada é transmitida para a classe *ClassificadorFerrugem* e todo o processo necessário para classificar a imagem é realizado.

A primeira etapa para realizar a classificação de imagem após coletada é normalizar esta imagem para colocá-la no padrão de entrada de dados exigido pelo modelo de RNA treinado para este projeto. A RNA foi definida e treinada com todas as imagens em uma dimensão de  $224 \times 224$  pixels, com cada pixel possuindo três valores que representam o RGB de cada pixel. A classe de captura de imagem deste projeto já foi definida para capturar imagens nesta resolução, por isso nenhuma alteração no tamanho da imagem foi necessária. Após a imagem estar disponível foi necessário então transformar a imagem coletada para seu formato numérico, que consistiu em uma matriz tridimensional com dimensões de  $224 \times 224 \times 3$ . A altura e largura

desta matriz iriam representar cada *pixel* da imagem e sua profundidade de tamanho três iria representar os valores RGB de cada *pixel*. Após realizar esta transformação a imagem já estava representada em um formato aceito pelo modelo treinado é já seria possível realizar a classificação da imagem coletada.

Como o classificador foi treinado com todas as imagens da base de dados em uma escala de cinza então foi necessário adicionar um passo adicional a mais antes de realizar a classificação da imagem coletada. Esta etapa consistiu em normalizar a imagem coletada para escala de cinza, assim colocando a imagem a ser classificada no mesmo padrão de imagens da base de dados. O processo realizado foi o mesmo aplicado nas imagens da base de treinamento, todos os *pixels* da imagem tiveram seus valores RGB unificados em um único canal através da fórmula aplicada na conversão das imagens durante a etapa de pré-processamento.

### 3.5 APLICAÇÃO DOS TESTES E MÉTRICAS DE AVALIAÇÃO

Após o término do desenvolvimento do aplicativo deu-se início ao processo de testes para que fosse realizada uma avaliação aprofundada dos resultados obtidos pelo aplicativo e o classificador construído. Os testes foram realizados com aplicativo criado e o modelo de RNA resultante do treinamento realizado na etapa 5 (Tabela 1), o qual apresentou a melhor acurácia durante os treinamentos (Tabela 2). Cada folha foi analisada por um período de tempo de aproximadamente 20 segundos. Dentro deste período de tempo cada Frame coletado pela câmera foi classificado e o resultado foi exibido na interface para o usuário. O menor e maior valor resultante das classificações dentro desse período de tempo foram salvos para gerar uma análise da precisão do aplicativo construído. O dispositivo com a aplicação foi posicionado a uma distância próxima a 10 centímetros e colocado em um ambiente com boa iluminação para que a capturas das imagens ocorressem com maior qualidade.

Ao total foram utilizadas 458 imagens para realização dos testes aplicados. Estas imagens utilizadas nos testes não estavam presentes na base utilizada no treinamento do modelo utilizado neste projeto. As imagens de testes estavam divididas em imagens de folhas de soja contaminadas com ferrugem asiática (200 imagens no total), imagens de folhas de soja saudáveis (116 imagens no total) e também folhas contaminadas com outras doenças da cultura da soja como a doença do olho de rã, mancha parda, oídio da soja além de outras doenças não

identificadas, totalizando 142 imagens. O Quadro 5 ilustra a quantidade de imagens de cada classe utilizadas no processo de testes.

Quadro 5: Classes de folhas utilizadas no processo de testes.

<b>Classe</b>	<b>Quantidade</b>
Folhas contaminadas com ferrugem	200
Folhas saudáveis	116
Folhas contaminadas com olho de rã	36
Folhas contaminadas com mancha parda	27
Folhas contaminadas com oídio da soja	35
Folhas contaminadas com doenças não identificadas	44

A fim de validar os resultados obtidos pelas classificações realizadas, foram aplicadas medidas de avaliações para estimar a taxa de acertos e erros do classificador e do aplicativo. Essas medidas foram extraídas a partir dos rótulos empregados nos resultados gerados nas classificações, que foram rotuladas em quatro termos adicionais que são descritos no Quadro 6.

Quadro 6: Termos atribuídos aos resultados das classificações.

<b>Termo</b>	<b>Significado</b>
Verdadeiro positivo (VP)	São as amostras que possuem a ferrugem asiática e foram corretamente classificadas pelo classificador.
Verdadeiro negativo (VN)	São as amostras que não possuem a ferrugem asiática e foram corretamente classificadas pelo classificador.
Falso positivo (FP)	São as amostras que não possuem a ferrugem asiática e foram classificadas como se a tivessem pelo classificador.
Falso negativo (FN)	São as amostras que possuem a ferrugem asiática e foram classificadas que se não tivessem pelo classificador.

Os termos VP e VN indicam as amostras corretamente classificadas pelo classificador e os FV e FN indicam as amostras que foram classificadas de maneira incorreta. Os termos atribuídos aos resultados obtidos nas classificações são importantes para calcular a acurácia e a taxa de erro obtidas, assim como outras métricas de desempenho utilizadas para avaliar este trabalho. A acurácia define a porcentagem de amostras que foram corretamente classificadas,

sejam elas positivas ou negativas. Em contrapartida, a taxa de erro indica o percentual de amostras positivas e negativas que foram incorretamente classificadas. A acurácia e a taxa de erro são calculadas pelas seguintes fórmulas:

$$\text{Acurácia} = \frac{(\text{VP} + \text{VN})}{(\text{VP} + \text{VN} + \text{FP} + \text{FN})}$$

$$\text{Taxa de erro} = \frac{(\text{FP} + \text{FN})}{(\text{VP} + \text{VN} + \text{FP} + \text{FN})}$$

Outra métrica calculada para avaliar este trabalho foi a sensibilidade. A sensibilidade irá descrever a proporção dos casos positivos que foram identificados corretamente. Neste trabalho a sensibilidade indicou a capacidade do aplicativo em detectar a ferrugem asiática da soja quando ela estiver presente em uma folha. A sensibilidade também é conhecida como taxa de verdadeiros positivos, sendo calculada pela seguinte fórmula:

$$\text{Sensibilidade} = \frac{(\text{VP})}{(\text{VP} + \text{FN})}$$

A métrica responsável por calcular a proporção dos casos negativos que foram classificados como positivos é a especificidade, também conhecida como taxa de falsos positivos. Neste trabalho a especificidade foi responsável por indicar a capacidade do aplicativo em classificar corretamente uma folha de soja quando ela não estivesse infectada pela ferrugem asiática da soja. A especificidade é calculada pela seguinte fórmula:

$$\text{Especificidade} = \frac{(\text{VN})}{(\text{VN} + \text{FP})}$$

A métrica de precisão será responsável por analisar a exatidão final do algoritmo usado nas classificações. O cálculo final na exatidão é dado pela seguinte fórmula:

$$\text{Precisão} = \frac{(\text{VP})}{(\text{VP} + \text{FP})}$$

A métrica de revocação, ou (Recall), irá indicar a frequência em que o classificador desenvolvido classifica corretamente os exemplos de uma classe esperada, ou seja, quando a folha estiver infectada com a ferrugem asiática soja, o quão frequente ela será classificada como uma folha que possui a doença esperada, ou ao contrário, quando uma folha da classe de imagens de folhas não contaminadas com a ferrugem asiática fosse classificada. O cálculo da revocação média é dado pela seguinte fórmula:

$$\text{Revocação} = \frac{(\text{VP})}{(\text{VP} + \text{FN})}$$

O Score-F considera a precisão e revocação para indicar um número que indique a qualidade geral do classificador. Esta técnica é eficiente para avaliações em conjuntos onde existem uma alta desproporção dos dados, ou seja, uma classe possui um número muito maior de dados do que a outra. O Score-F se baseia no resultado obtido pela precisão e revocação média. Esta métrica é calculada pela seguinte fórmula:

$$\text{Score - F} = \frac{(2 * \text{Precisão} * \text{Revocação})}{(\text{Precisão} + \text{Revocação})}$$

Todas as métricas citadas a cima foram calculadas para avaliar e definir a capacidade do aplicativo desenvolvido em detectar a ferrugem asiática em folhas de soja. As formulas acima retornam um valor numérico dentro do intervalo [0,1], porem neste trabalho o valor resultante em cada cálculo foi multiplicado por 100 para gerar uma porcentagem dentro do intervalo [0, 100].

## 4 ANÁLISE DE RESULTADOS

Todos os testes realizados foram aplicados separadamente em cada uma das bases de imagens descritas no Quadro 5. Para estimar se uma folha de soja estava contaminada com a ferrugem asiática foi realizado a média simples entre o valor mínimo e máximo obtido de chance de a folha estar com a ferrugem asiática em cada análise realizada e verificado o valor resultante correspondente. Caso a média entre o valor mínimo e máximo resultasse em um número superior a 50% para a classe de folhas com ferrugem esta folha analisada seria considerada como contaminada com ferrugem. Caso o valor resultasse em número superior a 50% para a classe de folhas sem ferrugem a folha seria considerada como não infectada com a ferrugem asiática da soja. Os gráficos ilustrados na Figura 24, Figura 25 e Figura 26 representam os valores mínimos e máximos de chances de uma folha estar contaminada com a ferrugem asiática da soja em cada análise realizada pelo aplicativo nos seis grupos distintos representados no Quadro 5. Nestes gráficos representados quanto mais próximo do ponto (100, 100) maior era a certeza de a folha estar contaminada com a ferrugem asiática e quanto mais próximo do ponto (0, 0) maior as chances da folha não possuir a doença.

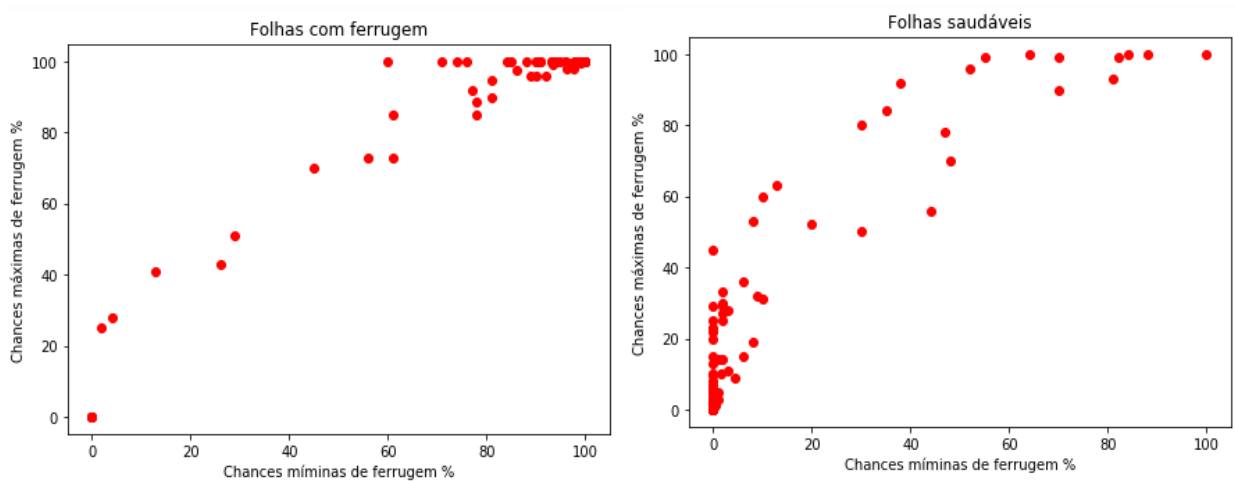


Figura 24. Intervalo mínimo e máximo nas classificações de folhas contaminadas com ferrugem e saudáveis.

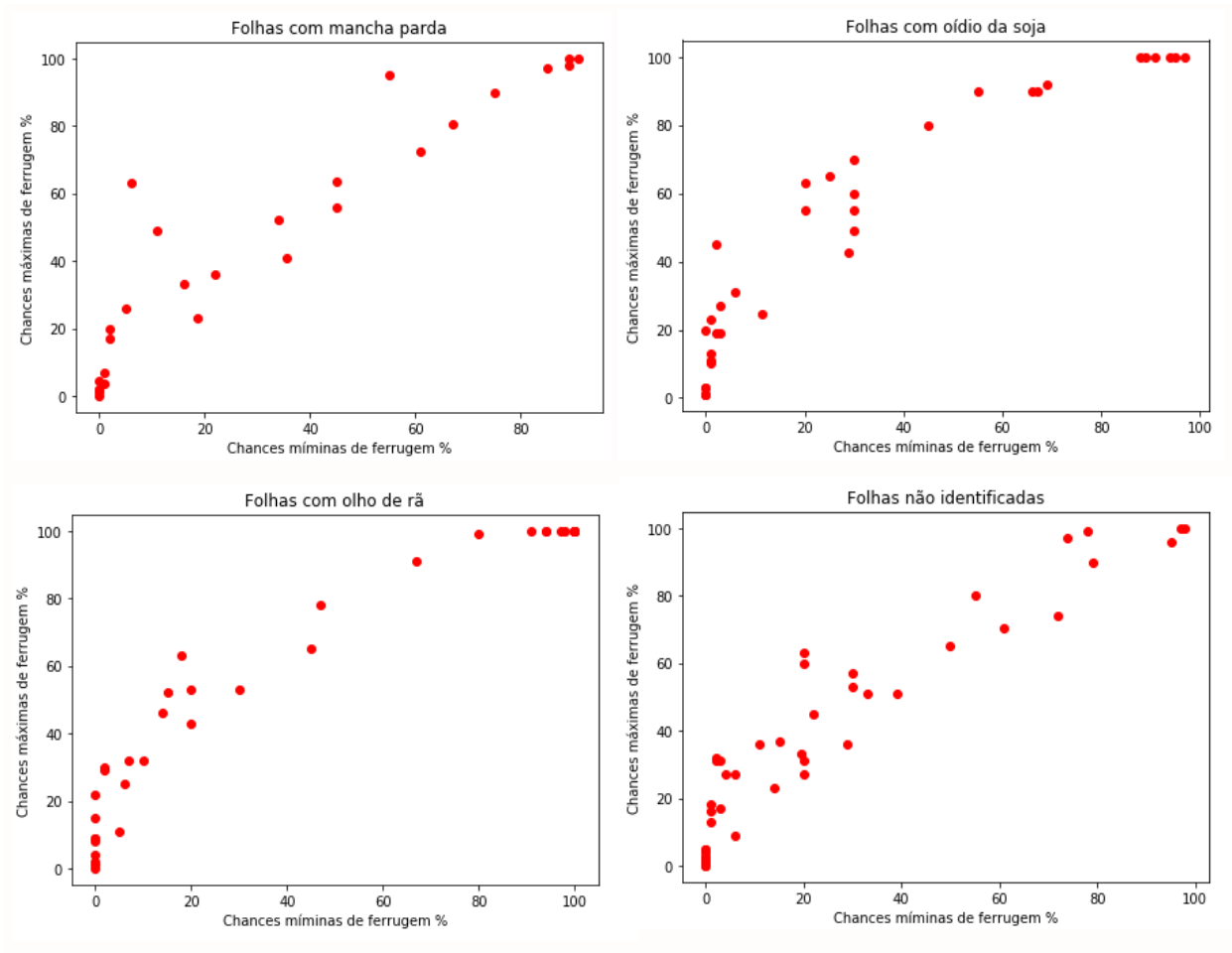


Figura 25. Intervalo mínimo e máximo nas classificações de folhas contaminadas com mancha parda, oídio da soja, olho de rã e manchas causadas por outras doenças.

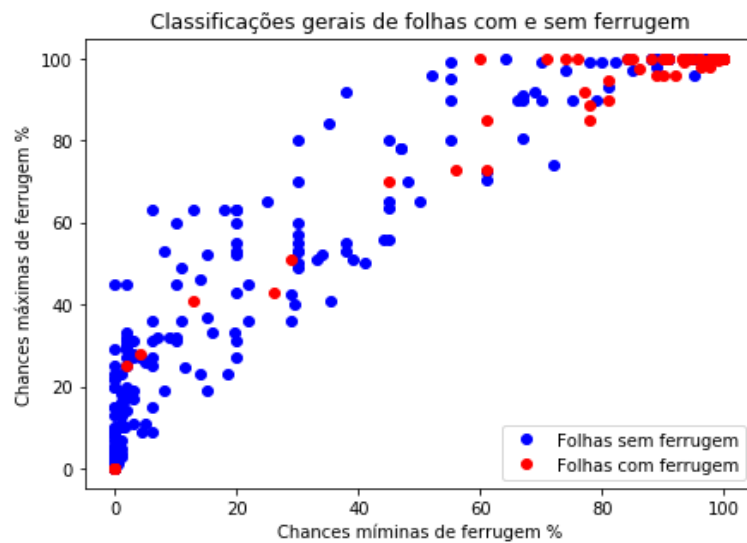


Figura 26. Intervalo mínimo e máximo nas classificações de folhas contaminadas com a ferrugem e com folhas não contaminadas com a ferrugem.



Os gráficos de pontos acima ilustraram visualmente como se comportou cada classe de imagem neste problema. Nestas ilustrações foi possível observar que as classes de folhas contaminadas com ferrugem e de folhas saudáveis tiveram melhores resultados nas análises realizadas, a classe de folhas com ferrugem obteve mais amostras próximas ao ponto (100, 100) que indica ferrugem, e as folhas totalmente saudáveis ficaram mais próximas do ponto (0, 0) que representa uma folha não contaminada com ferrugem.

As classes de imagens que possuíam lesões causadas por outro tipo de doença que não fosse a ferrugem asiática da soja, ou folhas com algum tipo de mancha específica apresentaram maior dificuldade nas análises realizadas, apresentando uma taxa de FP maior do que as demais classes de imagens analisadas (Estes dados são descritos precisamente na Tabela 3). O motivo da menor taxa de acerto nestas classes de imagens é a maior semelhança existente na textura de cada folha, alguns tendo uma distinção quase imperceptível visualmente. Nos níveis mais baixos de severidade, a ferrugem asiática da soja e mancha parda, por exemplo, são quase impossíveis de distinguir visualmente, podendo ter a necessidade de algum tipo de análise que não seja a visual para definir qual é a doença específica presente na folha. Os gráficos ilustrados nas imagens da Figura 27, Figura 28 e Figura 29 ilustram as classificações realizadas durante os testes com base na média entre o valor mínimo e máximo de chance de a folha estar contaminada com a ferrugem asiática da soja.

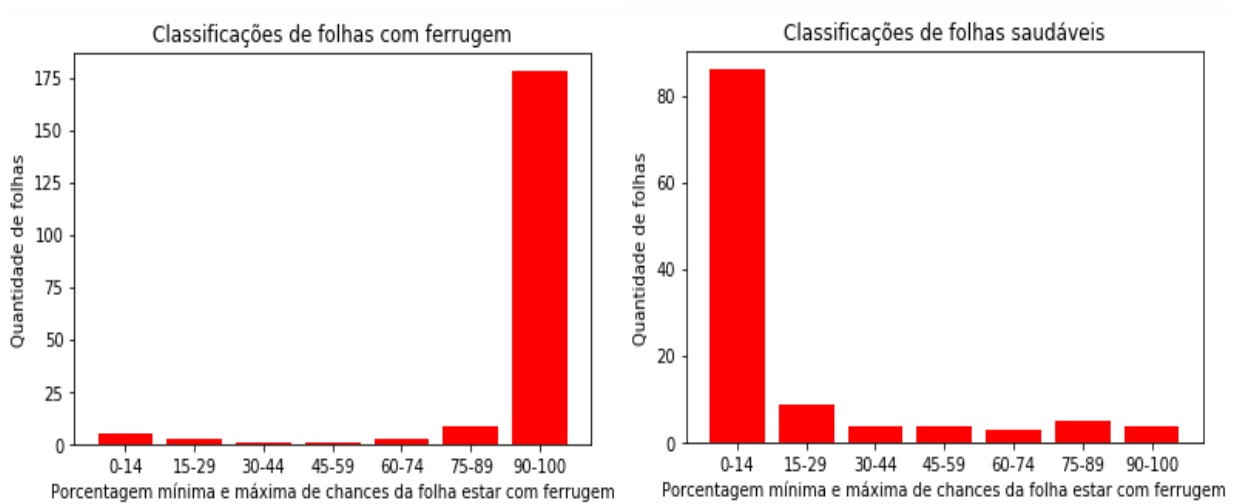


Figura 27. Classificações de folhas com ferrugem e saudáveis com base na média entre valor mínimo e máximo capturado na análise de cada folha.

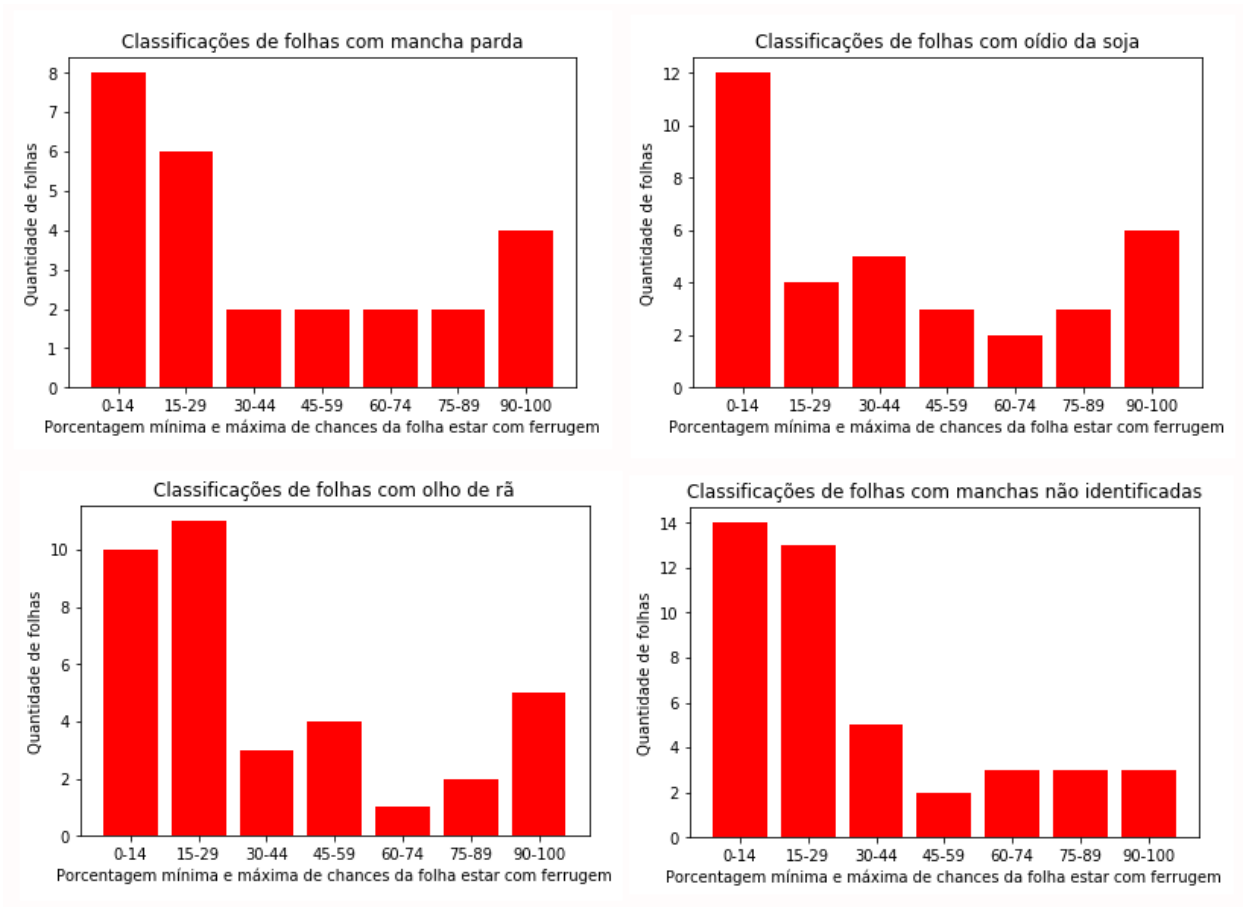


Figura 28. Classificações de folhas com mancha parda, oídio da soja, olho de rã e manchas causadas por outras doenças com base na média entre valor mínimo e máximo capturado na análise de cada folha.

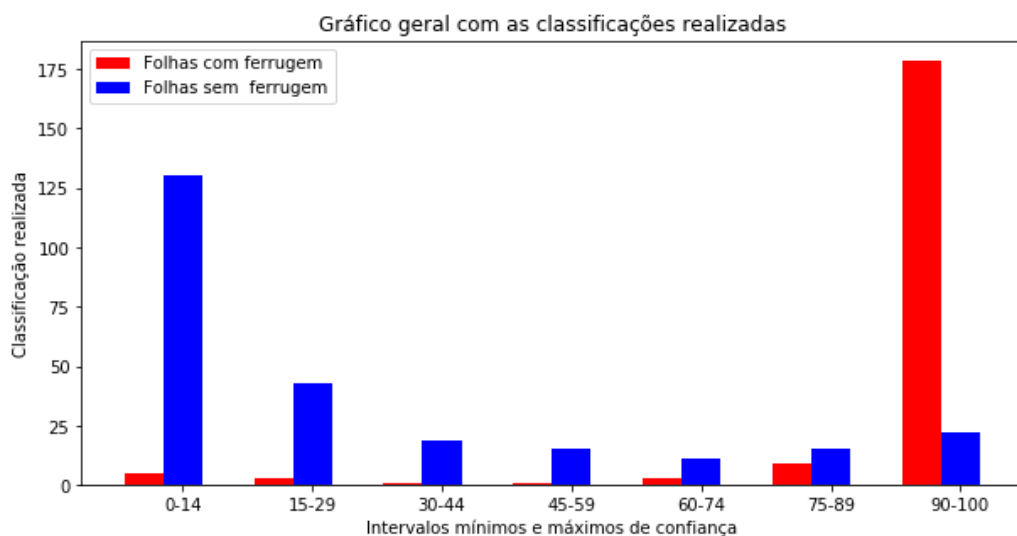


Figura 29. Classificações gerais realizadas de folhas com e sem ferrugem com base na média entre valor mínimo e máximo capturado na análise de cada folha.

Tabela 3 – Resultados gerais obtidos em testes divididos por cada classe de imagem.

<b>Classe de imagem</b>	<b>VP</b>	<b>VN</b>	<b>FP</b>	<b>FN</b>
Ferrugem	191	0	0	9
Saudável	0	100	15	0
Mancha parda	0	16	10	0
Olho de rã	0	21	15	0
Oídio da soja	0	24	11	0
Outras classes	0	33	10	0
<b>Totais</b>	<b>191</b>	<b>194</b>	<b>61</b>	<b>9</b>

Como representado na Tabela 3, ao total foram classificadas 191 VP (Folha com ferrugem que foram corretamente classificadas), 194 VN (Folhas sem ferrugem corretamente classificadas), 61 FP (folhas sem ferrugem que foram classificadas como contaminadas com ferrugem) e 9 FN (Folhas contaminadas com ferrugem que foram classificadas como se não estivessem contaminadas com a doença). Como demonstrado nos dados coletados em teste, pode-se analisar que, as maiores taxas de erro aconteceram na classificação de folhas que não estavam contaminadas com a ferrugem asiática da soja (FPs), dados especificados na Tabela 4.

Com base nos números de VP, VN, FP e FN coletados nas classificações se obteve uma acurácia geral de 84,61% e uma taxa de erro geral de 15,39% (Tabela 4). Além da acurácia e taxa de erro geral foram calculadas a acurácia e taxa de erro individual para cada classe de imagens presentes no Quadro 5. Estes dados são representados na Tabela 4. O resultado da acurácia obtido nos testes foi próximo a obtida da etapa 5 de treinamento da Tabela 2, etapa que resultou no modelo classificador que foi utilizado no aplicativo desenvolvimento para realização dos testes. A validação do treinamento alcançou uma acurácia de 87,9%, sendo 3,29 % superior a acurácia alcançada sobre a base de imagens utilizada para testes, que não fez parte da etapa de treinamento. Esta diferença entre o resultado da acurácia dos testes e do treinamento é considerada baixa para o problema em questão, o que significa que o classificador construído é capaz de acertar novas classificações de imagens de folhas de soja que não estavam na base de dados de treinamento.

As principais dificuldades encontradas de acordo com os dados coletados foram em realizar a diferenciação de uma folha contaminada com outra doença com lesões semelhantes a

ferrugem asiática e uma folha que estava contaminada com a ferrugem asiática. Esta dificuldade foi encontrada devido a semelhanças de cores existentes entre as lesões causadas pelas doenças que estavam na base de imagens, e também em alguns casos, o próprio formato das lesões cauda se assemelhava muito nas doenças comparadas com a ferrugem asiática da soja.

Tabela 4 – Acurácia e taxa de erro obtidos em testes dividido por cada classe de imagem.

<b>Classe de imagem</b>	<b>Acurácia (%)</b>	<b>Erro (%)</b>
Ferrugem	95,50	4,50
Saudável	86,96	13,04
Mancha parda	61,53	38,46
Olho de rã	58,34	41,66
Oídio da soja	68,57	31,42
Outras classes	76,75	23,25
<b>Base de testes geral</b>	<b>84,61</b>	<b>15,39</b>

Além da acurácia e taxa de erro apresentados na Tabela 4, as demais métricas de avaliação citadas na seção 3.5 foram calculadas sobre os resultados gerais obtidos nas análises sobre os testes realizados. As métricas foram calculadas a partir do número de VP, VN, FP, FN resultantes dos testes na base de imagens de folhas com ferrugem e demais imagens de folhas sem ferrugem presentes nos testes, onde uma classe de imagem era as folhas com ferrugem, as quais deveriam ser classificadas como positivas, e a outra classe era a de folhas sem ferrugem.

A Tabela 5 apresenta os valores calculados de tais métricas.

Tabela 5 – Resultados obtidos sobre as classificações realizadas.

<b>Métrica de avaliação</b>	<b>Resultados obtidos (%)</b>
Sensibilidade	95,50
Especificidade	76,07
Precisão	75,79
Revocação média	85,56
Score-F	80,48

A partir dos dados apresentados na Tabela 5 pode-se analisar que a taxa mais alta alcançada foi na métrica de sensibilidade. A taxa de sensibilidade alcançada indicou que em

95,50 % dos casos em que a folha analisada estava infectada com a ferrugem asiática da soja o aplicativo foi capaz de classificá-la corretamente. A taxa de especificidade obtida indicou que o aplicativo em 76,07% das classificações que foram realizadas sobre folhas de soja que não estavam contaminadas com a ferrugem asiática da soja se obteve a resposta esperada.

A taxa de precisão calculada indicou que em 76,79% das vezes em que o aplicativo classificou uma imagem como se ela estivesse contaminada com a ferrugem asiática da soja, o aplicativo estava correto na classificação. Enquanto a revocação média especificou que em 85,79% dos casos a resposta dada pelo aplicativo foi de acordo com a resposta esperada pela classe da imagem que estava sendo classificada. Baseado no valor obtido pela precisão e pela revocação média, foi obtido um Score-F final de 80,48%. Esta métrica é uma alternativa a acurácia, que leva em consideração não só os resultados positivos, mas também a capacidade do classificador em classificar ou não uma classe especificada, e também o número de amostras de cada classe no problema.

A partir de uma comparação realizada entre o valor da sensibilidade e especificidade obtidos (Tabela 5), pode-se concluir que, o aplicativo foi mais eficiente para realizar classificações de folhas que estavam contaminados com a ferrugem asiática, do que na realização de classificações de folhas que estão estavam contaminadas com a ferrugem asiática. Como descrito nos resultados da Tabela 4, os menores índices de acertos aconteceram nas classificações de outras doenças presentes na cultura da soja. A menor taxa de acertos nestas classes de imagens é justificada pela semelhança existente nos padrões visuais das lesões dessas doenças quando comparadas com a ferrugem asiática, e também o fato de que a quantidade de imagens dessas doenças presentes na base de treinamento e testes não serem tão grandes quanto a quantidade de imagens de folhas de soja com a ferrugem asiática e de folhas saudáveis.

## 5 CONCLUSÃO

A partir dos resultados obtidos neste trabalho pode se comprovar que o modelo de CNN Inception-V3 foi eficiente no reconhecimento de folhas saudáveis e com a ferrugem asiática da soja. Na identificação de outros tipos de doenças da cultura da soja o Inception-V3 se mostrou menos eficiente apesar de que em todas as classes presentes na base de testes o modelo conseguiu classificar corretamente a maioria dos casos de teste. O modelo de CNN utilizado e o TensorFlow se mostraram extremamente rápidos para realizar classificações de imagens, no presente estudo após uma imagem ser coletada a resposta entregue pelo aplicativo demorava aproximadamente de 2 a 3 segundos. O estudo realizado comprova a viabilidade da utilização destas tecnologias no uso da detecção da ferrugem asiática na folha da soja, assim como também servirá como referência teórica e prática para a construção de novos projetos envolvendo estas tecnologias com propósitos semelhantes.

Apesar de após comprovado com este estudo a capacidade de detectar a ferrugem asiática da soja utilizando o Inception-V3 e o TensorFlow a partir de imagens coletadas em tempo real pela câmera de um dispositivo com Android, algumas importantes particularidades observadas neste estudo devem ser consideradas. O aplicativo desenvolvido apresentou dificuldades em realizar classificações de maneira correta quando colocado a uma distância superior a 20 centímetros da folha a ser analisada. O fator da qualidade da câmera utilizada durante os testes realizados podem ter influenciado neste caso específico, já que as imagens que eram obtidas quando a câmera estava a mais de 20 centímetros da folha de soja não apresentavam os detalhes e características da folha com uma alta qualidade. Outro caso em que o aplicativo teve dificuldades em realizar classificações de maneira correta foi quando a folha de soja estava com pouca iluminação sobre ela. Com a baixa iluminação sobre a folha a câmera do dispositivo não conseguiu coletar as imagens com detalhes que descrevem a doença presente na folha, desta forma as classificações de imagens com uma baixa iluminação não se mostraram eficientes. Porém mesmo sem a utilização de um ambiente controlado para testes os resultados ainda se mostraram satisfatórios.

Em uma sequência futura deste projeto esperasse abrir a possibilidade da inclusão de novas imagens na base de treinamento, aumentando desta forma a capacidade do aplicativo em detectar a ferrugem asiática da soja com maior precisão. Também é desejado implementar uma

versão do aplicativo para a plataforma IOS, já que atualmente o TensorFlow já permite uma integração com esta plataforma. A expansão para plataforma irá permitir atingir um número maior de usuários, ajudando mais pessoas a detectarem a doença em lavouras de soja quando necessário. Também em versões futuras do aplicativo é desejável implementar uma plataforma de Feedback dentro do aplicativo, possibilitando o usuário relatar os resultados das classificações geradas, assim como as imagens que foram classificadas. Esta plataforma de Feedback também servirá para melhorar a performance do aplicativo, já que as imagens e respostas relatadas pelos usuários poderão ser utilizadas para alimentar a base de imagens de treinamento do aplicativo e gerar um classificador ainda mais eficiente. Também é desejado que futuramente este projeto seja testado por algum profissional especializado em doenças de culturas agrícolas, fornecendo desta forma, uma avaliação de um profissional especializado da área sobre o projeto desenvolvido.

## REFERÊNCIAS

ABADI, Martín et al. Tensorflow: A system for large-scale machine learning. In: **12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)**. 2016. p. 265-283.

AGROLINK. Sintomas. 2016. Disponível em: <[https://www.agrolink.com.br/culturas/soja/informacoes/sintomas\\_361550.html](https://www.agrolink.com.br/culturas/soja/informacoes/sintomas_361550.html)> Acesso em: 30 ago. 2018.

ALBUQUERQUE, M. P. A.; CANER, E. S.; MELLO, A. G.; ALBUQUERQUE, M. P. Análise de Imagens e Visão Computacional. **Anais da V Escola do CBPF. 1ed. Rio de**, v. 1, p. 145-176, 2012.

AMAKDOUF, Hicham et al. Classification and recognition of 3D image of Charlier moments using a multilayer perceptron architecture. **Procedia Computer Science**, v. 127, p. 226-235, 2018.

ANDRADE, W. et al. Segmentação baseada em Textura e Watershed aplicada a Imagens de Pólen. 2011.

ANANTRASIRICHAI, Nantheera; HANNUNA, Sion; CANAGARAJAH, Nishan. Automatic Leaf Extraction from Outdoor Images. **ArXiv preprint arXiv: 1709.06437**, 2017.

ASSAO, F. M. **Aprendizado semi-supervisionado e não supervisionado par análise de dados de expressão gênica**. 2008. Dissertação (Mestrado em Ciência da Computação) – Centro de ciências exatas e de tecnologia, Universidade federal de São Carlos.

BARELLA, C. F. **Abordagens estatísticas aplicadas ao mapeamento de susceptibilidade a movimentos de massa: Análise de diferentes técnicas no contexto do quadrilátero ferrífero**. 2016. Tese (Doutorado em Geotécnica) – Núcleo de Geotécnica, Universidade Federal de Ouro Preto, Ouro Preto.

BORTH, M. R.; LACIA, J. C.; PISTORI, H.; RUVIANO, C. F. A visão computacional no agronegócio: Aplicações e direcionamentos. **7º Encontro Científico de Administração, Economia e Contabilidade (ECAECO), 2014**.

BRAGA, A. de P. **Redes neurais artificiais: teoria e aplicações**. Livros Técnicos e Científicos, 2000.

CAMARGOS, R. Ferrugem asiática da soja. **Informativo técnico**. 3ed.

CASSIO, L. L. **Técnicas de visão computacional aplicadas ao reconhecimento de cenas naturais e locomoção autônoma em robôs agrícolas móveis**. 2011. Dissertação (Mestrado



em computação aplicada) – Escola de engenharia de São Paulo, Universidade de São Paulo, São Carlos.

CASSOL, Cidinei Luiz; AGOSTI, Cristiano. Detecção De Ferrugem Asiática Na Soja Analisando A Imagem Da Folha Utilizando Redes Neurais. *In: X CONGRESSO BRASILEIRO DE AGROINFORMÁTICA*, 2015, Ponta Grossa. **XSBIAGRO**. Ponta Grossa: [s. n.], 2015. E-book.

CHAGAS, E. M. P. F.; RODRIGUES, D. L. **Método de segmentação de objetos em imagens baseado em contornos ativos e algoritmo genérico**. 2018. Programa de Pós-Graduação em Engenharia Mecânica – Pontifca Universidade Católica, Minas Gerais.

CAVANI, F. A. **Análise de cenas de pomares de laranjeiras através de segmentação de imagens e reconhecimento de padrões**. 2007. Dissertação (Mestrado em engenharia mecânica) – Escola de engenharia de São Carlos, Universidade de São Paulo, São Carlos.

DALL'AGNOL, Amélio et al. O complexo agroindustrial da soja brasileira. **Embrapa Soja-Circular Técnica (INFOTECA-E)**, 2007.

Embrapa. Soja em números (safra 2017/2018). Disponível em: <<https://www.embrapa.br/en/soja/cultivos/soja1/dados-economicos>> Acesso em: 26 ago. 2018.

EMBRAPA. Tecnologias de Produção de Soja – Região Central do Brasil 2012 e 2013. **Londrina: Embrapa Soja**, 2011.

FERREIRA, Alessandro dos Santos. **Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja**. 2017. Dissertação de Mestrado.

FERREIRA, Anselmo Castelo Branco et al. Um estudo comparativo de segmentação de imagens por aplicações do corte normalizado em grafos. 2011.

FORSYTH, D. A.; PONCE, J. **Computer Vision: A Modern Approach**. 2nd Edition. Prentice Hall, 2011.

FRANCO, J. R. **Método computacional para identificação do fungo Cercospora Kikuchii em sementes de soja**. 2017. Dissertação (Mestrado em computação aplicada) – Setor de Ciências Agrárias e de Tecnologia, Universidade Tecnológica Federal do Paraná, Ponta Grossa.

FREITAS. M. C. M. A cultura da soja no Brasil: O crescimento da produção brasileira e o surgimento de uma nova fronteira agrícola. **Enciclopédia Biosfera-Centro Científico Conhecer, Goiânia-GO**. v. 7, n. 12, p. 1-12, 2011.

GARTNER. Smartphone shipments fail to impress in Q4 2018. 2018. Acesso em: <[https://www.gsmarena.com/gartner\\_smartphone\\_shipments\\_fail\\_to\\_impress\\_in\\_q4\\_2018-news-35627.php](https://www.gsmarena.com/gartner_smartphone_shipments_fail_to_impress_in_q4_2018-news-35627.php)>

GARDNER, Matt W.; DORLING, S. R. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. **Atmospheric environment**, v. 32, n. 14-15, p. 2627-2636, 1998.

GU, Jiuxiang et al. Recent advances in convolutional neural networks. **Pattern Recognition**, v. 77, p. 354-377, 2018.

HAYKIN, S. **Redes Neurais: Princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001.

HIJAZI, Samer; KUMAR, Rishi; ROWEN, Chris. Using convolutional neural networks for image recognition. **Cadence Design Systems Inc.: San Jose, CA, USA**, 2015.

INVASIVE. INVASIVE, 2019. Disponível em <https://www.invasive.org>. Acesso em: 20 jan. 2019.

IPMIMAGES. IPM IMAGES, 2019. Disponível em <https://www.ipmimages.org/>. Acesso em: 20 jan. 2019.

ISLAM, Mohammad Shakirul et al. InceptB: A CNN Based Classification Approach for Recognizing Traditional Bengali Games. **Procedia computer science**, v. 143, p. 595-602, 2018.

LACERDA, V. S. **Estimativa do índice de severidade de ferrugem asiática na cultura da soja por meio de imagens obtidas com aeronave remotamente pilotada**. 2017. Dissertação (Mestrado em Computação Aplicada) – Setor de Ciências Agrárias e de Tecnologia, Universidade Tecnológica Federal do Paraná, Ponta Grossa.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436, 2015.

MARENGONI, M. **Introdução a visão computacional usando openCV**. RITA, v. 13, n.1, 2009.

MATSUNAGA, V. Y. **Curso de redes neurais utilizando o Matlab**. 2012.

MELO, G. A. **Utilização de processamento digital de imagens e redes neurais artificiais para o reconhecimento de índices de severidade da ferrugem da soja**. 2015. Dissertação (Mestrado em Computação Aplicada) - Setor de Ciências Agrárias e de Tecnologia, Universidade Tecnológica Federal do Paraná, Ponta Grossa.

MELO, G. A.; MATHIAS, I. M.; DIAS, A. H. Uso de Redes Neurais Artificiais para o reconhecimento automático do Oídio da Soja via smartphone com o auxílio da biblioteca OpenCv. *In: XSBIAGRO*, 2015, Ponta Grossa. 2015.

PENG, Min et al. Dual temporal scale convolutional neural network for micro-expression recognition. **Frontiers in psychology**, v. 8, p. 1745, 2017.

PINTO, Tiago Willian. Segmentação de imagens digitais combinando watershed e corte normalizado em grafos. 2014.

PIRES, R. D. L. **Reconhecimento de doenças da cultura da soja usando vetores localmente agregados e atribuições leves**. 2017. Dissertação (Mestrado em Ciência da Computação) – Faculdade de computação, Universidade Federal do Mato Grosso do Sul, Campo Grande.

SANCHES, Altevir Costa; MICHELLON, Ednaldo; ROESSING, Antonio Carlos. Os limites de expansão da soja. **Informe GEPEC**, v. 9, n. 1, 2004.

SANCHES, M. K. **Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados**. 2003. Dissertação (Mestrado em Computação) – ICMC-USP, Universidade de São Paulo, São Paulo.

SEO, Yian; SHIN, Kyung-shik. Hierarchical convolutional neural networks for fashion image classification. **Expert Systems with Applications**, v. 116, p. 328-339, 2019.

SILVA, L. N. C. **Análise e síntese de estratégias de aprendizado para redes neurais artificiais**. 1998. Dissertação (Mestrado em Computação) – Faculdade de engenharia elétrica e de computação, Universidade estadual de Campinas, Campinas.

SILVA, Oscar Paesi da. Redimensionamento de imagens preservando a proporção dos objetos. 2009.

STRECK, N. A.; PAULA, G.; CAMERA, C.; MENEZES, N. L.; LAGO, I. Estimativa do plastocrono em cultivares de soja. **Bragantina**, Campinas, v. 76, p.67-73, 2008.

TENSORFLOW. An open source machine-learning library for research and production. **TensorFlow**. 2018. Disponível em: <<https://www.tensorflow.org/?hl=pt-br>>. Acesso em: oct. 2018.

USDA – UNITED STATES DEPARTMENT OF AGRICULTURE. FAS Databases. Disponível em: <<https://www.fas.usda.gov/data/databases>>. Acesso em: 26 ago. 2018.

VIANNA, M. E. **Calibração de Sistemas de Visão Computacional para Aplicação em Automação e Robótica**. Dissertação de Mestrado – Pontífca Universidade Católica de Minas Gerais (PUC-MG), 2009.

YORINORI, J. T.; NUNES JUNIOR, J.; LAZZAROTTO, J.J. Ferrugem “asiática” da soja no Brasil: evolução, importância econômica e controle. Londrina: Embrapa Soja, 2004.