

# IDENTIFICAÇÃO DE DOENÇAS EM PLANTAS POR MEIO DE PROCESSAMENTO DE IMAGENS: Redes Neurais Convolucionais como Auxílio à Agricultura

## *IDENTIFICATION OF DISEASES IN PLANTS THROUGH IMAGE PROCESSING: Convolutional Neural Networks as Aid to Agriculture*

Moisés J. SILVA

[jsilva.moises@gmail.com](mailto:jsilva.moises@gmail.com)

Pós-graduação em BI e Bigdata  
Centro Universitário Anchieta

Prof. Dr. Juliano SCHIMIGUEL

[schimiguel@gmail.com](mailto:schimiguel@gmail.com)

Professor no Centro Universitário Anchieta,  
e Universidade Cruzeiro do Sul

### RESUMO

Os sistemas inteligentes têm ganhado espaço em diversos segmentos da sociedade, desde sistemas autônomos implementados em veículos que reconhecem placas e obstáculos até sistemas críticos voltados à medicina. Tendo em vista o crescimento populacional e a crescente demanda por alimentos aliado ao potencial dessa tecnologia essa pesquisa busca se utilizar das pesquisas voltadas às técnicas de visão computacional para desenvolver um sistema destinado à agricultura com o objetivo de identificar doenças em plantas por meio das técnicas de inteligência artificial baseado em algoritmos de redes neurais, com o objetivo de oferecer uma ferramenta acessível aos pequenos produtores.

#### **Palavras-chaves:**

agricultura; inteligência artificial; redes neurais.

### ABSTRACT

The intelligent systems have gained space in various segments of society, since autonomous systems implemented in vehicles that recognize signs and barriers to critical systems geared to medicine. Having seen the population growth and the increasing demand for food coupled with the potential of this technology, this research seeks to use the research focused on the techniques of computer vision to develop a system for agriculture, with the objective of identifying diseases in plants by means of the techniques of artificial intelligence based on algorithms of neural networks, with the objective of providing a handy tool for small producers.

#### **Keywords:**

agriculture; artificial intelligence; neural networks.

## INTRODUÇÃO

A visão computacional tem ganhado destaque em diversas áreas, desde os veículos autônomos que são conduzidos por meio de inteligência artificial (IA) identificando placas, semáforos, pedestres e outros obstáculos a tecnologia pode ser útil em outros diversos segmentos.

Conforme (HAYKIN, 2001), as Redes Neurais Artificiais (RNA's) são tecnologias que têm suas raízes em diversos campos da ciência, englobando neurociência, estatística, física, matemática, ciência da computação e engenharia.

A capacidade que algoritmos computacionais inteligentes possuem de tomar decisões baseadas em aprendizado de máquina, permitem que os mais diversos campos do conhecimento humano possam se beneficiar das técnicas avançadas de IA.

De acordo com (SILVA & al., 2010) apenas a partir dos anos 1990 essa tecnologia teve maior interesse dos pesquisadores, apesar de os princípios de IA terem mais de 50 anos.

As técnicas de computação inteligente têm contribuído positivamente em áreas críticas como por exemplo a medicina, de acordo com (SANTOS, 2017) a construção de uma rede neural convolucional permite o diagnóstico médico por meio de processamento de imagens indicando se uma lesão de pele é um melanoma ou não.

O avanço tecnológico permite que pessoas e empresas sejam mais produtivas e proporcionam a redução de custos inerentes da atividade exercida. Diante dos resultados positivos observados da aplicação de técnicas de visão computacional esse artigo apresenta como essas técnicas podem ser aplicadas no setor rural afim de oferecer uma ferramenta barata de identificação de doenças em plantas.

### **Algoritmos Inteligentes**

Os algoritmos inteligentes são construídos utilizando em seu núcleo modelos matemáticos que permitem que ao longo do tempo os resultados apresentados sejam aperfeiçoados. Conforme (BRAGA & al., 2014), Warren MCCulloch e Walter Pitts são os pioneiros do paradigma de IA.

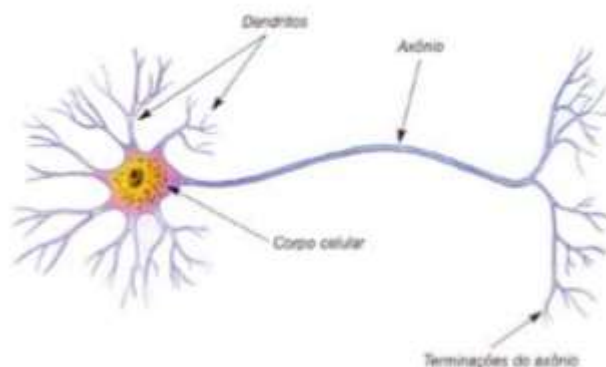
A área de IA é bastante vasta possuindo diversos algoritmos destinados aos mais variados problemas da sociedade, desse modo é importante conhecer como cada um funciona para que o algoritmo correto seja aplicado ao problema em questão.

### **Redes Neurais**

As RNA's são uma inspiração no funcionamento dos neurônios biológicos, em diversos momentos os avanços ocorrem pela observação do comportamento da natureza. Uma aeronave é uma das tecnologias que surgiram da busca pela compreensão do funcionamento da natureza.

Conforme (SILVA & al., 2010) a tecnologia tem potencial multidisciplinar com potencial de modificar os paradigmas de como as coisas são desenvolvidas se utilizando de algoritmos de IA.

A Figura 1 representa a estrutura de um neurônio biológico, composto por dendritos, axônio, corpo celular e terminações do axônio. Cada estrutura dessa tem função que pode ser adaptada para um modelo computacional.



*Figura 1 - Representação de um Neurônio Biológico*

*Fonte: (ROCHA, 2016)*

- **Dendritos:** Os dendritos são estruturas que recebem os sinais vindos de outros neurônios;
- **Corpo celular:** O Corpo celular é o meio onde ocorre a combinação dos sinais recebidos de outros neurônios, onde caso a combinação das entradas estejam acima de um limiar um impulso elétrico é enviado por meio do axônio;
- **Axônio:** estrutura responsável pelo transporte do impulso elétrico até suas terminações;
- **Terminações do axônio:** meio pelo qual é propagado o impulso elétrico recebido através do axônio.

Conforme (BRAGA & al., 2014) devido a diferentes concentrações de potássio no interior da célula em relação ao de sódio no exterior da célula é possível medir a diferença de potencial em volts. Um neurônio biológico emite um sinal de ativação quando há inversão no fluxo de potássio e sódio que se invertem tornando interior da célula positivo em relação a área externa.

Similar a um neurônio biológico, os neurônios artificiais possuem estruturas básicas que quando em conjunto formam a rede neural. A Figura 2 representa a estrutura de um neurônio computacional.

No modelo computacional os dendritos podem ser representados como os sinais de entrada, o corpo celular representado pela junção aditiva onde ocorre a soma ponderada das entradas e então é enviado para a função de ativação que nesse caso cumpre a função do axônio e por fim as terminações do axônio podem ser representadas pela saída da função de ativação.

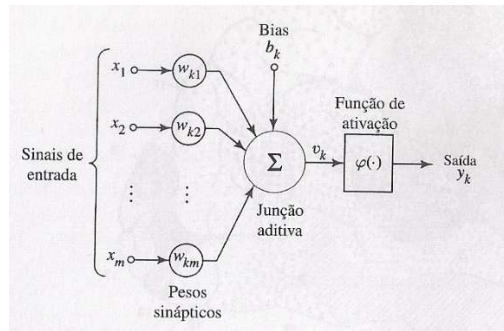


Figura 2 - Estrutura de um neurônio artificial

Fonte: (HAYKIN, 2001)

### Aprendizado Supervisionado

O aprendizado supervisionado é utilizado em algoritmos em que se possuem um conjunto de dados onde já existe um conhecimento prévio do comportamento dos mesmos de modo que os dados possam ser separados em um modelo  $\{key: value\}$ . Nesse modelo os dados são separados em duas partes, ou seja, valores que são variáveis de entrada e para cada entrada existe um valor de saída esperado.

A Figura 3 demonstra como os dados são organizados para que sejam processados em uma rede neural, nesse exemplo é informado ao modelo os dados de entrada que nesse caso são representados por  $x$  e  $y$  e a qual classe eles pertencem representados por  $z$ .

Entradas		Classe
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Figura 3 - Exemplo de dados com classes

Fonte: (GRANATYR, 2020)

### Aprendizado não Supervisionado

O Aprendizado não supervisionado é implementado quando existem um conjunto de dados em que não há conhecimento de como eles se comportam, ou seja, não existem *labels* que indicam ao sistema a qual classe um determinado registro pertence, ao invés disso o sistema deve se organizar e buscar encontrar clusters que representem os dados, baseando-se nos padrões encontrados durante o processamento.

Conforme (SILVA & al., 2010) esses algoritmos podem encontrar por si uma determinada quantidade de cluster ou o projetista pode predefinir quantos devem ser formados, caso tenha algum conhecimento do problema a ser tratado.

A Figura 4 demonstra a clusterização do banco de dados íris através da utilização do algoritmo *K-means* com objetivo de fazer separação dos dados em 3 clusters. A base de dados íris é muito utilizada na área de ciência de dados para aprendizado de máquina.

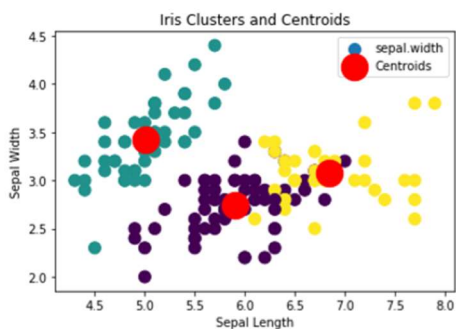


Figura 4 - Cluster criado com algoritmo *K-means*

Fonte: (GRANATYR, 2020), adaptado pelo autor.

### Aprendizado por Reforço

O aprendizado por reforço é um modelo em que um sistema é construído para que aprenda por tentativa e erro, nesse modelo o sistema trabalha com recompensas quando durante o processo de tomada de decisão o mesmo realiza uma ação corretamente.

Uma forma de compreender o funcionamento desse algoritmo é utilizar um game como exemplo, supondo que esse jogo possua como interações possíveis o pressionar das teclas esquerda, direita, para cima e para baixo o sistema irá realizar essas ações sem saber qual momento certo de pressionar cada uma delas quando o processo de aprendizado é iniciado.

Nesse exemplo o jogo começa realizando ações de forma aleatória, pressionando as teclas mencionadas anteriormente e os pesos da rede começam a serem ajustados à medida que o sistema começa a tomar decisões e o sistema começa a receber recompensas nos momentos em que completou uma ação corretamente.

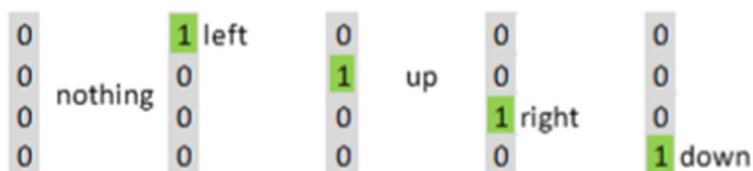


Figura 5 - Saídas possíveis de uma rede para game

Fonte: Próprio

A Figura 5 exemplifica as possíveis saídas de uma RNA com aprendizado por reforço, além das quatro possibilidades de ações em que algumas das teclas é pressionada existe a ação não fazer nada, já que durante o jogo nem sempre alguma tecla é pressionada. Durante o aprendizado o sistema irá identificar que algumas ações poderão ser combinadas como por exemplo acionar simultaneamente as teclas para cima e direita.

## Funções de Ativação

Funções de ativação são componentes importantes em uma rede neural, dependendo do problema o programador deverá saber qual se encaixa no problema a ser resolvido. Existe uma variedade de funções que são utilizadas em redes neurais, porém são destacados nesse trabalho apenas 3 dessas funções e que são as mais utilizadas.

### Step Function (Função degrau)

A *step function* é utilizada quando o problema a ser resolvido possui apenas duas classes e sua saída será 0 ou 1 para indicar a qual classe uma entrada pertence, o gráfico da equação (1) pode ser observado na Figura 6.

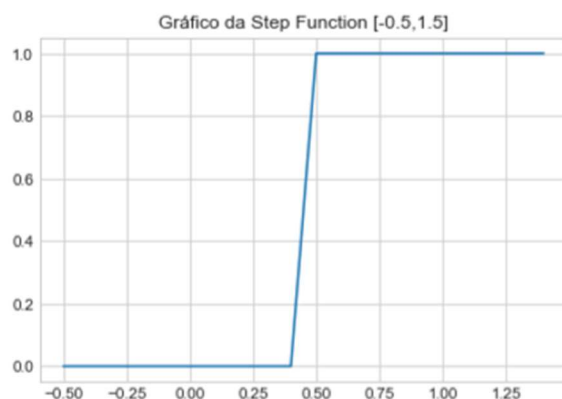


Figura 6 - Gráfico da step function (função degrau)

Fonte: (GRANATYR, 2020)

$$\begin{aligned}\varphi(\vartheta) &= (\vartheta < 0.5) = 0 \\ \varphi(\vartheta) &= (\vartheta \geq 0.5) = 1\end{aligned}\quad (1)$$

## Função Sigmoid

A função sigmoid é uma das funções mais utilizadas, ela oferece como retorno valores entre 0 e 1. Essa função geralmente é utilizada em problemas que compreendem mais de uma classe, oferecendo assim a possibilidade de indicar o quanto os dados apresentados se encaixam em determinada classe. O gráfico da equação (2) pode ser observado na Figura 7.

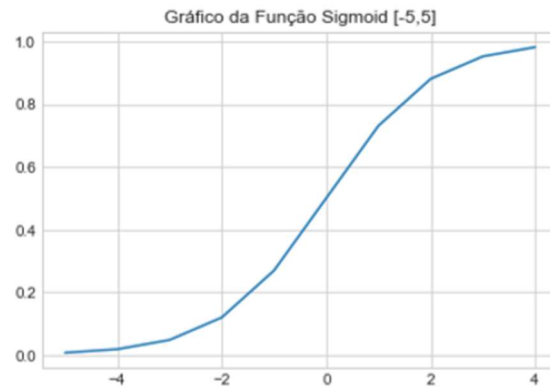


Figura 7- Gráfico da função sigmoid

Fonte: (LUDWIG & al., 2007)

$$\varphi(\vartheta) = \frac{1}{1 + e^{-\vartheta}} \quad (2)$$

### Hyperbolic Tanget (Tangente Hiperbólica)

Existem problemas em que há a necessidade de a RNA retornar valores negativos, para esses casos existe a função de ativação *hyperbolic tanget* (tangente hiperbólica) em que dado um valor x o seu retorno terá valores entre -1 e 1.

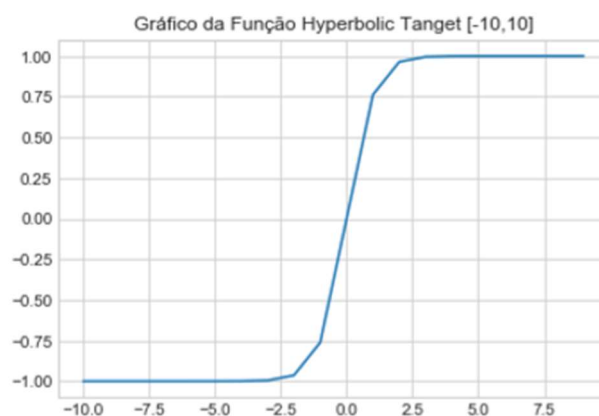


Figura 8 - Gráfico da função tangente hiperbólica

Fonte: (LUDWIG & al., 2007)

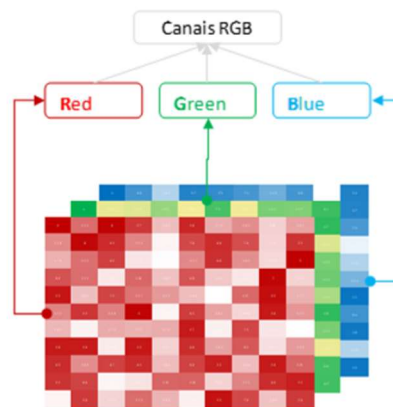
$$\varphi(\vartheta) = \frac{e^{\vartheta} - e^{-\vartheta}}{e^{\vartheta} + e^{-\vartheta}} \quad (3)$$

## PROCESSAMENTO DE IMAGENS PARA IA

O Processamento de imagens voltados ao uso em redes neurais convolucionais tem por objetivo extrair as principais características da imagem e reduzir o tamanho da mesma para que o processamento seja otimizado e mais eficiente em relação a outras técnicas de visão computacional.

As técnicas de convolução são realizadas aplicando matrizes convolucionais que realizam operações matemáticas na imagem que constitui da multiplicação da matriz da imagem pela matriz de convolução, extraíndo assim diferentes características relevantes da imagem.

Uma imagem internamente é formada por uma ou três matrizes que possui em seu interior valores de 0 a 255, a variação desse valor indica a intensidade do pixel no ponto (x,y). Uma imagem colorida é composta por 3 matrizes conforme Figura 9 que são chamados canais RGB (*Red, Green e Blue*), já uma imagem em escala de cinza possui apenas uma matriz.



*Figura 9 - Representação de canais RGB*

*Fonte: (SILVA & al., 2019)*

Como forma de reduzir a dimensão da imagem os algoritmos de redes neurais convolucionais aplicam camadas de pooling, um dos modelos utilizados é o max-pooling que consiste em deslizar sobre a matriz da imagem em uma janela 2 x 2 por exemplo, sobre os pixels, pegando o valor máximo dessa janela para construir uma imagem menor conforme Figura 10.



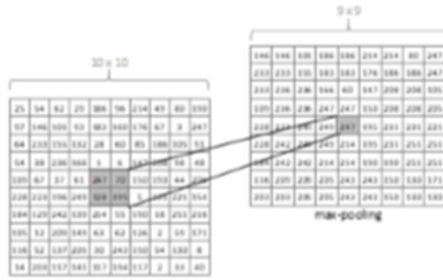


Figura 10 - Exemplo max-pooling

Fonte: (GRANATYR, 2020), adaptado pelo autor

As imagens passam por processamento sequencial de extração de características e pooling dessas características, ao final do processamento a imagem é convertida em *array*, que são de fato os valores que servirão de entradas para a rede neural como demonstrado na Figura 11.

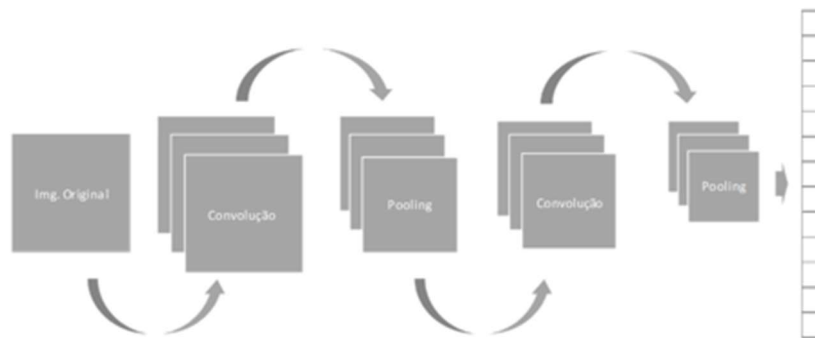


Figura 11 - Sequência de processamento

Fonte: (GÉRON, 2019), adaptado pelo autor

O passo final do processamento consiste na modificação da estrutura das imagens transformando-as do modelo matriz para o formato de array. A partir do ponto em que a imagem foi transformada em array a estrutura da rede é semelhante a um modelo que fosse construído para, por exemplo, processar um arquivo CSV (*Comma Separated Values* - Valores Separados Por Vírgula), onde cada linha seria o array de entrada da rede, essa estrutura pode ser observada na Figura 12.

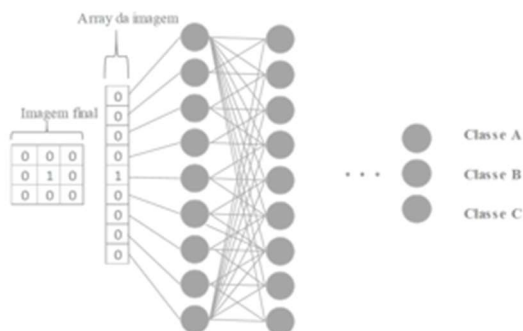


Figura 12 - Conversão da imagem para entradas de um RNA

Fonte: (GÉRON, 2019), adaptado pelo autor

## AGRICULTURA

Nas últimas décadas o setor agrícola brasileiro passou a cumprir papel fundamental no Produto Interno Bruto (PIB), com a avançada desindustrialização a partir dos anos 80 a indústria brasileira perdeu protagonismo e de acordo com (WENTZEL, 2016), em 2014 representava 10,9% do PIB enquanto que em 1970 essa participação era de 27,9%.

Em contrapartida os últimos 40 anos da agricultura brasileira foram de pesquisa e desenvolvimento o que permitiu que o país se tornasse o maior produtor de alimentos, sendo capaz de suprir a demanda interna dos principais produtos agrícolas consumidos no país e gerando excedentes que são exportados para diversos países (CNA, 2019).

No ano de 2018 o setor agro movimentou em bens e serviços R\$ 1,44 Trilhões, sendo 74% referente as demandas da agricultura e os 26% restantes vinculados a setores de pecuária.



Figura 13 - Produção agro 2018

Fonte: (CNA, 2019)

## Importância da Agricultura Brasileira no Mundo

A agricultura brasileira não é importante apenas internamente, investir em tecnologias que auxiliem o produtor rural garante ao país maior destaque no mundo como um fornecedor de alimentos com capacidade produtiva imensa já que devido as dimensões do país e clima diverso ao longo do território, permite que cada região produza alimentos que melhor se adequam ao clima desse ambiente.

A China com a sua população de 1,4 bilhões é um grande parceiro do Brasil na compra de alimentos, mas além da China diversos outros países consomem os produtos agrícolas brasileiro, ou os compram para alimentar animais com rações provenientes dos grãos produzidos no Brasil.

A Figura 14 demonstra o potencial do setor agro do país com destaque para a agricultura que está entre 1º e 4º produtores mundiais com os produtos agro mais essenciais para alimentação da humanidade, seja como produto final ou produto que servirá de alimentos para animais que posteriormente serão consumidos pela população.

Apesar de toda a capacidade produtiva do país os pequenos produtores passam por diversas dificuldades seja de financiamento da produção seja de acesso às tecnologias que auxiliem em maior produtividade. Para isso essa pesquisa visa atingir esse público com uma ferramenta de identificação de doenças podendo assim impactar positivamente no meio rural através da identificação eficiente de pragas.

O pequeno produtor tem papel fundamental na produção regional de alimentos e geralmente esses produtores buscam produzir seus produtos com baixo impacto no solo evitando o uso de pesticidas que contaminam tanto os trabalhadores como os alimentos e o solo.

Principais Produtos	Produção	Exportação	Part. Com. Inter. (Exp.)
Açúcar	2º	1º	34%
Café	1º	1º	26%
Suco de Laranja	1º	1º	78%
Soja em Grãos	2º	1º	52%
Carne de Frango	1º	1º	32%
Carne Bovina	2º	1º	21%
Milho	3º	3º	17%
Óleo de Soja	4º	2º	12%
Farelo de Soja	4º	2º	23%
Algodão	4º	2º	15%
Carne Suína	4º	2º	8%

Figura 14 - Participação no comércio internacional

Fonte: (CNA, 2019)

## Tecnologias na Agricultura

O setor agrícola pode se beneficiar de tecnologias de ponta de diversas formas. A Internet das Coisas (IoT- Internet of Things), pode ser aliado do agricultor contribuindo para uma agricultura mais inteligente e produtiva.

De acordo com (SILVA & MUXITO, 2018), hardwares de baixo custo como Arduino podem ser utilizados para automação em áreas rurais aplicados em tarefas repetitivas, essenciais e críticas como é o caso dos processos de irrigação, que pode por meio de sensores de humidade de solo tornar o processo de irrigação automatizado.

Além dos hardwares que podem ser ferramentas na agricultura, softwares podem auxiliar o produtor nos diversos pontos da produção contribuindo positivamente para organização e controle dos processos produtivos ou administrativos.

## **MATERIAIS E MÉTODOS**

### **Metodologia**

A pesquisa aqui apresentada se utiliza como principal modelo de pesquisa a pesquisa bibliográfica no qual são pontos de apoio a esse trabalho livros, teses, dissertações, reportagens e conteúdos produzidos por centros de pesquisas e desenvolvimento (GIL, 2002).

Como forma de demonstrar os conceitos abordados essa pesquisa se utiliza do modelo de pesquisa experimental com o objetivo de demonstrar a eficiência ou não dos modelos de IA voltados a visão computacional, com o objetivo de validar a eficiência e o nível de generalização de uma RNA convolucional. A pesquisa foi organizada da seguinte forma:

1. Estudo bibliográfico sobre funcionamento das redes neurais;
2. Estudo sobre os modelos de RNA's mais eficientes voltados a visão computacional;
3. Estudo sobre qual a linguagem mais adequada à elaboração do estudo de caso;
4. Campos em que modelos de RNA's poderiam ser implementadas;
5. Como a tecnologia pode auxiliar no campo de diferentes formas;
6. Obtenção de dados para testes iniciais;
7. Implementação de uma RNA para fazer os primeiros testes;
8. Análise sobre a capacidade de processamento do equipamento disponível para processamento em massa;
9. Definição da estrutura da RNA;
10. Definição da linguagem a ser utilizada na construção da rede neural;
11. Definição da arquitetura para o desenvolvimento do aplicativo mobile.

### **Ferramentas e Linguagens de Programação**

Com o objetivo de implementar a técnicas estudadas nesse trabalho algumas ferramentas e linguagens foram utilizadas. Afim de facilitar as configurações e bibliotecas necessárias

ao desenvolvimento do estudo de caso, o software Anaconda foi escolhido já que ao instalar essa ferramenta a mesma já traz consigo a linguagem Python e diversas bibliotecas de aprendizado de máquina e processamento de dados.

Esse trabalho se utiliza de bibliotecas disponibilizadas gratuitamente para a construção da rede neural, cálculos matemáticos e processamento de imagens conforme descrito a seguir:

- Numpy: biblioteca Python destinada a cálculos matemáticos;
- Matplotlib: biblioteca destinada a plotagem de gráficos ou imagens;
- TensorFlow: biblioteca disponibilizada de forma gratuita pela Google para projetos de IA;
- Keras: biblioteca que encapsula o TensorFlow como o objetivo de tornar a criação de modelos mais fácil.

Para a implementação do dispositivo móvel foi utilizado o framework Flutter que utiliza a linguagem Dart para a programação de ações dentro do aplicativo como também para a criação de componentes de interface. A escolha dessa ferramenta se deu pelo fato de possibilitar que uma aplicação criada nessa ferramenta seja compilada tanto para Android, IOS e web.

A digitação dos códigos foi feita através das IDE's Android Studio e VSCode para aplicação móvel e para os códigos Python foi utilizado a ferramenta Spyder que já é instalada quando o software Anaconda é instalado e em alguns momentos foi utilizado o VSCode.

## **Base de Dados**

A base de imagens utilizadas nessa pesquisa é uma base real obtida por meio da EMBRAPA. Conforme (BARBEDO & al., 2016), essa base tem como objetivo ser referência no desenvolvimento de métodos de detecção e reconhecimento de doenças em plantas.

De acordo com a EMBRAPA, a base pode ser considerada confiável já que todos os dados foram catalogados e etiquetados por profissionais experientes. A base de dados está separada em diversos arquivos *.zip* nomeados com as classes que pertencem.

Todas as bases disponíveis no site foram baixadas por meio de um *crawler* (robô) desenvolvido em Python afim de automatizar o download e descompactação das imagens.

Apesar de ter sido baixado todas as imagens as mesmas passaram por um processo de seleção e que apenas as imagens já cortadas foram utilizadas, outro critério aplicado foi que cada classe para fazer parte do treinamento contasse com no mínimo 100 amostras, já que na hora do treinamento seriam separados 70% para treino e 30% para validação da rede.

A base final foi reorganizada para que todas as imagens que fossem referentes a uma doença específica fossem agrupadas, de modo que se uma laranjeira e um limoeiro tenham a doença fumagina ambas as imagens farão parte da classe fumagina. A base final após a aplicação dos critérios mencionados pode ser observada na Tabela 1.

SEQ	CLASSE	QT_TREINO	QT_TESTE	% TESTE
1	ACARO	91	39	30,00%
2	ALGA	171	73	29,92%
3	ANTRACNOSE	592	253	29,94%
4	BACTERIOSE	455	195	30,00%
5	BRUSONE	173	74	29,96%
6	CANCRO	159	68	29,96%
7	CANCRO_BACTERIANO	465	199	29,97%
8	CLOROSE	216	92	29,87%
9	COCHONILHA	409	174	29,85%
10	CRESTAMENTO_BACTERIANO	2654	1137	29,99%
11	FERRUGEM	1586	679	29,98%
12	FERRUGEM_BRANCA	623	266	29,92%
13	FERRUGEM_POLISSORA	2134	914	29,99%
14	FITOTOXIDEZ	658	281	29,93%
15	FITO_DE_COBRE	1082	463	29,97%
16	FOLHA_CARIJO	218	93	29,90%
17	FUMAGINA	104	44	29,73%
18	LAGARTA_ENROLADEIRA	70	30	30,00%
19	LIXA_GRANDE	427	182	29,89%
20	LIXA_PEQUENA	137	58	29,74%
21	MANCHA_ALVO	677	289	29,92%
22	MANCHA_ANELAR	1160	496	29,95%
23	MANCHA_ANGULAR	933	399	29,95%
24	MANCHA_AUREOLADA	805	344	29,94%
25	MANCHA_BACTERIANA	119	50	29,59%
26	MANCHA_BIPOLARIS	2639	1131	30,00%
27	MANCHA_BRANCA	546	234	30,00%
28	MANCHA_CYLINDROCLADIUM	70	30	30,00%
29	MANCHA_DE_ALTERNARIA	94	39	29,32%
30	MANCHA_DE_MIROTECIO	110	46	29,49%
31	MANCHA_GRAXA	190	81	29,89%
32	MANCHA_MANTEGOSA	123	52	29,71%
33	MANCHA_PHYSODERMA	750	321	29,97%
34	MANCHA_TURCICUM	110	46	29,49%
35	MELA	117	49	29,52%
36	MILDIO	1615	691	29,97%
37	MOFO_PRETO	1480	634	29,99%
38	MOSAICO	308	132	30,00%
39	OIDIO	904	387	29,98%
40	PODRIDAO_VERMELHA	73	31	29,81%
41	RAMULARIA	1198	513	29,98%
42	SEPTORIA	874	374	29,97%
43	VERRUGOSE	108	45	29,41%

*Tabela 1 - Base final para treinamento*

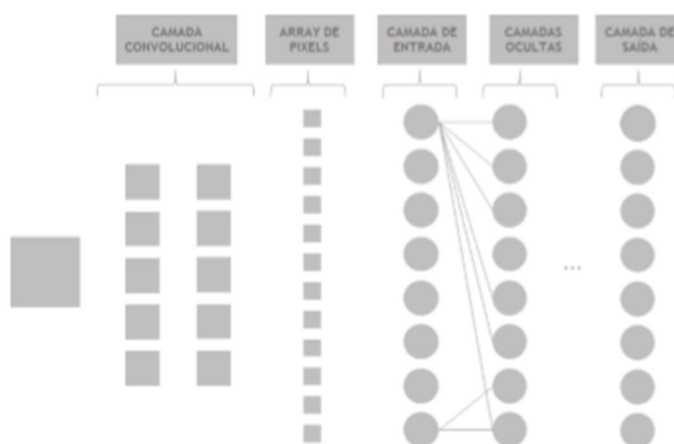
*Fonte: próprio*

## RESULTADOS

O processo de treinamento final foi realizado utilizando uma máquina na nuvem por meio da plataforma Digital Ocean com Ubuntu 20.01 (LTS) x64 com 8 GB de memória e 160 GB de disco. A utilização de uma máquina externa foi realizada pelo fato de o processamento de redes neurais serem demorados e qualquer problema na máquina local poderia interromper o processo de treinamento sendo obrigado a reiniciar novamente.

A rede neural foi configurada para passar por vários ciclos de treinamento afim de analisar a evolução da mesma conforme a quantidade de ciclos fossem aumentando. A cada ciclo de treinamento foi salvo a estrutura da rede, os pesos, os índices das classes e a imagem do gráfico de acuracidade da rede.

Como pode ser observado na Figura 15 as imagens antes de serem enviadas a camada de entrada da rede neural passa por um pré-processamento na camada convolucional que é o meio pelo qual as principais características são extraídas, nesse processamento as imagens são redimensionadas para o tamanho 64x64 onde também é aplicada uma função de normalização dos pixels, ou seja, o valor de cada pixel é dividido por 255 convertendo assim os valores para o intervalo entre 0 e 1.



*Figura 15 - Estrutura final da rede neural*

*Fonte: próprio*

A rede foi configurada para passar por vários ciclos de treinamento, onde a cada ciclo o número de épocas foi incrementado em 25 até o que o último ciclo fosse composto por 200 épocas, o processamento foi configurado dessa forma para verificar a evolução do sistema no quesito acuracidade conforme os números de épocas fossem sendo incrementado. O desempenho da rede pode ser observado na Figura 16 no último ciclo de treinamento composto por 200 épocas.

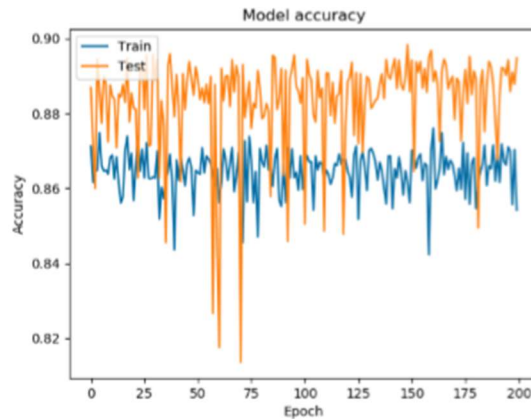


Figura 16 - Gráfico de acuracidade da rede

Fonte: próprio

Para a utilização do modelo treinado foi criado um *web service* utilizando a biblioteca Flask que permite a construção de aplicações web de forma simples e o fácil roteamento dos serviços disponibilizados configurando o serviço para receber uma imagem e submeter ao modelo carregado por meio dos arquivos que foram salvos durante o processo de treinamento da rede. O retorno da classificação foi definido para ser no formato Javascript Object Notation (JSON) como pode ser observado na Figura 17.

```

1 {
2   "msg": "Success",
3   "items": [{
4     "classe": "MANCHA_TURCICUM",
5     "valor": 97.82148599624634
6   }, {
7     "classe": "MANCHA_PHYSODERMA",
8     "valor": 0.752696767449379
9   }, {
10    "classe": "COCHONILHA",
11    "valor": 0.6280194502323866
12  },
13  ...]
14  {
15    "classe": "MANCHA_BIPOLARIS",
16    "valor": 0.3366563469171524
17  },
18  }

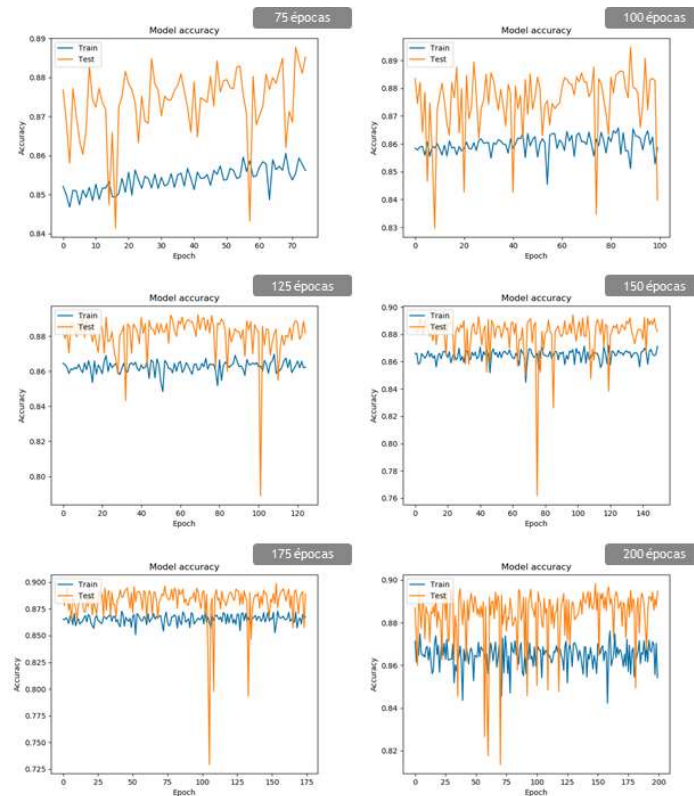
```

Figura 17 - Retorno de classificação de imagem (JSON)

Fonte: próprio

A figura demonstra a acuracidade do sistema utilizando os modelos salvos dos ciclos de treinamento com número de épocas acima de 75, como é possível observar a partir desse ponto a rede já não tem ganhos de acuracidade perceptíveis e que a partir desse ponto na maior parte do tempo a rede manteve o nível de acertos acima de 85% e que os treinamentos com épocas acima de 150 épocas obtiveram ganhos que não foram significativos para tornar a rede com maior acuracidade.





*Figura 18 - Gráfico de acuracidade por ciclos*

*Fonte: próprio*

O processo de validação final se deu utilizando-se das imagens reservadas ao teste sendo 30% do total de cada classe utilizando o modelo já treinado com as 200 épocas de aprendizado e verificando a acuracidade da rede neural para cada uma das classes.

Como pode ser observado os resultados finais apresentaram acuracidade média de 85,22% com alguns casos apresentando desempenho de classificação muito baixo. Esse desempenho baixo pode ser melhorado incrementando um volume maior de imagens com melhor qualidade de visualização, porém de forma geral a rede se mostrou bastante confiável com diversas classificações obtendo valores de acerto acima de 90%.

PERCENTUAL DE ACERTOS POR CLASSE

CLASSE	QTD_IMG	ERROS	ACERTOS	PERC. ACERTOS
ACARO	39	5	34	87,18%
ALGA	73	6	67	91,78%
ANTRACNOSE	253	33	220	86,96%
BACTERIOSE	195	22	173	88,72%
BRUSONE	74	26	48	64,86%
CANCRO	68	2	66	97,06%
CANCRO_BACTERIANO	199	27	172	86,43%
CLOROSE	92	8	84	91,30%
COCHONILHA	174	7	167	95,98%
CRESTAMENTO_BACTERIANO	1137	115	1022	89,89%
FERRUGEM	679	40	639	94,11%
FERRUGEM_BRANCA	266	25	241	90,60%
FERRUGEM_POLISSORA	914	32	882	96,50%
FITOTOXIDIZ	281	3	278	98,93%
FITO_DE_COBRE	463	50	413	89,20%
FOLHA_CARIJO	93	13	80	86,02%
FUMAGINA	44	1	43	97,73%
LAGARTA_ENROLADEIRA	30	10	20	66,67%
LIXA_GRANDE	182	16	166	91,21%
LIXA_PEQUENA	58	8	50	86,21%
MANCHA_ALVO	289	122	167	57,79%
MANCHA_ANELAR	496	16	480	96,77%
MANCHA_ANGULAR	399	74	325	81,45%
MANCHA_AUREOLADA	344	23	321	93,31%
MANCHA_BACTERIANA	50	7	43	86,00%
MANCHA_BIPOLARIS	1131	118	1013	89,57%
MANCHA_BRANCA	234	39	195	83,33%
MANCHA_CYLINDROCLADIUM	30	10	20	66,67%
MANCHA_DE_ALTERNARIA	39	17	22	56,41%
MANCHA_DE_MIROTECIO	46	11	35	76,09%
MANCHA_GRAXA	81	22	59	72,84%
MANCHA_MANTEGOSA	52	3	49	94,23%
MANCHA_PHYSODERMA	321	23	298	92,83%
MANCHA_TURCICUM	46	23	23	50,00%
MELA	49	13	36	73,47%
MILDIO	691	35	656	94,93%
MOFO_PRETO	634	47	587	92,59%
MOSAICO	132	15	117	88,64%
OIDIO	387	21	366	94,57%
PODRIDAO_VERMELHA	31	2	29	93,55%
RAMULARIA	513	19	494	96,30%
SEPTORIA	374	116	258	68,98%
VERRUGOSE	45	6	39	86,67%

Tabela 2 - Percentual de acertos por classe

Fonte: próprio

Os sistemas inteligentes são sistemas que nem sempre oferecem o resultado correto, isso é muito bom já que o objetivo não é que todos os resultados apresentados como resposta estejam corretos, se isso ocorrer a rede pode não estar generalizando e sim ter “decorado” as respostas para um conjunto de dados e se em algum momento os dados saírem desse padrão a rede não mais funcionará.

A Figura 19 demonstra a classificação realizada por meio do aplicativo, indicando que a possibilidade de a imagem apresentada estar com problemas de cochonilha é de 95,8%.

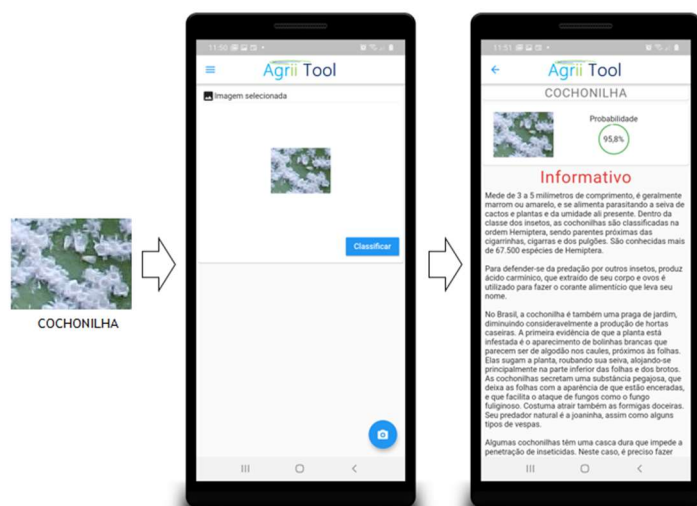


Figura 19 - Classe classificada corretamente

Fonte: próprio

Como já mencionado, um sistema inteligente nem sempre oferecerá resultados corretos, para isso é preciso definir um grau de aceitabilidade quando uma resposta pelo sistema é retornada a Figura 20 demonstra isso, como pode ser observado uma imagem com um problema ao ser submetida ao sistema retornou que com 37,44% de probabilidade o problema se trata de mancha turcicum, ou seja, além da classificação errada a probabilidade de pertencer a essa classe é muito baixa, devendo não ser utilizada a resposta da rede.

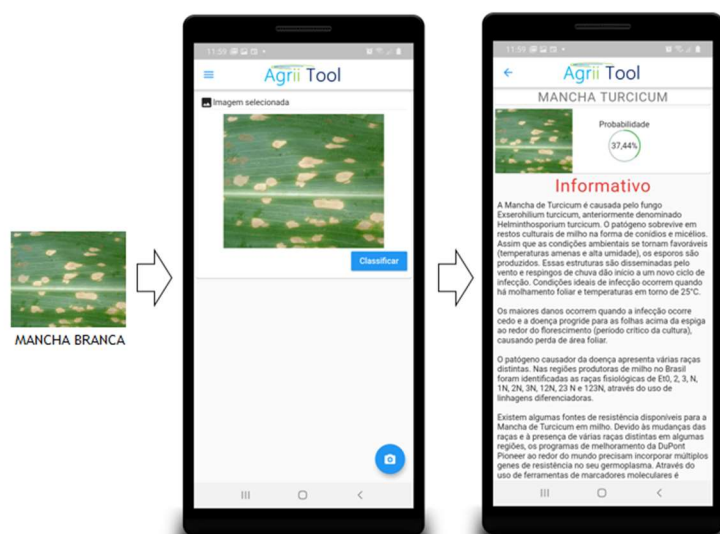


Figura 20 - Mancha branca classificada errada

Fonte: próprio

## CONSIDERAÇÕES FINAIS

O objetivo dessa pesquisa foi o de se utilizar de tecnologias disponíveis gratuitamente para assim construir um sistema de identificação de doenças em plantas com foco em pequenos agricultores. A base de dados utilizada demonstrou bom desempenho nos testes

finais apesar de algumas classes terem obtido desempenho abaixo do esperado, em teoria o baixo desempenho pode ser atribuído pela baixa quantidade de amostras dessas classes como também a qualidade baixa dessas imagens.

De modo geral evidenciou-se que as redes neurais convolucionais são tecnologias sofisticadas e que são capazes de auxiliar pequenos agricultores oferecendo sistemas inteligentes baseados nessa tecnologia. Outro ponto a destacar são as ferramentas escolhidas inicialmente que no decorrer dos testes se mostraram a escolha correta, tanto para o desenvolvimento dos sistemas na linguagem Python bem como o sistema móvel onde foi escolhido o framework Flutter.

Diante dos resultados apresentados pode-se considerar que sistemas baseados em visão computacional utilizando algoritmos de redes neurais convolucionais são eficientes e que para a melhor acuracidade são necessárias grandes quantidades de imagens que representem o problema a ser resolvido.

A pesquisa também demonstrou que algumas melhorias podem ser implementadas como por exemplo: banco de imagens com maior representatividade das doenças; implementar um número maior de camadas ocultas na rede; fazer testes com imagens coloridas e em escala de cinza bem como utilizar outras bibliotecas com o mesmo foco como por exemplo a PyTorch comparando assim o desempenho de ambas.

Diante das possibilidades apresentadas pelas redes neurais convolucionais permite apontar como trabalhos futuros a construção de sistemas inteligentes com capacidade de realizar a colheita automatizada de frutos em que esse processo é realizado manualmente devido ao amadurecimento em tempos diferentes, onde é colhido apenas os frutos que esteja com padrões de qualidades pré-definidos. Outra pesquisa pode focar na implementação de esteiras inteligentes capazes de realizar a separação de frutos baseada no tamanho, coloração ou manchas em sua superfície por meio de visão computacional.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARBEDO, J. G., & al., e. (2016). *Identifying multiple plant diseases using digital image processing. Biosystems Engineering*. Acesso em 15 de Dezembro de 2019, disponível em EMBRAPA: <https://www.digipathos-rep.cnptia.embrapa.br/jspui/>

BRAGA, A. P., & al., E. (2014). *Redes Neurais Artificiais: Teoria e Prática* (2ª ed.). Rio de Janeiro: LTC.

CNA. (2019). *Panorama do Agro*. Acesso em 04 de Abril de 2020, disponível em <https://www.cnabrazil.org.br>: <https://www.cnabrazil.org.br/cna/panorama-do-agro>

GÉRON, A. (2019). *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Rio de Janeiro: Alta Books.

GIL, A. C. (2002). *Como Elaborar Projetos de Pesquisa* (4ª ed.). São Paulo: Novatec.

GRANATYR, J. (20 de Março de 2020). *IAExpert*. Fonte: [iaexpert.com.br](https://iaexpert.com.br): <https://iaexpert.com.br/index.php/courses/redes-neurais-artificiais-em-python/>

- HAYKIN, S. (2001). *Redes Neurais: Princípios e Práticas*. Porto Alegre: Bookman.
- LUDWIG, J. O., & al., E. (2007). *Redes Neurais: Fundamento e Aplicações com Programas em C*. (P. A. Marques, Ed.) Rio de Janeiro: Ciência Moderna.
- ROCHA, B. (2016). *SUPERA*. Acesso em 23 de Fevereiro de 2020, disponível em SUPERA: <https://metodosupera.com.br/neuronios-glossario-do-cerebro/>
- SANTOS, A. E. (2017). Uma Abordagem de Classificação de Imagens Dermatoscópicas Utilizando Aprendizado Profundo com Redes Neurais Convolucionais. *XXXVII Congresso da Sociedade Brasileira de Computação*, pp. 2010-2019.
- SILVA, I. N., & al., E. (2010). *Redes Neurais Para Ciências Aplicadas: Curso Prático*. São Paulo: Artliber.
- SILVA, M. S., & MUXITO, E. M. (Outubro de 2018). Agricultura Inteligente: proposta de Automação de Pivôs e Canais de Irrigação com Prototipação por Arduino e Webservice. *V CIMATech*.
- SILVA, P. R., & al., e. (2019). Uma Abordagem de Classificação de Imagens Dermatoscópicas Utilizando Aprendizado Profundo com Redes Neurais Convolucionais. *ARITHMOS*, pp. 22-27.
- WENTZEL, M. (21 de Setembro de 2016). *Brasil passa por desindustrialização precoce, aponta pesquisa da ONU*. Acesso em 18 de 06 de 2020, disponível em BBC Brasil: <https://www.bbc.com/portuguese/brasil-37432485>