

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Gesiane Morales Pereira

CLASSIFICAÇÃO DE DOENÇAS NA FOLHA DO ARROZ ATRAVÉS DE  
PROCESSAMENTO DE IMAGENS E APRENDIZADO DE MÁQUINA

Florianópolis

2022

Gesiane Morales Pereira

**CLASSIFICAÇÃO DE DOENÇAS NA FOLHA DO ARROZ ATRAVÉS DE  
PROCESSAMENTO DE IMAGENS E APRENDIZADO DE MÁQUINA**

Trabalho de Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Joceli Mayer, Ph.D.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Pereira, Gesiane Morales

Classificação de Doenças na Folha do Arroz Através de  
Processamento de Imagens e Aprendizado de Máquina /  
Gesiane Morales Pereira ; orientador, Joceli Mayer, 2022.  
46 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Visão Computacional. 3.  
Processamento de Imagens. 4. Aprendizado de Máquina. 5.  
Rizicultura. I. Mayer, Joceli. II. Universidade Federal de  
Santa Catarina. Graduação em Engenharia Elétrica. III.  
Título.

Gesiane Moraes Pereira

**Classificação de Doenças na Folha do Arroz Através de Processamento de Imagens e Aprendizado de Máquina**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 14 de dezembro de 2022.



Documento assinado digitalmente

**Miguel Moreto**

Data: 14/12/2022 16:43:47-0300

CPF: \*\*\*.850.100-\*\*

Verifique as assinaturas em <https://v.ufsc.br>

---

**Prof. Miguel Moreto, Dr.**

Coordenador do Curso de Graduação em Engenharia Elétrica

**Banca Examinadora:**



Documento assinado digitalmente

**Joceli Mayer**

Data: 14/12/2022 16:40:03-0300

CPF: \*\*\*.833.519-\*\*

Verifique as assinaturas em <https://v.ufsc.br>

---

**Prof. Joceli Mayer, Ph.D.**

Orientador

Universidade Federal de Santa Catarina



Documento assinado digitalmente

**Bruno Catarino Bispo**

Data: 14/12/2022 19:53:29-0300

CPF: \*\*\*.973.077-\*\*

Verifique as assinaturas em <https://v.ufsc.br>

---

**Prof. Bruno Catarino Bispo, Dr.**

Universidade Federal de Santa Catarina



Documento assinado digitalmente

**Jefferson Luiz Brum Marques**

Data: 14/12/2022 18:53:03-0300

CPF: \*\*\*.624.060-\*\*

Verifique as assinaturas em <https://v.ufsc.br>

---

**Prof. Jefferson Luiz Brum Marques, Ph.D.**

Universidade Federal de Santa Catarina

Este trabalho é dedicado à minha família e amigos.

## **AGRADECIMENTOS**

Agradeço ao meu orientador, o professor Joceli Mayer, que sempre se colocou à disposição e que me deu apoio nos meus estudos desta área tão vasta que é processamento de imagens.

Ao meu pai, que possibilitou que eu me dedicasse integralmente aos estudos e que me incentivou a sempre buscar o conhecimento e enfrentar os desafios.

À minha irmã, que sempre esteve presente para me ouvir e me apoiar nos momentos mais difíceis da graduação e para comemorar cada pequena vitória.

Ao Gabriel, por estar ao meu lado e acreditar em mim.

À Pandora, pela companhia e carinho.

Aos meus amigos, por trazerem um pouco de leveza e esperança à essa longa jornada de estudos e aprendizado.



## RESUMO

O arroz é um dos alimentos mais consumidos no Brasil e no mundo. Doenças na planta podem reduzir drasticamente a produção do grão, sendo que, a Brusone e a Mancha-parda, são as principais doenças que afetam a rizicultura. Para agilizar a detecção de doenças na lavoura pode ser realizada a captura de imagens para posterior processamento e análise. Este trabalho busca propor um modelo de visão computacional que classifique as imagens das folhas do arroz entre as categorias doente com Brusone, doente com Mancha-parda e saudável. O modelo proposto se baseia no descritor de textura e padrões *Local Binary Pattern* e no algoritmo de aprendizado de máquina *Support Vector Machine*. A otimização de parâmetros através da busca utilizando o algoritmo de *Grid Search* apontou o melhor tipo de kernel e valores para os parâmetros C e *Gamma* do algoritmo de classificação para o problema em estudo. A validação cruzada forneceu informações para a avaliação de não existência de *overfitting*. O modelo obteve uma acurácia de 81,81%.

**Palavras-chave:** Visão Computacional, Processamento de Imagens, Aprendizado de Máquina, Rizicultura..

## ABSTRACT

Rice is one of the main foods consumed in Brazil and in the world. Diseases can drastically reduce the grain production, being Leaf Blast and Brown Spot the main ones that affect rice farming. To speed up the disease detection, images of the rice field can be obtained to later processing. The present study aims to propose a Computer Vision model that classifies rice images between the categories Leaf Blast, Brown Spot and healthy. The proposed model is based on the pattern and texture descriptor Local Binary Pattern and on the machine learning algorithm Support Vector Machine. The Grid Search algorithm pointed the best parameters for C, Gamma and the type of kernel for the problem in study. The cross validation algorithm offered informations to evaluate the non existence of overfitting. The model obtained an 81.81% accuracy.

**Keywords:** Computer Vision. Image Processing. Machine Learning. Rice Farming.

## LISTA DE FIGURAS

Figura 1 – Folhas da planta do arroz afetadas pela doença Brusone.....	16
Figura 2 - Folhas da planta do arroz afetadas pela doença Mancha-parda.....	17
Figura 3 – Fluxo das etapas de um sistema de Visão Computacional. ....	19
Figura 4 – Exemplos de possíveis vizinhanças a serem consideradas e funcionamento do LBP para uma delas. ....	22
Figura 5 – Histograma resultante da aplicação do LBP a uma imagem.....	23
Figura 6 – Funcionamento do SVM para a separação de duas classes. ....	25
Figura 7 - Efeito do kernel na separação do espaço de atributos entre as diferentes classes. ....	27
Figura 8 - Efeito do parâmetro <i>Gamma</i> na fronteira de separação, mantendo fixos os demais parâmetros. ....	28
Figura 9 - Efeito do parâmetro <i>C</i> na fronteira de separação. ....	28
Figura 10 - Exemplo de funcionamento do <i>K-Fold</i> e validação cruzada. ....	30
Figura 11 - Resultados de acurácia obtidos por Pothen e Pai (2020). ....	31
Figura 12 - Imagens classificadas corretamente.....	38
Figura 13 - Imagens classificadas incorretamente. ....	39
Figura 14 - Matriz de confusão. ....	40

## LISTA DE TABELAS

Tabela 1 – Resultados de acurácia de diferentes combinações de parâmetros do SVM. .....	37
Tabela 2 – Resultados individuais da validação cruzada. ....	41

## **LISTA DE ABREVIATURAS E SIGLAS**

SVM Support Vector Machine

LBP Local Binary Pattern

HOG Histogram of Oriented Gradients

KNN K-Nearest Neighbors

CNN Convolutional Neural Network

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
1.1	OBJETIVO GERAL.....	18
1.2	OBJETIVOS ESPECÍFICOS .....	18
1.3	ESTRUTURA DO DOCUMENTO .....	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>19</b>
2.1	VISÃO COMPUTACIONAL .....	19
<b>2.1.1</b>	<b>LOCAL BINARY PATTERN .....</b>	<b>21</b>
2.2	APRENDIZADO DE MÁQUINA .....	23
<b>2.2.1</b>	<b>SUPPORT VECTOR MACHINE.....</b>	<b>24</b>
2.2.1.1	<i>KERNEL</i> .....	25
2.2.1.1.1	PARÂMETRO GAMMA.....	27
2.2.1.2	<i>HIPERPARÂMETRO C</i> .....	28
<b>2.2.2</b>	<b>GRID SEARCH .....</b>	<b>29</b>
<b>2.2.3</b>	<b>VALIDAÇÃO CRUZADA.....</b>	<b>29</b>
2.3	TRABALHOS RELACIONADOS .....	30
<b>3</b>	<b>MATERIAIS E MÉTODO .....</b>	<b>33</b>
3.1	MATERIAIS.....	33
3.2	MÉTODO .....	33
<b>3.2.1</b>	<b>GRID SEARCH .....</b>	<b>34</b>
<b>3.2.2</b>	<b>ALGORITMO PRINCIPAL .....</b>	<b>34</b>
<b>3.2.3</b>	<b>VALIDAÇÃO CRUZADA.....</b>	<b>35</b>
<b>4</b>	<b>RESULTADOS .....</b>	<b>37</b>
4.1	RESULTADOS DA OTIMIZAÇÃO DE PARÂMETROS.....	37
4.2	RESULTADO DA CLASSIFICAÇÃO .....	37
4.3	RESULTADO DA VALIDAÇÃO CRUZADA.....	41
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>43</b>

<b>REFERÊNCIAS.....</b>	<b>44</b>
-------------------------	-----------

## 1 INTRODUÇÃO

O arroz é um dos alimentos mais importantes do mundo, sendo o segundo cereal mais cultivado (SARAVIA, 2005). Ele é componente básico da alimentação de bilhões de pessoas, principalmente nos países asiáticos. No Brasil, maior produtor e consumidor de arroz fora do continente asiático, ele compõe juntamente com o feijão a base das principais refeições diárias da população. “Em 2021, o consumo médio aparente de arroz beneficiado (58% grãos limpos e inteiros) foi de 29,6 kg/hab.” (SILVA, 2022).

As doenças do arroz podem afetar expressivamente a produção do grão, tanto em termos de quantidade como em qualidade. As principais doenças que afetam a rizicultura no Brasil e no Mundo são, nesta ordem de importância, a Brusone e a Mancha-parda (LORENSI *et al.*, 2021). Também vale mencionar outras doenças de importância significativa, como a Escaldadura e a Mancha-estreita (PRABHU *et al.*, 1995).

Brusone é uma doença causada pelo fungo *Pyricularia oryzae* e que pode provocar perdas de até 100% da produção (OLIVEIRA NETO, 2015). De acordo com Silva-Lobo e Filippi (2017), ela se inicia com uma aparência de pequenos pontos marrons e, à medida que aumentam, essas lesões passam a apresentar uma forma elíptica com o centro geralmente cinza e bordas marrons, às vezes circundada por um halo amarelado, como pode ser observado na Figura 1. Estima-se que a Brusone seja responsável por cerca de 30% das perdas de produção de arroz no mundo todo (SCHEUERMANN; NESI, 2021).

A Mancha-parda é causada pelo fungo *Drechslera oryzae* (PRABHU *et al.*, 1995). De acordo com Silva-Lobo e Filippi (2017), ela se manifesta sob a aparência de lesões circulares ou ovais, de coloração marrom, como se observa na Figura 2. Em sua versão mais severa pode ocasionar perdas de produtividade de até 52% (SCHEUERMANN; NESI, 2021).

Figura 1 – Folhas da planta do arroz afetadas pela doença Brusone.



Fonte: SAPUTRA (2022).

Figura 2 - Folhas da planta do arroz afetadas pela doença Mancha-parda.



Fonte: SAPUTRA (2022).

A aplicação de fungicidas é normalmente adotada como medida preventiva, porém, sendo identificada a doença, a aplicação o mais logo possível pode ajudar a reduzir a velocidade do avanço pela lavoura (FILIPPI *et al.*, 2015). Muitos fungicidas são indicados para apenas um tipo de doença, mas alguns deles são recomendados tanto contra Brusone quanto contra Mancha-parda, podendo ser ainda mais abrangentes (PRABHU *et al.*, 1995).

Identificar com precisão as regiões infectadas o mais cedo possível permite uma ação mais pontual, aplicando o fungicida apenas nas áreas afetadas e suas proximidades, resultando em uma economia de recursos e redução de custos e perdas. Neste sentido, poderia ser realizado o mapeamento e captura de imagens da lavoura através de um drone, as quais forneceriam dados importantes para a análise e identificação de áreas infectadas.

A aplicação de técnicas de processamento de imagens, visão computacional e aprendizado de máquina tem o potencial de auxiliar na detecção de doenças no cultivo do arroz. Com este intuito, foi proposta neste trabalho a utilização de um modelo baseado no descritor de

textura *Local Binary Pattern* (LBP) e no algoritmo de classificação *Support Vector Machine* (SVM).

### 1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é avaliar o uso de visão computacional e aprendizado de máquina para a classificação de imagens de folhas do arroz e identificação das doenças. Com base nos resultados dos trabalhos relacionados, foi estabelecido como objetivo a obtenção de um modelo com acurácia superior a 90%.

### 1.2 OBJETIVOS ESPECÍFICOS

- Propor um modelo de classificação de imagens da folha do arroz que diferencie as folhas com Brusone, folhas com Mancha-parda e folhas saudáveis;
- Verificar as limitações do modelo proposto.

### 1.3 ESTRUTURA DO DOCUMENTO

Este trabalho é estruturado da seguinte forma:

A seção seguinte se concentrará na fundamentação teórica das ferramentas e conceitos utilizados como base para a elaboração deste trabalho.

A seção 3 apresentará os materiais utilizados e descreverá em detalhes o método proposto.

A seção 4 apresentará os resultados obtidos e discorrerá acerca deles.

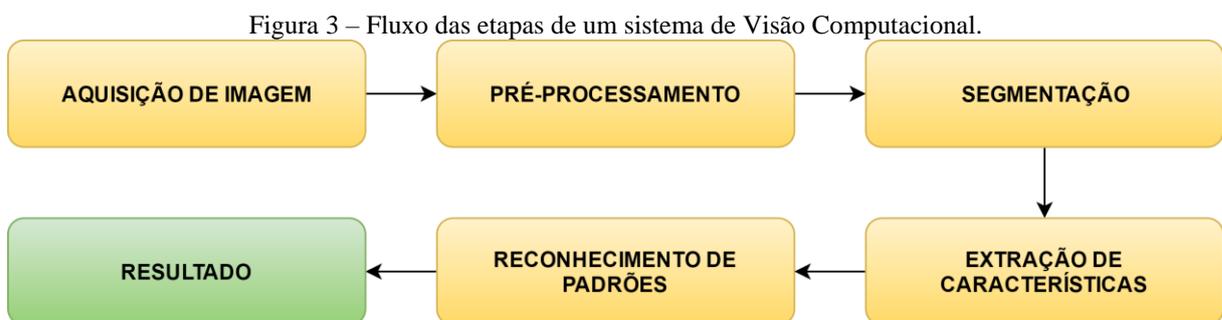
Por fim, a seção 5 consistirá em reflexões e conclusões sobre o que foi feito no trabalho e seus resultados.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão abordados alguns conceitos básicos relacionados às principais ferramentas utilizadas para a realização deste trabalho, bem como sua importância, funcionamento e aplicações. Ao fim, será feita uma revisão dos trabalhos relacionados, que têm uma proposta semelhante à deste.

### 2.1 VISÃO COMPUTACIONAL

De acordo com Ballard e Brown (1982 apud BARELLI, 2019, p. 1), “Visão Computacional pode ser definida como a ciência que estuda e desenvolve tecnologias que permitem que máquinas enxerguem e extraiam características do meio, através de imagens capturadas por diferentes tipos de sensores e dispositivos”. A visão computacional busca prover a sistemas eletrônicos a capacidade de enxergar o ambiente de maneira semelhante à dos humanos. Essa capacidade pode envolver tarefas como a aquisição das imagens, tratamento, segmentação das regiões de interesse, extração de suas características e realização do reconhecimento e interpretação. Uma representação do fluxo de sistemas de visão computacional pode ser observada na Figura 3.



Fonte: BARELLI (2019).

Na etapa de aquisição das imagens utiliza-se dispositivos ou sensores para capturar e digitalizar a imagem ou vídeo. Essas imagens podem ser em tons de cinza ou coloridas. No caso de imagens coloridas, elas são tipicamente representadas no espaço de cores RGB, onde a imagem é composta por três canais: vermelho (*Red*), verde (*Green*) e azul (*Blue*). Outro exemplo de espaço de representação de cores para imagens coloridas é o HSV, onde os três

canais são: matiz (*Hue*), saturação (*Saturation*) e o valor (*Value*) (BARELLI, 2019). Tendo as imagens, a próxima etapa é a de pré-processamento.

De acordo com Barelli (2019), na etapa de pré-processamento são realizadas operações com o objetivo de melhorar a qualidade da imagem e facilitar as etapas seguintes. Aqui são realizadas filtragens de ruídos, realce de bordas, melhoria de contraste, correção de distorções de perspectiva, operações morfológicas, entre outras.

Dentre os filtros contra ruídos mais populares estão: o filtro gaussiano, que apresenta bons resultados em imagens com ruído gaussiano (a variação aleatória do sinal segue a distribuição gaussiana), o filtro de mediana, adequado para ruídos do tipo "sal e pimenta", e o filtro bilateral, que se destaca pela sua capacidade de suavizar a imagem preservando detalhes de bordas e contornos (BARELLI, 2019).

Para o realce de bordas, se destacam: o operador de Sobel (ou filtro de Sobel), que realça linhas verticais e horizontais; o operador laplaciano, que geralmente produz bordas mais finas que o Sobel, e o filtro máscara de desaguçamento (ou filtro de realce), que é baseado na subtração da imagem original por uma versão suavizada dela (BARELLI, 2019).

Para melhoria de contraste, uma importante técnica é a equalização de histograma. Nela, para imagens em tons de cinza, é realizada uma melhor distribuição dos valores de intensidade de cada pixel, de modo que, histogramas com uma distribuição muito concentrada em determinada faixa de valores passem a apresentar uma distribuição mais espalhada. Já para imagens coloridas, essa equalização pode ser feita no canal V da imagem, que representa informações relacionadas à intensidade de luz no pixel (BARELLI, 2019).

Conforme exposto por Barelli (2019), os ajustes de perspectiva são úteis nos casos onde o posicionamento do dispositivo no momento da captura da imagem estava inclinado e causou distorção dos objetos na imagem. Para corrigir este problema a biblioteca OpenCV oferece como alternativa o uso das funções `getPerspectiveTransform` e `warpPerspective`.

Por fim, as operações morfológicas possibilitam a alteração do formato ou estrutura dos objetos na imagem. No caso da operação de erosão, esta consiste em, percorrendo a imagem com um elemento estruturante preestabelecido, "corroer" as bordas do objeto de modo que sua área fique menor do que originalmente. Já a operação de dilatação, funciona de maneira parecida, mas ao contrário da erosão, o objeto tem as suas bordas expandidas e sua área se torna maior (BARELLI, 2019).

Na etapa de segmentação, os objetos ou regiões de interesse são separados do resto da imagem para facilitar a etapa de extração de características. Aqui, de acordo com Barelli (2019),

pode ser empregada a segmentação por binarização, onde um limiar é utilizado na comparação com os pixels e a imagem resultante é composta apenas por pixels pretos e brancos. Também pode ser realizada a segmentação por cor, onde uma imagem colorida representada no espaço HSV tem seus pixels comparados com uma faixa de valores para os canais H, S e V. Se os valores do pixel se encontrarem dentro da faixa, será representado pela cor branca e os demais pixels serão representados na cor preta. Outra alternativa seria a segmentação por bordas, onde apenas os contornos do objeto são destacados do resto da imagem.

Tendo o objeto isolado do resto da imagem, se torna fácil a tarefa de extrair as suas características, que podem ser referentes por exemplo, a cor, dimensões e formato. Também podem ser calculados os momentos invariantes de Hu, que dentre várias utilidades, podem fornecer as informações necessárias para a determinação das coordenadas do centroide do objeto. Cabe aqui destacar o descritor de textura e padrões *Local Binary Pattern* (LBP), que será abordado de maneira mais detalhada no decorrer desta seção.

Por último, na etapa de reconhecimento de padrões, o objetivo é reconhecer o objeto e identificar a classe à qual ele pertence. Nessa etapa é comum a utilização de ferramentas de aprendizado de máquina, assunto que também será descrito de maneira detalhada mais adiante nesta seção.

A versatilidade da aplicação da visão computacional é muito ampla, podendo ser empregada na resolução de problemas das mais diversas áreas como saúde, segurança, indústria, agricultura e mobilidade. Na área da saúde os sistemas podem auxiliar os médicos na detecção de anomalias presentes em resultados de exames por imagem. Na segurança, as imagens de câmeras de monitoramento podem fornecer material para o reconhecimento facial de suspeitos. O reconhecimento facial também tem aplicação na autenticação de operações, tornando-as mais confiáveis. Na indústria, o controle de qualidade pode contar com a automatização da inspeção das peças e produtos para identificação de defeitos e não conformidades. Na agricultura, as máquinas ganham a capacidade de identificar frutos maduros para realizar a colheita, localizar ervas daninhas e, como explorado neste trabalho, realizar a detecção de doenças nas culturas de alimentos. Na mobilidade, se aplica ao sistema de navegação de carros autônomos e controle de semáforos para otimização do fluxo de veículos.

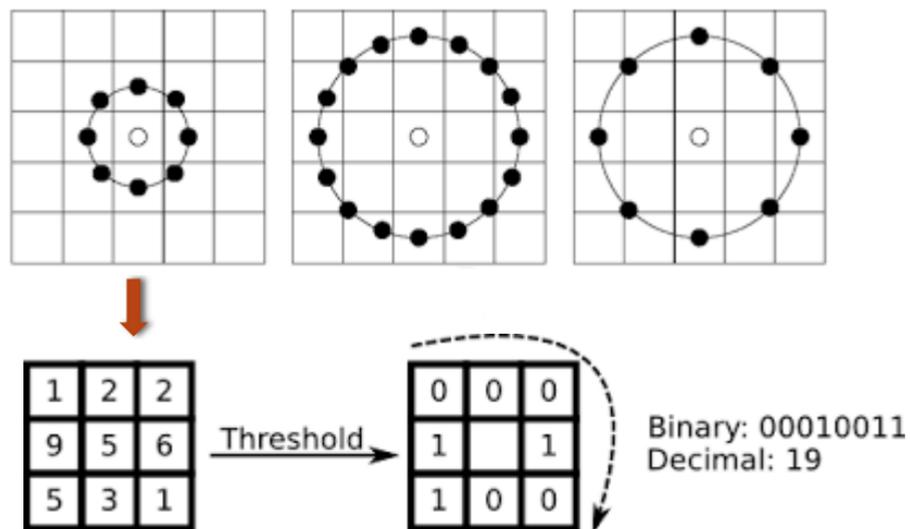
### **2.1.1 LOCAL BINARY PATTERN**

O LBP é uma ferramenta de visão computacional que permite realizar a extração de características de textura local através da comparação do valor de cada pixel com o de seus vizinhos em uma imagem.

Definindo-se a quantidade de vizinhos e distância deles até o pixel central que serão considerados, o LBP começa comparando o valor do pixel central ao de cada um dos seus vizinhos em uma imagem em tons de cinza (ROSEBROCK, 2015). Se o pixel vizinho for inferior ao central, a ele é atribuído o valor 0, e caso contrário, ele recebe o valor 1.

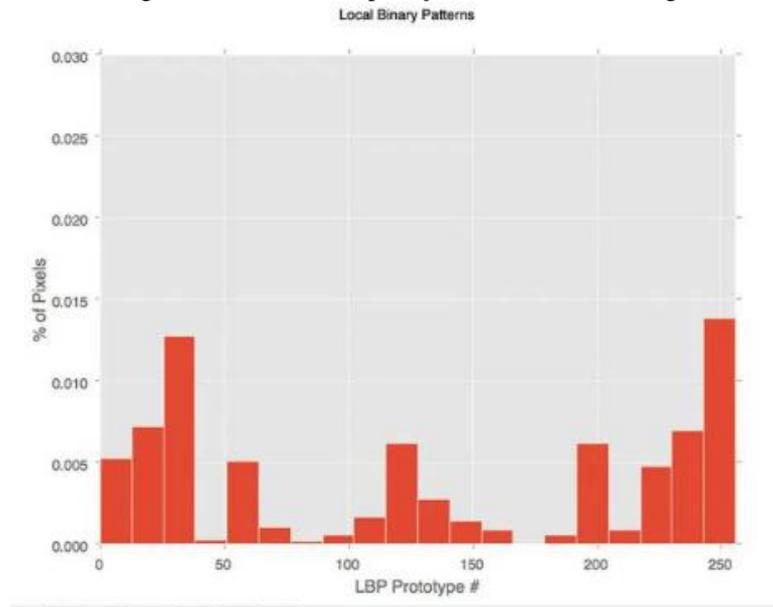
Ainda conforme Rosebrock (2015), partindo de um desses pixels vizinhos e percorrendo os demais na sequência, obtemos um número binário composto pela mesma quantidade de dígitos que a quantidade de pixels vizinhos considerados. Esse número binário é então convertido para a base decimal e seu valor atribuído à posição do pixel central da comparação. Essas etapas são representadas na Figura 4. Esse processo é repetido para todos os pixels da imagem, de modo que a cada momento um deles vai ser o pixel central sob comparação. Ao final deste processo, cada pixel terá como valor o número decimal fruto da conversão do binário obtido e a ocorrência destes valores será computada na forma de histograma, como observado na Figura 5. O resultado final é um vetor que representa a distribuição deste histograma.

Figura 4 – Exemplos de possíveis vizinhanças a serem consideradas e funcionamento do LBP para uma delas.



Fonte: Adaptado de BRAZ JUNIOR ([20--]).

Figura 5 – Histograma resultante da aplicação do LBP a uma imagem.



Fonte: Adaptado de ROSEBROCK (2015).

O LBP tem como vantagens a sua simplicidade e o fato de ser computacionalmente eficiente e robusto à variação monotônica de iluminação, e como desvantagens podemos mencionar o seu baixo desempenho em situações em que o ruído aleatório se faz presente e o fato de não considerar o sinal da diferença dos valores de cinza comparados (BRAZ JUNIOR, [20-]).

## 2.2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é uma subárea da inteligência artificial que envolve uma série de técnicas, permitindo que os computadores aprendam a desempenhar uma tarefa com base nos dados que são apresentados a eles, sem a necessidade de serem explicitamente programados para executar tal tarefa (GÉRON, 2019).

Um modelo de aprendizado de máquina consiste no sistema originado do algoritmo que passou pela etapa de treinamento, sendo capaz de realizar a tarefa proposta quando novos dados forem apresentados. Ele pode ser empregado em problemas de classificação, quando uma classe deve ser atribuída ao item sob análise, ou de regressão, quando deve ser feita a predição do valor numérico de uma variável (GÉRON, 2019).

Dentre os tipos de algoritmos de aprendizado mais populares estão: a regressão linear, K vizinhos mais próximos (KNN), máquinas de vetor de suporte (SVM), árvores de decisão, florestas aleatórias, redes neurais e clusterização *k-means*.

De acordo com Géron (2019), o aprendizado pode ser supervisionado, quando se fornece ao algoritmo exemplos de treinamento que contenham a resposta, ou não supervisionado, quando nos dados de treinamento não é fornecida a resposta do rótulo referente aos itens. O aprendizado pode ainda ser do tipo semi-supervisionado, quando os dados do conjunto de treinamento estão parcialmente rotulados, ou ser do tipo aprendizado por reforço, onde o sistema seleciona e executa ações, recebendo uma recompensa ou penalidade com base nas suas escolhas e atualiza sua estratégia visando maximizar as recompensas.

Em aprendizado de máquina temos o conceito de hiperparâmetros, que são atributos que controlam o treinamento do modelo, mas que não podem ser aprendidos ajustando o modelo aos dados (HERICLIS, 2022). Eles devem ter seu valor definido antes de o treinamento ser iniciado e a escolha desse valor é geralmente feita através do uso de algoritmos como o Grid Search, que será explicado posteriormente.

Entre os exemplos de aplicações do aprendizado de máquina podemos citar os filtros de spam do e-mail, que tenta identificar mensagens recebidas que sejam indesejadas ou tentativas de golpe, detecção de fraudes com cartões de crédito, onde o algoritmo tenta determinar se uma operação foi mesmo realizada pelo dono do cartão, podendo usar como base as informações de padrões de consumo e localidade onde ocorreram as transações. Podemos citar ainda, os exemplos já mencionados de aplicação na área da saúde, para detecção de anomalias nos exames por imagem, e na agricultura, para a identificação de frutos maduros em uma plantação.

### **2.2.1 SUPPORT VECTOR MACHINE**

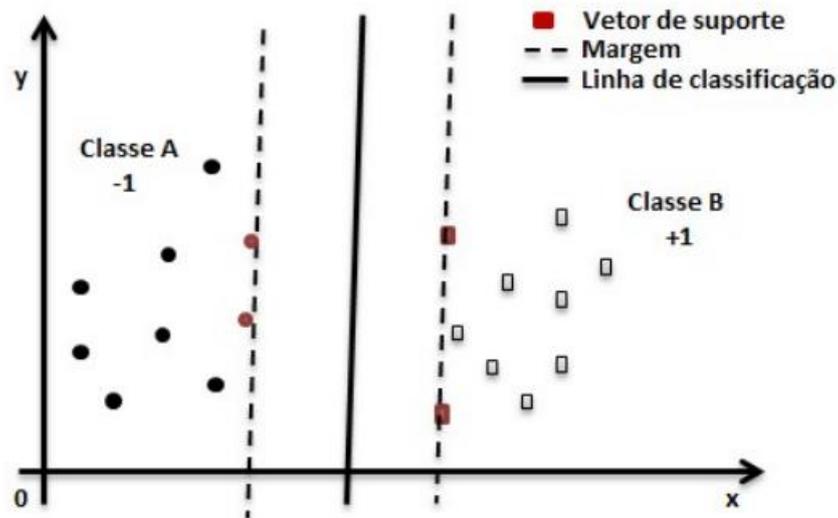
O *Support Vector Machine* (SVM) é um algoritmo de aprendizado de máquina que é capaz de realizar a previsão dos rótulos de objetos nunca expostos a ele antes, desde que, anteriormente tenha sido realizado um treinamento adequado do modelo.

O algoritmo se propõe a estabelecer a melhor linha, plano ou hiperplano para separar duas classes ou mais em seu espaço de atributos (características). A melhor resposta é a que maximize a distância de tal fronteira até os pontos de cada classe mais próximos (LOCA, 2020). Ao mesmo tempo, é desejado que a quantidade de pontos de uma classe que acabam se

localizando no lado oposto da fronteira seja a menor possível. A priorização entre esses dois objetivos pode ser regulada na forma de parâmetros da função que implementa o SVM.

Os atributos dos objetos sob estudo podem ser representados em um espaço  $n$ -dimensional, onde  $n$  é o número de atributos. O SVM é treinado com os dados e estabelece qual o melhor hiperplano de separação entre as classes, de modo a maximizar a margem de separação entre elas. Conforme exposto por Marsland (2009 apud Miranda, 2011, p. 40), os pontos de uma mesma classe que se encontram mais próximos da outra classe são chamados de vetores de suporte e ajudam a estabelecer o posicionamento da fronteira entre as classes, como pode ser observado na Figura 6.

Figura 6 – Funcionamento do SVM para a separação de duas classes.



Fonte: MIRANDA (2011).

O SVM é muito popular porque frequentemente oferece soluções melhores do que outros algoritmos de classificação (MARS LAND, 2008 apud MIRANDA, p. 39). Ele possui vantagens, como as de detectar automaticamente os exemplos mais relevantes do conjunto de dados e ser robusto a *outliers* (CARVALHO, 2005). Os pontos negativos são que seu desempenho pode não ser bom para um conjunto de dados grande, devido à dificuldade do treinamento, e seu mal desempenho na presença de ruído nos dados (ADDAN, 2019).

### 2.2.1.1 KERNEL

Quando o problema de SVM não é linearmente separável no espaço de atributos, pode ser utilizado o chamado truque do kernel. As funções de kernel permitem que em tais situações, o espaço original seja mapeado para um espaço alternativo do kernel, no qual a separação pode potencialmente ser feita de forma mais simples (LIMA, 2014).

Entre os tipos de kernel mais utilizados estão o linear, o RBF, o polinomial e o sigmóide. O kernel linear é o mais simples e é adequado para problemas linearmente separáveis. O kernel RBF (*Radial Basis Function*) é o mais utilizado para problemas não linearmente separáveis, mas o kernel polinomial também pode ser utilizado nestas situações. Exemplos do efeito do kernel na separação do espaço de atributos podem ser observados na Figura 7, onde a divisão das cores ao fundo representa a área correspondente a cada uma das classes. A figura apresenta um exemplo de resultado obtido através da implementação dos kernels dos tipos linear, polinomial e RBF através da função SVC, que implementa o SVM.

As funções utilizadas por cada tipo de kernel podem ser verificadas nas equações abaixo. A equação (1) representa a função que caracteriza o kernel linear. Na equação (2), podemos observar a função do kernel polinomial, onde  $d$ ,  $r$  e  $\gamma$  (*gamma*) são determinados pelos parâmetros *degree*, *coef0* e *gamma* da função que implementa o SVM, respectivamente. A equação (3) representa a função do kernel RBF e o parâmetro  $\gamma$  deve ser maior do que 0. Por fim, a equação (4) representa a função do kernel do tipo sigmoide.

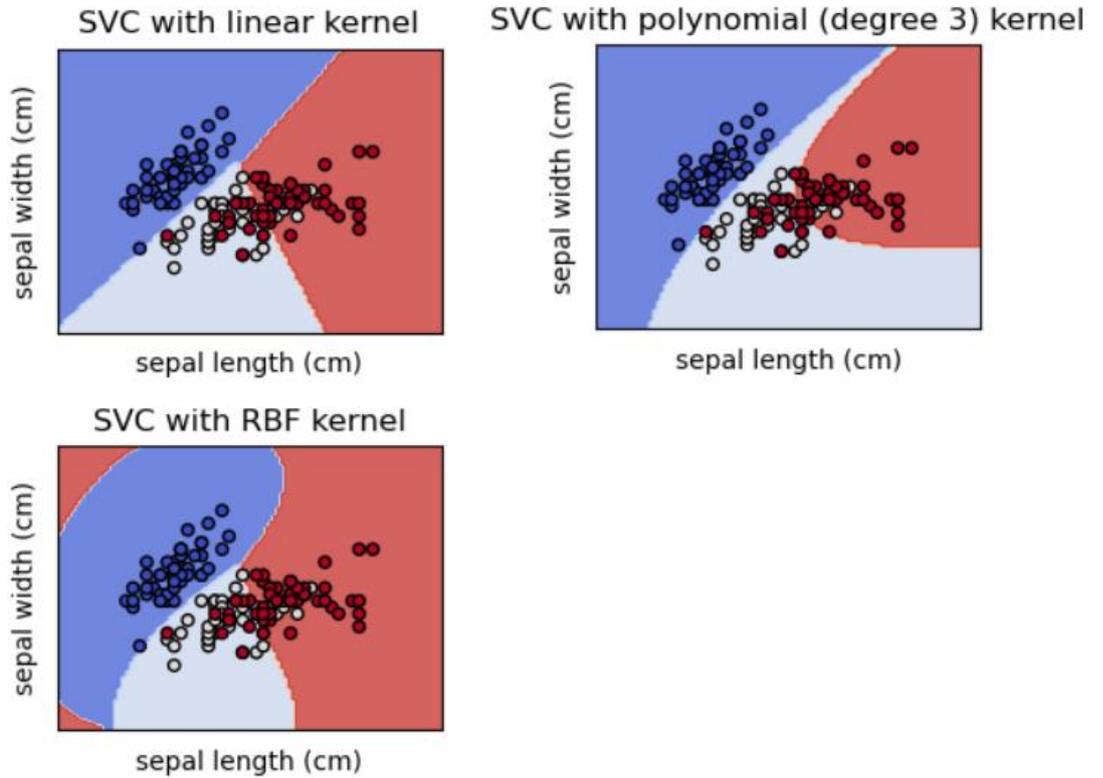
$$K(x, x') = \langle x, x' \rangle \quad (1)$$

$$K(x, x') = (\gamma \langle x, x' \rangle + r)^d \quad (2)$$

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (3)$$

$$K(x, x') = \tanh(\gamma \langle x, x' \rangle + r) \quad (4)$$

Figura 7 - Efeito do kernel na separação do espaço de atributos entre as diferentes classes.

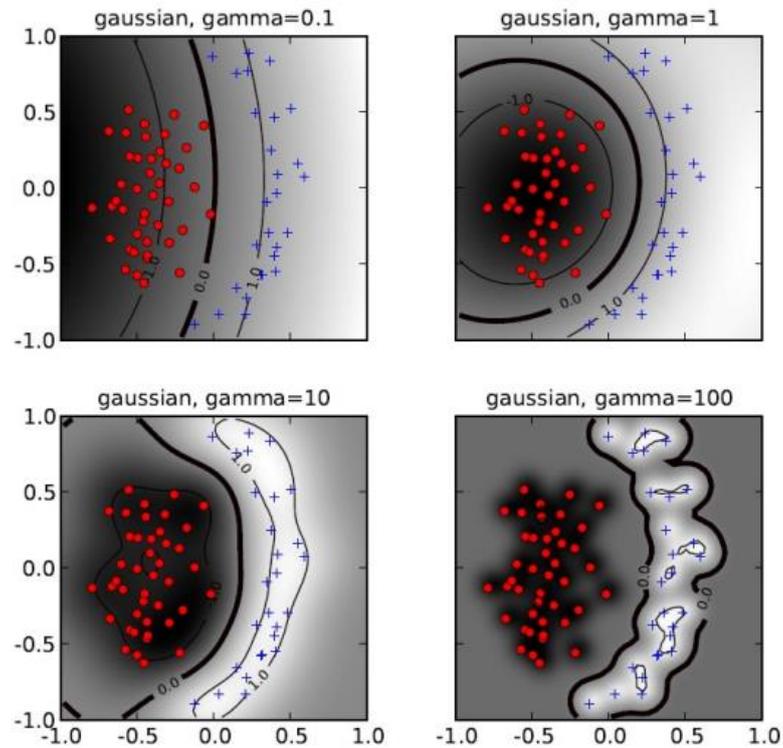


Fonte: Adaptado de PEDREGOSA et al. ([20--])

#### 2.2.1.1.1 PARÂMETRO GAMMA

O *Gamma* é um parâmetro do kernel, aplicável para os kernels dos tipos RBF, polinomial e sigmoide. Ele controla a flexibilidade do classificador, de modo que a fronteira entre as classes tem a sua forma mais ajustada ao conjunto (LIMA, 2014). O efeito do *Gamma* pode ser observado na Figura 8, onde à medida em que *Gamma* aumenta, a fronteira se torna mais flexível e envolve uma das classes de maneira mais ajustada aos seus objetos.

Figura 8 - Efeito do parâmetro *Gamma* na fronteira de separação, mantendo fixos os demais parâmetros.

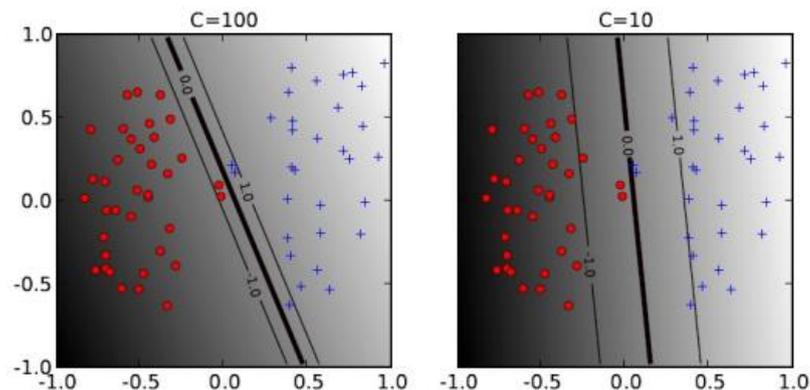


Fonte: BEN-HUR; WESTON (2009).

### 2.2.1.2 HIPERPARÂMETRO C

O Hiperparâmetro C do SVM é um parâmetro de regularização e caracteriza o peso da penalidade por classificações incorretas (LIMA, 2014). É o “custo dos erros”. Ele acaba influenciando no tamanho da margem de separação entre as classes. Se o custo é alto, essas margens se tornam mais estreitas, e se o custo é mais baixo, as margens se tornam mais largas, como pode ser observado na Figura 9.

Figura 9 - Efeito do parâmetro C na fronteira de separação.



Fonte: BEN-HUR; WESTON (2009).

### 2.2.2 GRID SEARCH

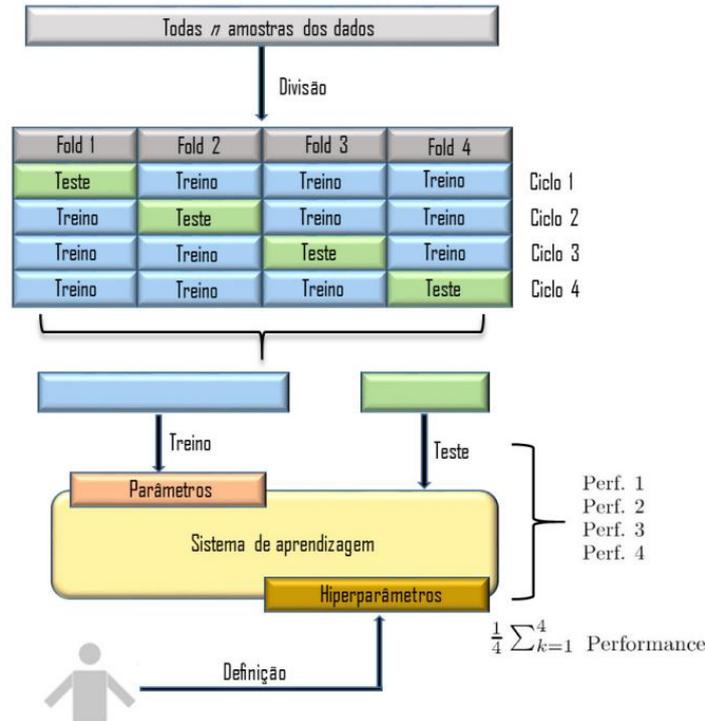
O conceito de *Grid Search* está ligado à otimização de parâmetros e se baseia no teste de desempenho do modelo com diferentes combinações destes (LIMA, 2014). São estabelecidas faixas dos parâmetros que se deseja otimizar e o algoritmo calcula a acurácia de cada uma das combinações, e por fim informa qual a melhor escolha.

Para SVMs o algoritmo de *Grid Search* é usualmente empregado para apontar quais os melhores valores para o parâmetro *Gamma* do kernel, o hiperparâmetro C e qual o tipo de kernel mais adequado à situação.

### 2.2.3 VALIDAÇÃO CRUZADA

A validação cruzada, também conhecida como *Cross Validation*, é utilizada para verificar a validade do modelo e para evitar o sobreajuste, ou *overfitting*. O sobreajuste acontece quando um modelo se ajusta muito bem ao conjunto de dados de treinamento, mas se mostra ineficaz ao ser aplicado a novos conjuntos de dados (GÉRON, 2019). Conforme exposto por Fontana (2020), um dos métodos mais utilizados para a validação cruzada é o *K-Fold*. Nele, os dados disponíveis e já rotulados são divididos em N subconjuntos, onde valores tipicamente utilizados de N são 5 ou 10. Um exemplo do que é feito no *K-fold* e do cálculo da performance média pode ser verificado na Figura 10, neste caso separando os dados em 4 subconjuntos. A partir daí é realizada a verificação de desempenho ao utilizar N-1 subconjuntos como conjunto de treinamento e o subconjunto restante como conjunto de teste, de modo que a cada combinação o conjunto de teste será diferente (FONTANA, 2020). Alternativamente, pode ser definida a porcentagem dos dados que será utilizada como conjunto de teste. Com as estatísticas de desempenho de cada uma das combinações podemos calcular a acurácia média, e assim chegar a uma medida mais confiável que nos indique a validade e consistência do modelo.

Figura 10 - Exemplo de funcionamento do *K-Fold* e validação cruzada.



Fonte: LOCA (2020).

## 2.3 TRABALHOS RELACIONADOS

O desenvolvimento de modelos de aprendizado de máquina para a classificação de doenças através da análise de imagens de folhas do arroz já foi explorado em alguns estudos. A maioria dos trabalhos consultados empregava abordagens baseadas em aprendizado profundo (*deep learning*), enquanto as outras empregavam abordagens baseadas em SVMs ou árvores de decisão.

Lorensi *et al.* (2021) empregou árvores de decisão para a classificação das imagens referentes a três doenças: Brusone, Mancha-parda e Mancha Estreita. Foi utilizado um conjunto de 112 imagens. O conjunto de dados gerado a partir das imagens era composto por 10 atributos. Desses atributos, quatro foram obtidos com o uso do *Software InfraView 64-bit Version*: averageR, averageG, averageB e averageGray. Os demais atributos foram obtidos através da análise da imagem, sendo eles: averageRGBGray, formato\_doenca, area\_doenca, borda\_folha\_manchada e granularidade\_manchas. Utilizando para a avaliação final uma parcela de 29% das imagens disponíveis, foi obtida uma acurácia de 100% no conjunto de testes (32 imagens).

Pothen e Pai (2020) realizaram a segmentação das imagens utilizando a binarização de Otsu e extraíram características de textura com LBP e de estrutura e forma com HOG (histograma de gradientes orientados), para em seguida realizar a classificação com um modelo baseado em SVM. Neste trabalho as doenças em estudo foram a Mancha Bacteriana das Folhas (*Bacterial Leaf Blight*), o Carvão da folha (*Leaf Smut*) e a Mancha-parda (*Brown Spot*). Foram realizadas duas versões do método, uma utilizando o descritor LBP e a outra utilizando o HOG, ambas com classificação por SVM. O melhor resultado foi uma acurácia de 94,6%, utilizando HOG e o SVM com kernel polinomial. Os resultados como um todo, considerando as variações da abordagem e explicitando o uso de diferentes tipos de kernel, podem ser visualizados na Figura 11. Vale ressaltar que a saída do LBP é um vetor muito mais compacto que a do HOG.

Figura 11 - Resultados de acurácia obtidos por Pothen e Pai (2020).

Feature Descriptor \ Classifier	LBP	HOG
SVM Linear	89.00	92.01
SVM: Polynomial	90.23	<b>94.6</b>
SVM: RBF	86.21	89.0

Fonte: POTHEN E PAI (2020).

Hossain *et al.* (2021) desenvolveu um estudo para a classificação das imagens entre 5 categorias, cada uma referente a uma doença do arroz, empregando um modelo de rede neural convolucional (CNN) customizado, utilizando camadas de convolução, pooling, camada densa e camada *softmax*. As doenças analisadas foram Brusone (*Leaf Blast*), Mancha-parda (*Brown Spot*), Mancha Bacteriana das Folhas (*Bacterial Leaf Blight*), Queima da Bainha (*Sheath Blight*) e Tungro (*Tungro*). O conjunto de imagens foi obtido através da aplicação de *data augmentation* em imagens obtidas do Instituto Internacional do Arroz e Instituto do Arroz de Bangladesh. Neste estudo havia a preocupação em permitir a utilização em dispositivos com limitação de memória. A acurácia obtida foi de 97,82%. As CNNs são classificadas como um algoritmo de aprendizado profundo.

Ainda nas abordagens envolvendo aprendizado profundo, Deng *et al.* (2021) utilizou uma estratégia baseada no uso de *Ensemble Model*, para a classificação de seis doenças do arroz, obtendo uma acurácia geral de 91%, enquanto Wang, Wang e Peng (2022)

desenvolveram uma estratégia baseada em redes neurais profundas com otimização Bayesiana, aplicada a uma classificação entre 4 doenças do arroz e obtendo uma acurácia de 94,65%.

### 3 MATERIAIS E MÉTODO

Nesta seção serão abordados os principais materiais utilizados na elaboração deste trabalho e o funcionamento do método proposto.

#### 3.1 MATERIAIS

Para a elaboração deste trabalho foi importante o uso de materiais como o conjunto de imagens rotuladas do problema e as bibliotecas de programação.

O conjunto de imagens utilizado é composto por um conjunto de treinamento contendo 350 imagens de cada uma das três classes: Brusone, Mancha-parda e Saudável, bem como por um conjunto de testes de 88 imagens de cada uma das três classes mencionadas, totalizando 1314 imagens. Todas as imagens estão rotuladas de acordo com a classe à qual pertencem. Os dados foram obtidos de Saputra (2022), disponíveis no site Kaggle.

O algoritmo foi desenvolvido na linguagem de programação Python e as principais bibliotecas utilizadas foram OpenCV e Scikit-Learn. A biblioteca OpenCV oferece ferramentas que facilitam a manipulação e processamento de imagens digitais, permitindo realizar tarefas como redimensionamento, aplicação de filtros contra ruídos e obtenção de contornos. A biblioteca Scikit-Learn oferece ferramentas que tornam fácil a implementação de algoritmos de aprendizado de máquina. Além destas, também foram utilizadas as bibliotecas NumPy, Scikit-Image, Imutils, Argparse, OS e Matplotlib.

#### 3.2 MÉTODO

A sequência de implementação dos algoritmos neste trabalho consistiu em três etapas. Inicialmente foi executada a aplicação do LBP às imagens do conjunto de treinamento e utilização do *Grid Search* para definir os melhores parâmetros para utilizar no SVM. Em seguida foi implementado o algoritmo principal, que aplica o LBP às imagens de treinamento, utiliza o resultado para treinar o SVM, aplica o LBP às imagens de teste e faz a classificação destas. Por fim, o algoritmo de validação cruzada testou o desempenho do algoritmo, aplicando o LBP e SVM a subdivisões do conjunto de treinamento.

### 3.2.1 GRID SEARCH

Primeiramente, foram importadas as bibliotecas a serem utilizadas e inicializadas as variáveis que seriam usadas em seguida. Foi estabelecida a faixa de valores a serem testados para os parâmetros do SVM, sendo: para o parâmetro  $C$ , os valores 10, 100, 1000 e 10000, para o  $\Gamma$ , os valores 1, 10, 100 e 1000, e para o kernel, as variações linear, RBF, polinomial e sigmoide.

Realizando a varredura da pasta de treinamento, foi aplicado o LBP a cada uma das imagens do conjunto, convertidas as imagens para uma escala de tons de cinza e redimensionadas para um tamanho padrão, e as características e rótulos armazenados em duas listas separadas.

Foi aplicada a função `ShuffleSplit` para a divisão do conjunto de treinamento em 10 subconjuntos, com combinações destes de modo que os subconjuntos de teste representem 20% do total das imagens. A função `GridSearchCV` foi chamada para realizar a combinação de parâmetros do SVM e o método `fit` para treinar o modelo. Para finalizar, foi retornada e informada a melhor combinação de valores testados.

### 3.2.2 ALGORITMO PRINCIPAL

No algoritmo principal, que aplica o LBP a todas as imagens dos conjuntos e realiza a classificação de todas as imagens de teste utilizando o SVM, primeiramente, são importadas as bibliotecas e inicializadas as variáveis.

De maneira semelhante à descrita anteriormente, foi realizada a varredura da pasta de treinamento, aplicado o LBP a cada uma das imagens do conjunto, em tons de cinza e redimensionadas, e as suas características e rótulos armazenados em duas listas separadas.

O modelo de SVM com os melhores parâmetros  $C = 1000$ ,  $\Gamma = 100$  e kernel RBF foi treinado com as características e rótulos extraídos na etapa anterior.

Da mesma forma, foi realizada a varredura da pasta de teste, aplicado o LBP a cada uma das imagens do conjunto, em tons de cinza e redimensionadas, e as suas características extraídas foram utilizadas para a predição do rótulo de classe, o qual foi armazenado em uma lista. A informação do rótulo da classe verdadeira também foi armazenada em uma lista, para posterior definição da acurácia do modelo.

Para avaliar o desempenho do modelo foi calculada a acurácia e plotado o gráfico da matriz de confusão.

### **3.2.3 VALIDAÇÃO CRUZADA**

Para a validação cruzada foi iniciado de maneira semelhante aos anteriores quanto às bibliotecas e variáveis. Foi realizada também, a varredura das imagens de treinamento e as características extraídas e rótulos utilizados no SVM.

Foi aplicada a função `ShuffleSplit` para a divisão do conjunto de treinamento em 10 subconjuntos, com combinações destes, de modo que os subconjuntos de teste representem 20% do total das imagens. A função `cross_val_score` foi chamada para realizar a determinação da acurácia para cada combinação de subconjuntos. Para finalizar foram informados os resultados de acurácia de cada combinação, a acurácia média e seu desvio-padrão.



## 4 RESULTADOS

Nesta seção serão expostos os resultados obtidos nas etapas deste trabalho, desde o apontamento dos melhores valores para os parâmetros do SVM, o resultado da execução do algoritmo principal com os conjuntos de treinamento e de teste completos, até a validação do modelo.

### 4.1 RESULTADOS DA OTIMIZAÇÃO DE PARÂMETROS

A execução do algoritmo de otimização de parâmetros *Grid Search* apontou os melhores valores dentre as alternativas como sendo:  $C = 1000$ ,  $\text{Gamma} = 100$  e  $\text{kernel} = \text{'rbf'}$ .

Como extensão desta informação, pode ser observado na Tabela 1 os principais resultados das combinações de parâmetros e a influência de cada um individualmente.

Tabela 1 – Resultados de acurácia de diferentes combinações de parâmetros do SVM.

Parâmetros			Acurácia
C	Gamma	Kernel	
100	100	RBF	79,17%
10000	100	RBF	80,68%
1000	10	RBF	76,89%
1000	1000	RBF	81,44%
1000	100	RBF	81,81%
1000	100	Polinomial	78,79%

Fonte: Autor (2022).

### 4.2 RESULTADO DA CLASSIFICAÇÃO

Foi obtida para o modelo proposto, com os conjuntos de treinamento e teste inteiros, uma acurácia de 81,81%. Exemplos de classificações corretas e incorretas podem ser observadas na Figura 12 e Figura 13, respectivamente. Nestas imagens, o texto no canto superior esquerdo representa a classificação realizada pelo modelo, enquanto no canto superior

direito é informada a verdadeira classe à qual a imagem pertence. A matriz de confusão pode ser observada na Figura 14.



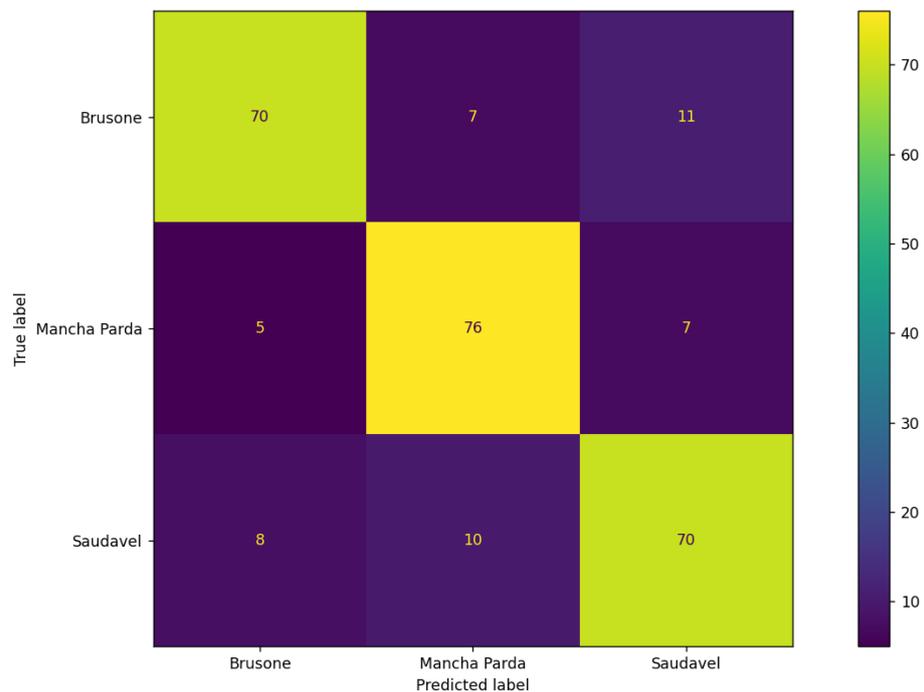
Fonte: Adaptado de SAPUTRA (2022).

Figura 13 - Imagens classificadas incorretamente.



Fonte: Adaptado de SAPUTRA (2022).

Figura 14 - Matriz de confusão.



Fonte: Autor (2022).

Observando os resultados de classificações incorretas, podemos entender as limitações do modelo. Possíveis motivos para algumas das classificações incorretas são: quando as manchas nas folhas se encontram em estágios iniciais, sendo poucas e pequenas, podendo ser interpretadas pelo modelo como saudáveis; quando as duas doenças apresentam aparência semelhante, por exemplo no começo da manifestação do Brusone na folha com manchas ainda predominantemente marrons, onde uma doença pode ser confundida com a outra; e por fim nos casos onde as folhas saudáveis apresentam pequenas imperfeições que apesar de não representarem a manifestação de doença acabam sendo interpretadas como tal.

O resultado geral de acurácia de 81,81% se mostrou um valor razoavelmente satisfatório, apesar de inferior aos 90% estabelecidos como objetivo. Uma das justificativas do menor desempenho em relação aos trabalhos relacionados pode ter sido a utilização de um método mais simples, onde apenas as características extraídas pelo LBP foram utilizadas.

Para melhorar a acurácia do modelo, podem ser adicionadas novas características à análise, de modo a fornecer mais informações sobre cada imagem e permitindo uma melhor diferenciação entre as classes. Essas características podem ser por exemplo de cor ou formato. O problema de adicionar novas características a serem analisadas é que o modelo se torna cada

vez mais complexo, e a melhora da acurácia pode ser significativa ou não, cabendo a cada situação ser avaliada para decidir se vale a pena esta complexidade adicional.

#### 4.3 RESULTADO DA VALIDAÇÃO CRUZADA

A validação cruzada resultou em uma acuraria média de 80,67%, com um desvio padrão de 2,46%. Os resultados individuais estão representados na Tabela 2. A validação mostrou que os resultados estão próximos entre si e a acurácia média apresentou um bom resultado. Podemos dizer que não houve *overfitting* do modelo.

Tabela 2 – Resultados individuais da validação cruzada.

Combinação	Acurácia
1	78,10%
2	76,67%
3	80,95%
4	81,43%
5	81,90%
6	82,86%
7	80,95%
8	85,71%
9	79,05%
10	79,05%

Fonte: Autor (2022).



## 5 CONCLUSÃO

Neste trabalho foi explorado o problema das principais doenças que afetam o cultivo de arroz e proposta uma solução baseada em visão computacional e aprendizado de máquina. A extração das características a serem analisadas foi feita através do uso do LBP, e o modelo de aprendizado de máquina para a classificação foi baseado no algoritmo de SVM. Com a otimização de parâmetros foi possível atingir o desempenho máximo para as ferramentas e conjuntos de imagens utilizados. A validação cruzada apontou que o modelo não está sobreajustado aos dados de treinamento. Foi possível verificar os resultados e entender as limitações da abordagem proposta, que apesar de ter obtido um resultado inferior ao desejado, ainda pode ter um desempenho considerado bom.

Foi evidenciada a importância de técnicas auxiliares como o Grid Search e validação cruzada, para garantir a otimização de parâmetros e validade do modelo proposto. A otimização de parâmetros permitiu observar a influência de cada parâmetro testado, de maneira individual. Além disso, ficaram claras as situações que representaram dificuldade para a correta classificação pelo modelo, representando as suas limitações.

O estudo para realizar este trabalho possibilitou a obtenção de uma visão geral das principais ferramentas das áreas de processamento de imagens, visão computacional e aprendizado de máquina.

Com o trabalho elaborado foi possível vivenciar na prática a execução das etapas da resolução de um problema utilizando as ferramentas das áreas mencionadas. Ele permitiu verificar que é possível chegar a soluções muito interessantes mesmo com o uso de ferramentas relativamente simples.

Como possibilidade de trabalhos futuros fica estabelecido o estudo de novas características a serem incorporadas à análise do problema, visando aumentar os níveis de acurácia do modelo.

## REFERÊNCIAS

- ADDAN, Diego. **Support Vector Machine**. [S.L.]: Unibrasil, 2019. 34 slides, color.
- BARELLI, Felipe. **Introdução à Visão Computacional**: uma abordagem prática com python e opencv. São Paulo: Casa do Código, 2019. 256 p.
- BEN-HUR, Asa; WESTON, Jason. A User's Guide to Support Vector Machines. **Methods In Molecular Biology**, [S.L.], v. 609, p. 223-239, 30 out. 2009. Humana Press.  
[http://dx.doi.org/10.1007/978-1-60327-241-4\\_13](http://dx.doi.org/10.1007/978-1-60327-241-4_13).
- BRAZ JUNIOR, Geraldo. **Introdução a Local Binary Patterns**. [S.I.], [20--]. 22 slides, color. Disponível em: [https://nca.ufma.br/~geraldo/vc/14.b\\_lbp.pdf](https://nca.ufma.br/~geraldo/vc/14.b_lbp.pdf). Acesso em: 25 nov. 2022.
- CARVALHO, Bernardo. **O estado da arte em métodos para reconhecimento de padrões: Support Vector Machine**. Universidade Federal de Minas Gerais - UFMG, 2005.
- DENG, Ruoling *et al.* Automatic Diagnosis of Rice Diseases Using Deep Learning. **Frontiers In Plant Science**, [S.L.], v. 12, p. 1-16, 19 ago. 2021. Frontiers Media SA.  
<http://dx.doi.org/10.3389/fpls.2021.701038>. Disponível em:  
<https://www.frontiersin.org/articles/10.3389/fpls.2021.701038/full>. Acesso em: 20 jul. 2022.
- FERNANDES NETO, Euripedes Purcinio. **VISÃO COMPUTACIONAL PARA IDENTIFICAÇÃO DE CORES EM TEMPO REAL COM OPENCV E PYTHON**. 2020. 17 f. TCC (Graduação) - Curso de Engenharia de Computação, Faculdade de Tecnologia e Ciências Sociais Aplicadas – Fatecs Curso, Centro Universitário de Brasília, Brasília, 2020. Disponível em:  
<https://repositorio.uniceub.br/jspui/bitstream/prefix/15103/1/Eur%C3%ADpedes%20Vis%C3%A3o%20Computacional%20VF%20PB.pdf>. Acesso em: 21 nov. 2022.
- FILIPPI, Marta Cristina Corsi de *et al.* **Brusone no arroz**. Brasília: Embrapa, 2015. 26 p.
- FONTANA, Éliton. **Introdução aos Algoritmos de Aprendizagem Supervisionada**. [Curitiba], 2020. 26 slides, color. Disponível em:  
[https://fontana.paginas.ufsc.br/files/2018/03/apostila\\_ML\\_pt2.pdf](https://fontana.paginas.ufsc.br/files/2018/03/apostila_ML_pt2.pdf). Acesso em: 20 ago. 2022.
- GÉRON, Aurélien. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow**: conceitos, ferramentas e técnicas para a construção de sistemas inteligentes. Rio de Janeiro: Alta Books, 2019. 576 p.
- HARRISON, Matt. **Machine Learning**: guia de referência rápida. São Paulo: Novatec, 2020. 272 p.
- HERICLIS, Samuel. **Hiperparâmetros - por quê são importantes**. 2022. Disponível em:  
<https://blog.dsbrigade.com/hiperparametros-por-que-sao-importantes/>. Acesso em: 20 nov. 2022.

HOSSAIN, Syed Md. Minhaz *et al.* Rice Leaf Diseases Recognition Using Convolutional Neural Networks. **Advanced Data Mining And Applications**, [S.L.], v. 0, n. 0, p. 299-314, jan. 2021. Springer International Publishing. [http://dx.doi.org/10.1007/978-3-030-65390-3\\_23](http://dx.doi.org/10.1007/978-3-030-65390-3_23). Disponível em: [https://www.researchgate.net/publication/348251143\\_Rice\\_Leaf\\_Diseases\\_Recognition\\_Using\\_Convolutional\\_Neural\\_Networks](https://www.researchgate.net/publication/348251143_Rice_Leaf_Diseases_Recognition_Using_Convolutional_Neural_Networks). Acesso em: 20 jul. 2022.

LIMA, Rodrigo Lucio de. **AVALIAÇÃO DO ALGORÍTIMO SVM NA DETECÇÃO DE COMPORTAMENTOS SUSPEITOS EM CENAS DE VÍDEO**. 2014. 60 f. TCC (Graduação) - Curso de Análise e Desenvolvimento de Sistemas, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2014. Disponível em: [https://repositorio.utfpr.edu.br/jspui/bitstream/1/16763/3/PG\\_COADS\\_2014\\_2\\_04.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/16763/3/PG_COADS_2014_2_04.pdf). Acesso em: 25 nov. 2022.

LOCA, Antonio Luiz da Silva. **Uma metodologia experimental para avaliar abordagens de aprendizado de máquina para diagnóstico de falhas com base em sinais de vibração**. 2020. 126 f. Dissertação (Mestrado) - Curso de Informática, Centro Tecnológico, Universidade Federal do Espírito Santo, Vitória, 2020.

LORENSI, Rui Antonio da Rocha *et al.* Um modelo de classificação de doenças da folha do arroz através de imagens. In: CONGRESO ARGENTINO DE AGROINFORMÁTICA, 8., 2021, [S.I.]. **XIII Congreso de AgroInformática (CAI 2021) - JAIIO 50 (Modalidad virtual)**. [S.I.]: Sociedad Argentina de Informática e Investigación Operativa, 2021. p. 115-124. Disponível em: <http://sedici.unlp.edu.ar/handle/10915/140693>. Acesso em: 10 ago. 2022.

MIRANDA, Bruno de Souza. **ALGORITMOS CLÁSSICOS DE APRENDIZADO DE MÁQUINA APLICADOS AO PROBLEMA DO RECONHECIMENTO DE IMAGENS**. 2011. 87 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal do Pampa, Alegrete, 2011.

NETO, Aroldo Antonio de Oliveira (org.). **A cultura do arroz**. Brasília: Conab, 2015. 180 p. Disponível em: [https://biblioteca.conab.gov.br/phl82/pdf/2015\\_Cultura\\_do\\_arroz.pdf](https://biblioteca.conab.gov.br/phl82/pdf/2015_Cultura_do_arroz.pdf). Acesso em: 01 nov. 2022.

PEDREGOSA, Fabian *et al.* Scikit-learn: Machine Learning in Python. **Journal Of Machine Learning Research**. [S.L.], p. 2825-2830. out. 2011.

PEDREGOSA, Fabian *et al.* **Support Vector Machines**. [20--]. Disponível em: <https://scikit-learn.org/stable/modules/svm.html#svm-kernels>. Acesso em: 20 nov. 2022.

POTHEN, Minu Eliz; PAI, Maya L. Detection of Rice Leaf Diseases Using Image Processing. In: FOURTH INTERNATIONAL CONFERENCE ON COMPUTING METHODOLOGIES AND COMMUNICATION (ICCMC 2020), 4., 2020, [S.I.]. **Proceedings of the Fourth International Conference on Computing Methodologies and Communication (ICCMC 2020)**. [S.L.]: Ieee, 2020. p. 424-430.

PRABHU, Anne Sitrama *et al.* **Principais doenças do arroz no Brasil**. 3. ed. Goiânia: Embrapa-Cnpaf, 1995. 43 p.

- ROSEBROCK, Adrian. **Local Binary Patterns with Python & OpenCV**. 2015. Disponível em: <https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>. Acesso em: 20 nov. 2022.
- SAPUTRA, Dede Ikhsan Dwi. **Rice Leafs Disease Dataset**. 2022. Disponível em: <https://www.kaggle.com/datasets/dedeikhsandwisaputra/rice-leafs-disease-dataset>. Acesso em: 10 ago. 2022.
- SARAVIA, Cristine Thomaz. **MANEJO DA TEMPERATURA DO AR NA SECAGEM INTERMITENTE EM SEMENTES DE ARROZ**. 2006. 58 f. Tese (Doutorado) - Curso de Ciências, Programa de Pós-Graduação em Ciência e Tecnologia de Sementes, Universidade Federal de Pelotas, Pelotas, 2006. Disponível em: [http://guaiaca.ufpel.edu.br/bitstream/123456789/1444/1/tese\\_cristine\\_saravia.pdf](http://guaiaca.ufpel.edu.br/bitstream/123456789/1444/1/tese_cristine_saravia.pdf). Acesso em: 20 nov. 2022.
- SCHEUERMANN, Klaus Konrad; NESI, Cristiano Nunes. Controle químico de brusone e mancha parda na cultura do arroz irrigado. **Summa Phytopathologica**, [S.L.], v. 47, n. 3, p. 168-172, set. 2021. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/0100-5405/251530>.
- SILVA, Osmira Fátima da. **Consumo Per capita de Arroz (Oryza sativa L.) e de Feijão-comum (Phaseolus vulgaris L.), no Brasil, de 1985 a 2021\***. 2022. Disponível em: <http://www.cnpaf.embrapa.br/socioeconomia/docs/arroz/consumopercapitaarrozfeijao.htm>. Acesso em: 25 nov. 2022.
- SILVA-LOBO, Valacia Lemes; FILIPPI, Marta Cristina Corsi de. **Manual de Identificação de Doenças da Cultura do Arroz**. Brasília: Embrapa, 2017. 45 p.
- WANG, Yibin; WANG, Haifeng; PENG, Zhaohua. Rice Diseases Detection and Classification Using Attention Based Neural Network and Bayesian Optimization. **Elsevier**, [S.L.], p. 1-22, 5 jan. 2022.