Universidade Estadual de Londrina

ANDRÉ FERREIRA CORDEIRO

CLASSIFICAÇÃO DE IMAGENS DE SATÉLITE DA FLORESTA AMAZÔNICA UTILIZANDO PROCESSAMENTO DE IMAGEM E APRENDIZADO DE MÁQUINA

LONDRINA
2023

ANDRÉ FERREIRA CORDEIRO

CLASSIFICAÇÃO DE IMAGENS DE SATÉLITE DA FLORESTA AMAZÔNICA UTILIZANDO PROCESSAMENTO DE IMAGEM E APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Departamento de Computação da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alan Salvany Felinto

LONDRINA
2023

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

F383c Cordeiro, André Ferreira.
CLASSIFICAÇÃO DE IMAGENS DE SATÉLITE DA FLORESTA AMAZÔNICA UTILIZANDO PROCESSAMENTO DE IMAGEM E APRENDIZADO DE MÁQUINA / André Ferreira Cordeiro. - Londrina, 2023.
57 f. : il.

Orientador: Alan Salvany Felinto.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Graduação em Ciência da Computação, 2023.
Inclui bibliografia.

1. Aprendizado de Máquina - TCC. 2. Processamento de Imagem - TCC. 3. Floresta Amazônica - TCC. 4. Segmentação de imagem - TCC. I. Felinto, Alan Salvany. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Graduação em Ciência da Computação. III. Título.

CDU 519

ANDRÉ FERREIRA CORDEIRO

CLASSIFICAÇÃO DE IMAGENS DE SATÉLITE DA FLORESTA AMAZÔNICA UTILIZANDO PROCESSAMENTO DE IMAGEM E APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação do Departamento de Computação da Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina

Prof. Dr. Daniel dos Santos Kaster
Universidade Estadual de Londrina – UEL

Prof. Dr. Osvaldo Coelho Pereira Neto
Universidade Estadual de Londrina – UEL

Londrina, 2 de junho de 2023.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a minha família e amigos por todo suporte durante estes anos e por todo o incentivo para que eu trilhasse meu caminho até aqui. Agradeço também aos meus amigos e colegas de turma por todo o apoio e pelos momentos de descontração, tão necessários nos últimos anos. E, por último, agradeço aos professores da universidade por todo o conhecimento compartilhado e, em especial, ao Professor Alan Salvany Felinto por toda a orientação oferecida durante a realização deste trabalho e ao Professor Osvaldo Coelho Pereira Neto pelo auxílio no desenvolvimento do projeto.

"We are taught hopeless death
To break down our resistance to it
This is a hallmark of the unholy temple
We must fight the unnatural cause of
anti-humanity
And listen to the Earth's word" (Gus
Lobban e Sarah Bonito, 2021).

CORDEIRO, A. F.. **Classificação de imagens de satélite da floresta amazônica utilizando Processamento de Imagem e Aprendizado de Máquina**. 2023. 57f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

O ecossistema da floresta amazônica desempenha um papel fundamental para o planeta, auxiliando no combate às alterações climáticas e provendo serviços ecológicos como a garantia da qualidade dos solos e das reservas de água. Apesar disso, nas últimas décadas o processo de desmatamento na região tem avançado a níveis preocupantes, ameaçando tanto a sua biodiversidade quanto a permanência das comunidades locais. Com isso, diversos projetos têm sido desenvolvidos para monitoramento das transformações que ocorrem na região, visando a proteção do ambiente contra o desmatamento. Somado a isso, os recentes avanços nas áreas de Sensoriamento Remoto, Aprendizado de Máquina e Processamento de Imagem têm auxiliado na automatização desse processo e na melhoria dos resultados. Portanto, visando contribuir para o estado da arte neste tópico, o objetivo deste trabalho foi realizar o teste de diversos algoritmos de aprendizado de máquina na classificação de imagens de satélite da região da floresta amazônica, com o intuito de desenvolver uma tecnologia que possa ser usada para monitorar as mudanças na cobertura de solo da região. Para isso, foram testados quatro algoritmos de classificação sobre variações de uma mesma base de dados, que se diferenciam na aplicação de um filtro gaussiano no pré-processamento das imagens originais. Ao final, foi desenvolvida uma estratégia de segmentação por partes, que se utiliza de dois modelos aplicados em sequência sobre uma imagem de entrada. Um modelo para classificação de pixels de floresta e corpos d'água, e um segundo modelo para diferenciação de pixels de solo nu e agropecuária, que alcançaram, respectivamente, 93,6% e 95,1% de acurácia utilizando o algoritmo *Random Forest*. Com isso, concluiu-se que a estratégia implementada teve um desempenho satisfatório e que há indícios de que a aplicação do filtro gaussiano nas imagens pode melhorar os resultados obtidos.

Palavras-chave: Aprendizado de Máquina. Processamento de Imagem. Floresta Amazônica. Segmentação de imagem.

CORDEIRO, A. F.. **Classification of satellite images of the Amazon rainforest using Image Processing and Machine Learning**. 2023. 57p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina, 2023.

ABSTRACT

The Amazon rain forest plays a fundamental role for the planet, helping to stabilize climate change and providing ecological services such as guaranteeing the quality of soils and water reserves. Despite this, in recent decades the process of deforestation in the region has advanced to worrying levels, threatening both it's biodiversity and the permanence of local communities. As a result, several projects have been developed to monitor the changes taking place in the region, seeking to protect the environment from deforestation. Besides this, recent advances in Remote Sensing, Machine Learning and Image Processing have helped to automate this process and improve the results. Therefore, aiming to contribute to the state of the art in this topic, the goal of this project was to test many machine learning algorithms for classification and, later, segmentation of satellite images of the Amazon rainforest, seeking to develop a technology that can be used to monitor land cover changes in the region. For this purpose, four different classification algorithms were tested on variations of the same dataset, which differ in the application of a Gaussian filter in the preprocessing of the images. In the end, a part-based segmentation strategy was developed, which uses two models applied sequentially on an input image. One model for classifying forest and water pixels, and a second model for differentiating bare soil and agriculture pixels, achieving accuracy rates of 93,6% and 95,1%, respectively, using the Random Forest algorithm. Therefore, it was concluded that the implemented strategy had satisfactory performance and there are indications that the application of the Gaussian filter to the images can improve the obtained results.

Keywords: Machine Learning. Image Processing. Amazon Rainforest. Image segmentation.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

TP	Verdadeiro Positivo
FP	Falso Positivo
TN	Verdadeiro Negativo
FN	Falso Negativo
INPE	Instituto Nacional de Pesquisas Espaciais
CBERS	China-Brazil Earth Resources Satellite
MCC	Coeficiente de Correlação de Matthews
KNN	K-Nearest Neighbors
SVM	Support Vector Machines
CNN	Rede Neural Convolucional

SUMÁRIO

1 INTRODUÇÃO

A floresta tropical amazônica é um bioma de extrema importância no combate às alterações climáticas e ao aquecimento global, abrigando uma vasta biodiversidade, com milhares de plantas e animais, compondo o território de nove países sul americanos [4]. A Amazônia Legal, por sua vez, é uma região que engloba oito estados brasileiros (Acre, Amapá, Amazonas, Mato Grosso, Pará, Rondônia, Roraima e Tocantins) e parte do estado do Maranhão. Ela abrange 59% do território brasileiro (cerca de 5 milhões de km²) e representa 67% das florestas tropicais do mundo (2009) [5, 6].

Mesmo com esses fatos, dados do INPE (Instituto Nacional de Pesquisas Espaciais) indicam que o processo de desmatamento da região aumentou significativamente durante os anos de 1990 e início dos anos 2000, de forma que a organização World Wildlife Fund estima que, se o estado atual se mantiver, até 2030 mais de um quarto da floresta amazônica já terá sido devastada. A principal causa do desmatamento que assola esse bioma são atividades econômicas praticadas de forma não sustentável, como cultivo de soja, mineração ilegal, pecuária, extração de madeira e queimadas. Essas intervenções, por si, representam uma das maiores fontes de emissão de CO_2 provenientes da ação humana, grande responsável pela intensificação do processo de aquecimento global [7].

Para contornar essa situação, diversos projetos têm sido desenvolvidos nos últimos anos na área de sensoriamento remoto, dados os diversos avanços que atualmente facilitam a aquisição de grandes volumes de dados [8]. O INPE, por exemplo, possui programas como o PRODES (Programa de Monitoramento do Desflorestamento na Amazônia Legal) e o TerraClass, que auxiliam no monitoramento das transformações na cobertura do solo da região, mas que muito se baseiam na interpretação visual e trabalho manual de especialistas [9]. Então, com o intuito de diminuir a necessidade de intervenção humana e, consequentemente, otimizar o processo, muitos trabalhos têm sido desenvolvidos utilizando técnicas avançadas de Aprendizado de Máquina [7].

1.1 Objetivos

Dadas as questões apresentadas, percebe-se cada vez mais a importância da criação de mecanismos de proteção desse bioma e de como as novas tendências na área de Processamento de Imagem e Aprendizado de Máquina podem auxiliar na automatização desses processos. Dessa forma, este projeto teve como objetivo principal desenvolver um software automatizado de segmentação das imagens de satélite da floresta amazônica entre as regiões de florestas, agropecuária, corpos d'água e solo nu. Para isso, pretendeu-se atingir os seguintes objetivos:

- Estudo e aplicação de algoritmos de aprendizado de máquina na classificação de imagens de satélite da região amazônica
- Verificação dos resultados da aplicação do filtro gaussiano no pré-processamento das imagens, visando a eliminação de ruídos
- Determinação de qual(is) das estratégias de processamento e classificação de imagens testadas estão mais aptas a serem aplicados neste tipo de problema

1.2 Organização do trabalho

Este documento está organizado da seguinte forma:

- Capítulo 2 - Fundamentação Teórica: Apresenta os conhecimentos básicos envolvendo Processamento de Imagem e Aprendizado de Máquina para o desenvolvimento do projeto.
- Capítulo 3 - Procedimentos Metodológicos: Apresenta os passos realizados durante o desenvolvimento do projeto, juntamente das técnicas e algoritmos selecionados.
- Capítulo 4 - Resultados e análise: Apresenta uma análise dos resultados obtidos pelos algoritmos testados.
- Capítulo 5 - Conclusão: Apresenta as considerações finais acerca dos resultados alcançados pelo projeto, bem como possíveis indicações a serem seguidas em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo visa apresentar os principais conceitos teóricos e ferramentas utilizadas para o desenvolvimento do projeto, a maneira com que eles foram utilizados em trabalhos correlatos para a resolução de problemas similares e os resultados por eles alcançados.

2.1 Sensoriamento remoto da região amazônica

Sensoriamento remoto pode ser definido como:

> (...) uma ciência que visa o desenvolvimento da obtenção de imagens da superfície terrestre por meio da detecção e medição quantitativa das respostas das interações da radiação eletromagnética com os materiais terrestres [10].

As atividades que compõem essa área visam a definição das propriedades de objetos naturais e artificiais por meio da detecção, registro e análise da energia radiante emitida ou refletida pela superfície e, posteriormente, capturada por sensores [8].

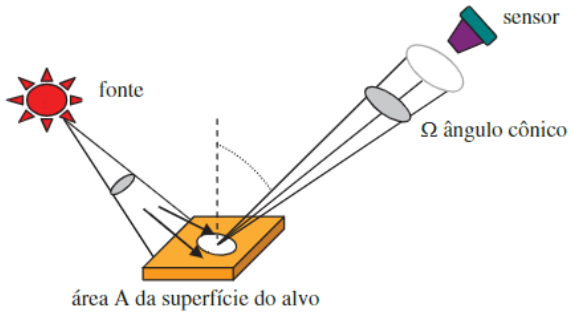

Figura 1 – Trajetória da radiação eletromagnética, da fonte ao sensor (Adaptado de: [1])

Para a realização deste trabalho foram selecionadas imagens captadas pelo sensor MUX (Câmera Multiespectral Regular) do satélite CBERS-4A, disponibilizadas pelo programa CBERS (China-Brazil Earth Resources Satellite) do INPE[1]. Cada imagem desse sensor possui quatro bandas, sendo elas: banda 5 (radiação azul), banda 6 (radiação verde), banda 7 (radiação vermelha) e banda 8 (infravermelho próximo) [11]. Na figura 2 são apresentadas as informações técnicas sobre os sensores imageadores do satélite em questão.

[1] <https://www.gov.br/inpe/pt-br/acesso-a-informacao/dados-abertos>

cbers.inpe.br/sobre/cameras/cbers04a.php

Sumário das características das Câmeras do CBERS 04A

Características das câmeras do CBERS 04A

Característica	WPM	MUX	WFI
Bandas Espectrais	0,45-0,52µm (B) 0,52-0,59µm (G) 0,63-0,69µm (R) 0,77-0,89µm (NIR) 0,45-0,90 µm (PAN)	0,45-0,52µm (B) 0,52-0,59µm (G) 0,63-0,69µm (R) 0,77-0,89µm (NIR)	0,45-0,52µm (B) 0,52-0,59µm (G) 0,63-0,69µm (R) 0,77-0,89µm (NIR)
Resolução	2 m 8 m	16,5 m	55 m
Largura da Faixa Imageada	92 km	95 km	684 km
Visada Lateral de Espelho	não	não	não
Revisita	31 dias	31 dias	5 dias
Quantização	10 bits	8 bits	10 bits
Taxa de Dados Bruta	1800.8 Mbps 450.2 Mbps	65 Mbps	50 Mbps

Figura 2 – Informações técnicas sobre os sensores do satélite CBERS-4A

Para este estudo foi escolhida uma área situada na fronteira agrícola de Rondônia, região com intenso processo de desmatamento, considerando que o percentual de área desmatada do estado chega a cerca de 28,5% (2009) [5]. O formato das imagens captadas é quadrilátero de aproximadamente 3,5 km de lado e situa-se entre os municípios de Machadinho D'Oeste e de Vale do Anari, a nordeste do Estado de Rondônia, próximo às divisas dos estados de Mato Grosso e Amazonas, como pode ser visualizado na figura 3.

Figura 3 – Localização da área de estudo (Fonte: Adaptado do Google Earth)

2.2 Processamento de imagem

A área de processamento de imagem pode ser definida como o estudo das operações que envolvem a criação de uma imagem resultante a partir de uma imagem de entrada. São as técnicas utilizadas para melhorar características de imagens, removendo ruídos e distorções, extrair elementos desejados, entre outras operações.

No meio digital, uma imagem é representada como uma matriz bidimensional, onde cada um dos seus elementos é chamado de pixel, que, por sua vez, armazena um valor de intensidade de luz associado aquela posição da imagem [12]. Para a representação das intensidades de cor das imagens, existem diversos modelos com diferentes aplicações; alguns deles são: RGB, HSV e LAB.

2.2.1 Modelo de cor RGB

O modelo de cor RGB é formado pela composição das cores primárias R (vermelho), G (verde) e B (azul), onde cada uma possui um valor de intensidade que varia de 0 a 1, ou de 0 a 255 em um sistema de 8 bits por camada. O branco é alcançado por meio da composição do valor máximo de cada cor (1 ou 255), enquanto o preto é obtido pela composição do valor mínimo de cada cor (0). Este modelo é comumente utilizado em equipamentos eletrônicos como televisores e monitores e a quantidade de valores que ele consegue representar é suficiente para cobrir a maioria das cores distinguíveis pelo ser humano [2].

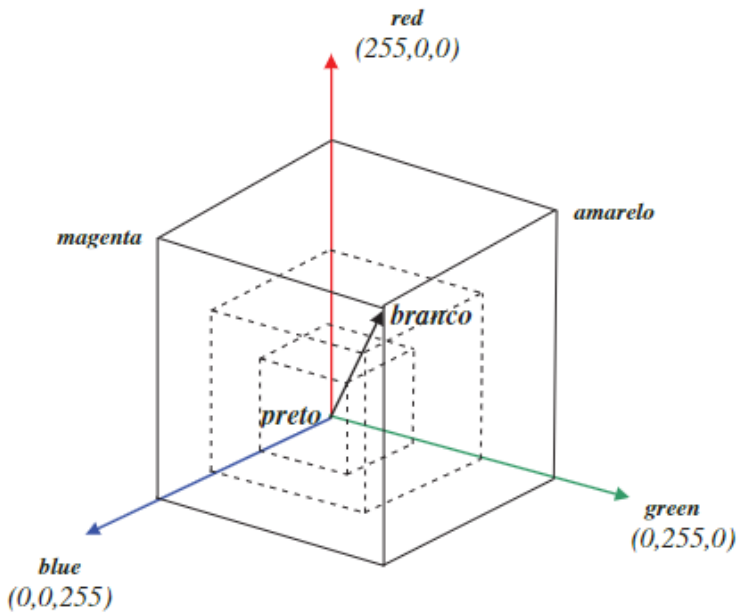

Figura 4 – Intervalo de intensidades do modelo RGB representado em um plano de três dimensões (Fonte: [1])

2.2.2 Modelo de cor HSV

O modelo HSV, diferentemente do RGB, é mais intuitivo, sendo constituído dos seguintes componentes:

- *Hue* (Matiz): Representa a cor pura como um ângulo, como pode ser visualizado na figura 5, e varia entre os valores de 0º a 360º
- *Saturation* (Saturação): Representa a quantidade de luz branca presente na cor e atinge valores de 0 (totalmente branca) a 1 (cor pura)
- *Value* (Valor): Representa a intensidade da cor e atinge valores de 0 (intensidade nula) a 1 (intensidade máxima)

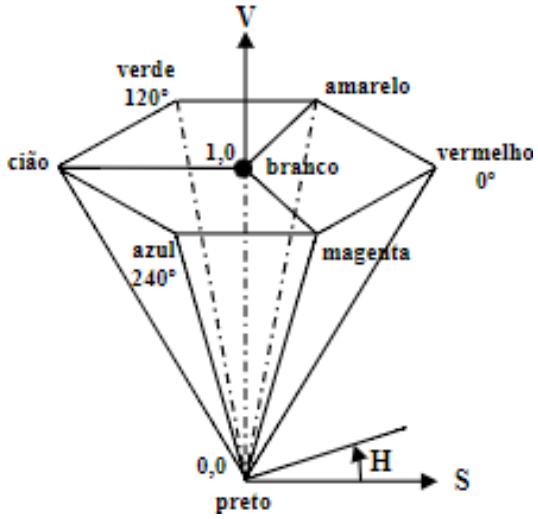

Figura 5 – Representação do espaço de cor HSV (Fonte: [2])

2.2.3 Modelo de cor LAB

O modelo de cor LAB é considerado um modelo *device-independent*, o que significa que as cores por ele representadas não são afetadas pelo aparelho sendo utilizado e, devido a isso, ele é comumente utilizado em transmissões via internet que precisam trafegar por diferentes dispositivos. Nesse modelo as cores são determinadas com base na sua posição em um espaço de cor de três dimensões [3], como pode ser visualizado na figura 6, definido pelos seguintes componentes:

- L: representa a intensidade de luz e atinge valores de 0 (cor preta) a 100 (cor branca)
- A: componente de cor que varia entre valores de verde a vermelho
- B: componente de cor que varia entre valores de azul a amarelo

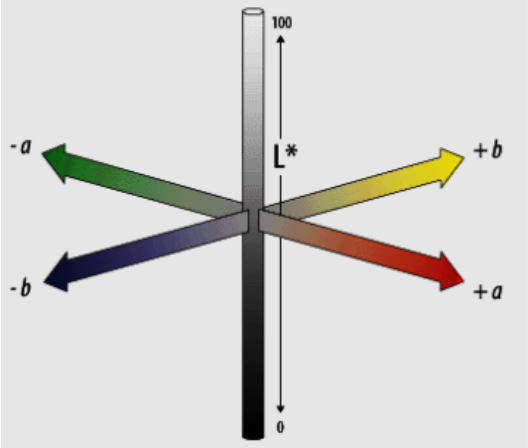

Figura 6 – Representação do espaço de cor LAB (Fonte: [3])

2.2.4 Filtragem espacial de imagens

As técnicas de filtragem espacial são normalmente utilizadas no pré-processamento de imagens, visando a remoção de ruídos, realce de regiões desejadas, entre outras operações que auxiliem etapas seguintes de processamento. Essas transformações são baseadas no deslocamento de uma máscara (*kernel*), com número ímpar de linhas e colunas, em cada um dos pixels da imagem, como pode ser visualizado na figura 7. A máscara é posicionada de forma que o elemento central seja o pixel sendo analisado e o cálculo do seu valor na imagem resultante é obtido com base tanto no seu valor de intensidade quanto no de seus vizinhos [12].

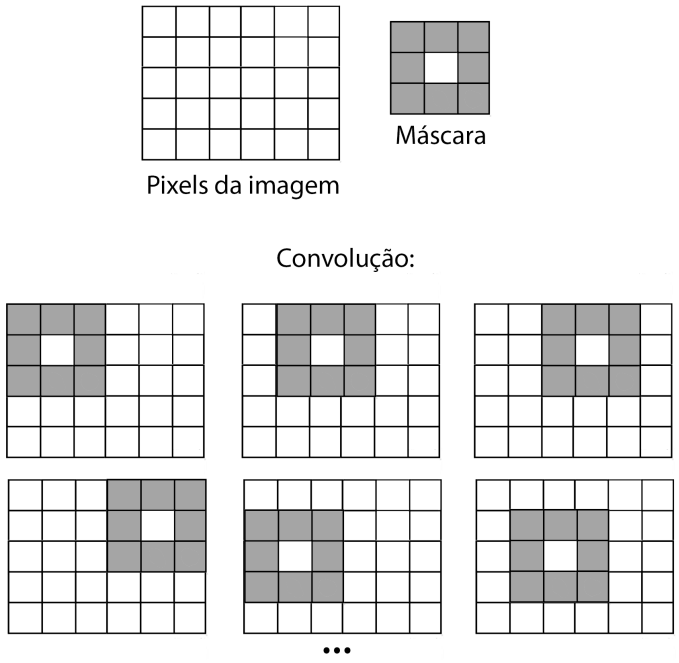

Figura 7 – Ilustração do processo de convolução realizado na aplicação de um filtro de máscara 3x3 (Fonte: O autor)

Existem diversos tipos de filtros, que se diferenciam no cálculo dos valores dos pixels, bem como na finalidade para a qual são aplicados. O filtro gaussiano, por exemplo, pode ser definido como uma aproximação da função gaussiana e é usado para eliminar ruídos indesejáveis. Ruídos podem ser definidos como pixels aleatórios com valores de intensidade muito distantes dos seus vizinhos que, devido a isso, prejudicam a qualidade da imagem e a interpretação dos dados.

A função gaussiana é definida pela seguinte equação:

$$f(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{2.1}$$

Onde:

- σ: desvio padrão

Porém, a sua aplicação como filtro espacial exige uma aproximação para o formato matricial, representado como uma máscara de convolução. Um exemplo de aproximação com $\sigma = 1$ e máscara de tamanho 3x3 pode ser visualizado na figura 8.

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 8 – Máscara de convolução do filtro gaussiano de tamanho 3x3 e $\sigma = 1$ (Fonte: O autor)

Neste trabalho, utilizou-se do filtro gaussiano como técnica de pré-processamento das imagens, visando a aplicação da operação antes do treinamento com os algoritmos de aprendizado de máquina. O intuito de sua utilização foi de contornar um problema enfrentado na classificação pixel a pixel de imagens de satélite: a grande presença de ruídos. Por meio disso, foi proposto a interpretação de cada pixel das imagens como representante de uma região ao seu redor e não apenas de sua área delimitada.

2.2.5 Segmentação de imagem

Segmentação é o processo de separar uma imagem entre grupos de pixels similares [13] ou de subdivisão da imagem em partes ou objetos. Os algoritmos de segmentação para

imagens monocromáticas se baseiam na descontinuidade, dividindo a imagem com base na alteração brusca dos níveis de cor (presente em bordas, pontos isolados e linhas), ou na similaridade, baseada na limiarização e no crescimento de regiões [1]. Na segmentação semântica, foco deste trabalho, o objetivo é classificar cada um dos pixels presentes em uma imagem.

2.3 Descritores de imagem

Descritores de imagem são características que podem ser percebidas nos objetos pertencentes à mesma. Essas características são utilizadas para agrupar objetos com características semelhantes em uma mesma classe [1]. Existe uma gama muito grande de descritores que podem ser obtidos, podendo ser descritores de forma, cor, textura, etc [14], escolhidos com base no tipo de problema a ser resolvido.

Em trabalhos relacionados à classificação de imagens de satélite da floresta amazônica os descritores mais utilizados têm sido as bandas R (vermelho), G (verde), B (azul), NIR (infravermelho próximo) e infravermelho de ondas curtas [4, 9].

2.4 Aprendizado de Máquina

Aprendizado de máquina é um ramo da Inteligência Artificial focado no desenvolvimento de algoritmos que simulam o aprendizado humano. Esses modelos aprendem, por meio de treinamento com um conjunto de dados, a tomar decisões, identificar padrões e fazer predições [14]. Um modelo de classificação, por exemplo, após ser treinado, deve ser capaz de receber um conjunto de amostras de um problema e atribuir uma classe ou rótulo para cada uma delas.

O aprendizado de um modelo deste tipo consiste no processo que o próprio computador realiza de identificar os padrões e as relações entre os dados que o auxiliam a entender o problema e, posteriormente, a realizar predições acerca do mesmo. Existem, basicamente, três formas de realizar esse aprendizado; são elas: Aprendizado Supervisionado, Aprendizado Não-Supervisionado e Aprendizado por Reforço.

No Aprendizado Supervisionado, o mais comumente utilizado, o modelo é treinado recebendo um conjunto de instâncias representadas por um par entrada-saída, onde a entrada são os atributos que a representam e a saída é a classe que ela pertence. Em áreas como classificação de imagem, essas amostras são escolhidas por um especialista no problema em questão [8] e, a partir delas, o modelo deverá ser capaz de aprender uma função que mapeia um conjunto de dados de entrada para uma saída [13].

Existem diversos algoritmos de aprendizado de máquina já estabelecidos na literatura, mas para a resolução de problemas semelhantes ao apresentado neste trabalho há

um destaque para a utilização de Redes Neurais Convolucionais (CNN), além de algoritmos de classificação mais clássicos, como *Support Vector Machine* (SVM) e algoritmos de árvore de decisão.

Neste projeto, pretende-se a criação de um modelo de classificação pixel a pixel, onde cada uma das instâncias usadas para treinamento e teste será um pixel da imagem e a tarefa de classificação será atribuir uma classe para cada um desses pixels.

2.4.1 *Random Forest*

O algoritmo *Random Forest* é considerado um método do tipo *ensemble*, cujo intuito é combinar as predições de vários modelos diferentes, de forma a construir um mecanismo de predição mais robusto. Nesse algoritmo, cada um dos modelos é uma árvore de decisão construída a partir da seleção de um conjunto aleatório de amostras, definido utilizando a técnica de *bootstrap*, que permite a escolha de instâncias repetidas.

Além da aleatoriedade na seleção das amostras, na construção de cada árvore de decisão os melhores pontos de *split* são encontrados a partir da avaliação de um subconjunto aleatório dos atributos de entrada. Dessa maneira, realizando a média das predições de cada árvore, esse modelo consegue contornar os problemas de alta variância e *overfitting* comuns de ocorrer em uma simples árvore de decisão [15].

2.4.2 *K-Nearest Neighbors*

O *K-Nearest Neighbors*, diferentemente de outros algoritmos de aprendizado de máquina, não visa construir modelo de generalização, funcionando, basicamente, por meio do armazenamento dos dados de treinamento. Dessa forma, a classificação de uma instância é feita com base no seu posicionamento no plano que armazena os dados de treinamento e na análise dos seus k vizinhos mais próximos, de forma que, se eles pertencerem, em sua maioria, a uma classe x, a instância em questão será classificada como sendo da classe x [16, 15].

Na figura 9 podemos visualizar o processo de classificação de uma instância (representada por um ponto amarelo) por um modelo treinado para diferenciar duas classes (representadas em azul e vermelho), utilizando $k = 3$.

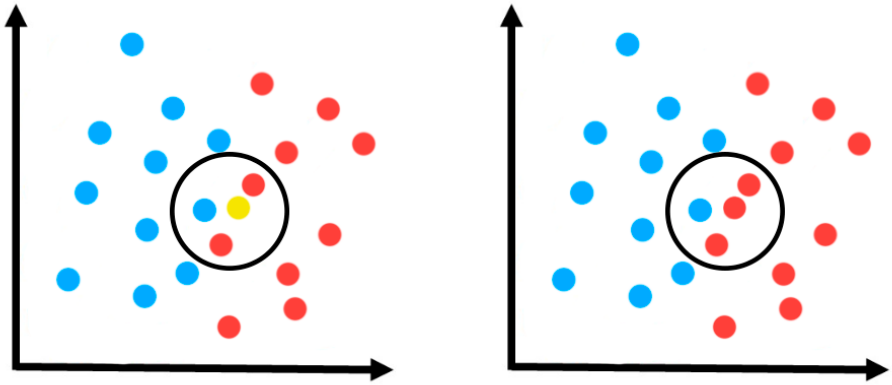

Figura 9 – Representação do processo de classificação de uma instância pelo KNN utilizando $k = 3$ (Fonte: O autor)

2.4.3 *Support Vector Machine*

Support Vector Machine (SVM), ou Máquina de Vetores de Suporte, é um algoritmo que visa encontrar uma curva ou fronteira de decisão que melhor diferencie as classes em uma base de dados definida sobre um plano n-dimensional. Os vetores de suporte são os elementos pertencentes à borda da margem da curva, como pode ser visualizado na figura 10 [15].

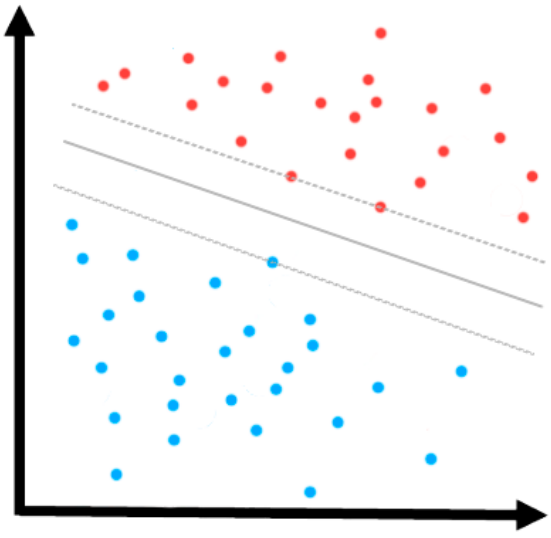

Figura 10 – Exemplo de uma fronteira de decisão em um plano construído pelo SVM (Fonte: O autor)

2.4.4 *Naive Bayes*

O algoritmo de classificação *Naive Bayes* se baseia na suposição "ingênua" (*naive*) de que os atributos do problema são independentes entre si; dessa maneira, as correlações

entre eles são ignoradas de forma que um problema de múltiplas variáveis se torne um problema de apenas uma variável [17].

2.5 Seleção de atributos

A seleção de atributos é tipo de pré-processamento que envolve selecionar um subconjunto dos atributos mais relevantes para a resolução do problema. Ela é utilizada com o objetivo de melhorar o desempenho das predições, reduzindo o tempo computacional gasto e fornecendo um melhor entendimento acerca do problema [18].

Um dos motivos para sua utilização é a eliminação de atributos redundantes e irrelevantes. Atributos redundantes são aqueles que compartilham a mesma informação com um ou mais atributos e, portanto, apenas um deles se torna suficiente. Já os atributos irrelevantes são aqueles que não possuem relação com o resultado que o modelo quer atingir, servindo apenas para aumentar o gasto com tempo de processamento e introduzir ruídos no processo de classificação [18].

Existem vários métodos para a realização dessa operação, e eles podem ser divididos em: métodos *Filter*, métodos *Wrapper* e métodos embutidos.

2.5.1 Métodos *Filter*

Os métodos do tipo *Filter* usam técnicas estatísticas para avaliar a relação entre cada atributo e a classe do problema, atribuindo uma pontuação para cada um deles, que, posteriormente, será utilizada para filtrar aqueles que serão utilizados no modelo. Devido a essa característica, esses métodos tendem a ser mais rápidos que os métodos *Wrapper*, sendo bastante úteis para conjuntos de dados muito grandes.

Comumente utiliza-se como base para esse tipo de técnica métricas estatísticas de correlação entre as variáveis de saída e entrada. Devido a isso, a escolha das métricas depende bastante do tipo da variável de entrada e de saída [19]. Dentre os métodos desse tipo estão: o Coeficiente de Correlação de Pearson, a Informação Mútua, entre outros.

Com o Coeficiente de Correlação de Pearson é possível avaliar a relação linear entre duas variáveis. Se a correlação for próxima de -1 (correlação negativa) ou de 1 (correlação positiva) as variáveis são fortemente correlacionadas. Se o seu resultado for 0, as variáveis não possuem correlação. Como forma de interpretação dos valores desse coeficiente, podemos adotar as seguintes regras [20]:

1. Muito forte: 0,9 - 1
2. Forte: 0,7 - 0,9
3. Moderada: 0,5 - 0,7

4. Fraca: 0,3 - 0,5
5. Correlação pequena ou inexistente: 0,3 - 0

Já a Informação Mútua procura medir a dependência entre duas variáveis, ou seja, o quanto de informação que pode ser obtida por uma variável aleatória ao observar outra variável aleatória. Dessa forma, é possível calcular a informação mútua entre cada atributo e a classe do problema, auxiliando na determinação dos atributos irrelevantes ou redundantes para o problema.

2.5.2 Métodos *Wrapper*

Os métodos *Wrapper* se baseiam na criação de diversos modelos preditivos com diferentes subconjuntos de atributos. Cada um desses modelos é testado e, ao final, é gerado um ranking dos subconjuntos de atributos de acordo com seu desempenho em alguma métrica.

Esse tipo de método normalmente gera melhores resultados que os métodos *Filter*, pois consegue avaliar a importância de todos os atributos em conjunto, além de testar diferentes combinações dos mesmos. Por outro lado, como se faz necessário o teste de desempenho de diferentes subconjuntos de atributos, esses métodos tendem a ser bastante custosos para uma quantidade de dados muito grande.

2.5.3 Métodos embutidos

Métodos embutidos são algoritmos de aprendizado de máquina que executam a seleção de atributos automaticamente, por exemplo, os algoritmos de árvore de decisão, como o *Random Forest*. Neste caso, a seleção de atributos é incorporada no processo de treinamento do modelo, reduzindo o custo computacional gerado pelo treinamento de diferentes subconjuntos, como é feito nos métodos *Wrapper* [18].

2.6 Métricas de avaliação de modelos

Métricas são medidas utilizadas para avaliar o resultado de algoritmos de aprendizado de máquina, com o intuito de auxiliar a mensurar o quanto aquele modelo se adequa ao problema em questão.

A forma mais básica de se avaliar um modelo de classificação é pela Matriz de Confusão. Por meio dela, é possível visualizar as taxas de acerto e erro do modelo de classificação. Em um problema de classificação binária (classe positiva ou negativa) a matriz de confusão é construída como apresentado na tabela 1, onde cada entrada na matriz representa uma porção das previsões realizadas, sendo elas:

- **Verdadeiro Positivo (TP):** Quantidade de previsões corretas para instâncias positivas
- **Verdadeiro Negativo (TN):** Quantidade de previsões corretas para instâncias negativas
- **Falso Positivo (FP):** Quantidade de instâncias negativas erroneamente classificadas como positivas
- **Falso Negativo (FN):** Quantidade de instâncias positivas erroneamente classificadas como negativas

Tabela 1 – Matriz de confusão

	Positivo Predito	Negativo Predito
Positivo Real	Verdadeiro Positivo (TP)	Falso Negativo (FN)
Negativo Real	Falso Positivo (FP)	Verdadeiro Negativo (TN)

Dessa forma, a matriz de confusão representa uma maneira simples, porém nem sempre suficiente, de visualizar o quão bom um modelo é na resolução de determinado problema. Para avaliar uma solução de maneira satisfatória, se faz necessário trabalhar com métricas de avaliação mais completas, como Acurácia, Precisão, Sensibilidade, F-score, Estatística Kappa, Curva ROC, entre outras, que em sua maioria derivam dos resultados apresentados na matriz de confusão.

2.6.1 Acurácia

A acurácia mede quantas instâncias foram classificadas corretamente, independente da classe, por meio da razão entre o que o modelo acertou e a quantidade de instâncias classificadas.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

Uma desvantagem dessa métrica é que ela gera um resultado inadequado quando estamos tratando de um problema com classes desbalanceadas, ou seja, onde existem mais instâncias de uma classe do que de outra. Além disso, ela também atribui peso igual para ambos os erros, algo que pode não ser muito interessante em determinados problemas.

2.6.2 Precisão

A precisão é a quantidade de instâncias classificadas corretamente como positivas dividida pelo total de instâncias classificadas como positivas.

$$P = \frac{TP}{TP + FP} \tag{2.3}$$

Ela é usada quando é preciso uma confiança muito grande de que os casos positivos são verdadeiramente positivos. Por exemplo, quando o caso positivo representa a detecção de um e-mail como *spam* é imprescindível que o número de e-mails normais classificados erroneamente como *spam* seja mínimo (garantir um FP pequeno).

2.6.3 Sensibilidade

A sensibilidade é a quantidade de instâncias classificadas corretamente como positivas dividida pelo total de instâncias verdadeiramente positivas.

$$R = \frac{TP}{TP + FN} \tag{2.4}$$

É usada quando a ocorrência de falsos negativos é "inaceitável". Por exemplo, em um problema de detecção de uma doença não se deve, em hipótese alguma, classificar um paciente doente (caso positivo) como saudável (garantir um FN pequeno).

2.6.4 F-score

O F-score é calculada pela média harmônica entre a Precisão e a Sensibilidade:

$$F = 2 * \frac{P * R}{P + R} \tag{2.5}$$

onde

- P: Precisão
- R: Sensibilidade

Dessa forma, essa é uma boa medida caso deseja-se maximizar os resultados de ambas as métricas associadas, já que a média harmônica implica que o resultado da F-score será bom apenas caso ambos os valores de Precisão e Sensibilidade sejam bons.

2.6.5 Coeficiente Kappa

O coeficiente kappa avalia o nível de concordância entre o real e o predito em uma tarefa de classificação e indica o quanto as observações se afastam daquelas esperadas.

$$k = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{2.6}$$

onde

- p_0: porcentagem de concordância observada
- p_e: porcentagem de concordância esperada

  e

- $k = 1$: indica concordância perfeita
- $k = 0$: indica não haver concordância

2.6.6 Curva ROC

Um modelo de classificação pode, no processo de classificação, atribuir uma probabilidade de uma instância pertencer a diferentes classes. Numa classificação binária, por exemplo, quanto mais próxima essa probabilidade for de 0, é provável que essa instância pertença à "classe negativa", da mesma forma, quanto mais próxima essa probabilidade estiver de 1, é provável que essa instância pertença à "classe positiva".

Porém, para realizar a divisão entre as classes é preciso definir um limiar, que nada mais é que um valor numérico. Com um limiar igual a 0,5, por exemplo, todas as instâncias com probabilidade maior ou igual a 0,5 serão classificadas como positivas e todas aquelas com probabilidade menor que 0,5 serão classificadas como negativas.

Utilizando desse conceito, a métrica de avaliação da curva ROC visa demonstrar o quão bem um modelo é em distinguir entre duas classes, se baseando no cálculo da Taxa de Verdadeiro Positivo (TPR) (ou Sensibilidade) e na Taxa de Falso Positivo (FPR) para diferentes valores de limiar. Para tal, é construído um gráfico, com o TPR traçado no eixo y e o FPR traçado no eixo x, onde cada ponto no gráfico representa o desempenho do classificador com determinado limiar.

$$TPR = \frac{TP}{TP + FN} \qquad (2.7) \qquad\qquad FPR = \frac{FP}{FP + TN} \qquad (2.8)$$

Para simplificar a sua interpretação e auxiliar na escolha do melhor modelo existe a medida da Área sob a Curva ROC (AUC), que consiste no cálculo da área abaixo da curva ROC. Quanto maior for essa área, melhor é o modelo na tarefa de distinguir entre as classes. Para interpretar seu resultado, podemos seguir as seguintes indicações:

- AUC próximo de 1: indica que o modelo foi capaz de separar as classes
- AUC próximo de 0,5: indica que o modelo não foi capaz de separar as classes
- AUC próximo de 0: indica que o modelo inverteu a classificação de todas as instâncias

2.6.7 MCC

O coeficiente de correlação de Mathews também mede a correlação entre o real e o predito. Uma vantagem de sua utilização é que ele é simétrico, de forma que nenhuma classe é mais importante que a outra.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \tag{2.9}$$

onde

- $FP = FN = 0 \Rightarrow MCC = 1$: indica concordância perfeita
- $MCC = 0$: indica que a predição é tão boa quanto uma aleatória
- $TP = TN = 0 \Rightarrow MCC = -1$: indica correlação negativa perfeita, ou seja, total discordância entre o real e o predito

2.7 Técnicas de avaliação de modelos

Quando trabalhamos na criação de um modelo preditivo de aprendizado de máquina, uma das maiores dificuldades está em garantir que ele seja confiável para lidar com novos dados (sejam dados de teste ou de entrada em um sistema já modelado) [21], pois é importante que, ao ser treinado, ele não sofra de um problema chamado *overfitting*, que ocorre quando o modelo aprende tão bem com a base de dados que acaba decorando as relações nela presentes (incluindo os possíveis ruídos), perdendo sua capacidade de generalização.

Para garantir essa confiabilidade é necessária a utilização de técnicas para organização da base de dados no processo de treinamento e teste do modelo. Existem diversas técnicas para esse intuito, mas o foco aqui será discutir sobre duas: a divisão percentual da base de dados e a validação cruzada. Essas técnicas se diferem em como a base de dados disponível é dividida entre treinamento e teste.

Na divisão percentual, a base de dados é dividida, aleatoriamente, em uma parte para treinamento e outra para o teste. Já na validação cruzada, a base de dados é dividida uma única vez em k pedaços (*folds*) e o processo de treinamento e teste é repetido k vezes. Sendo que em cada uma das execuções é escolhida um dos pedaços para teste e o restante é usado para o treinamento. Ao final, o resultado é obtido por meio do cálculo da média de todas as execuções.

A validação cruzada é uma técnica bastante utilizada pois gera uma boa confiabilidade nos resultados, ao garantir que os conjuntos de teste e treino sejam sempre distintos

e que variações diferentes desses conjuntos sejam testadas, dando uma maior noção acerca da capacidade do modelo de lidar com diferentes situações. Por outro lado, quando a base de dados utilizada é muito grande, o processo pode levar muito tempo de processamento e, nesse caso, utilizar a divisão percentual pode ser mais interessante, tendo em vista que uma base de dados enorme e com bastante informação pode ser suficiente para auxiliar na construção de um modelo confiável.

2.8 Weka

O Weka (*Waikato Environment for Knowledge Analysis*) é um *software* desenvolvido em Java pela Universidade de Waikato, Nova Zelândia. Ele permite a aplicação de métodos de aprendizado de máquina para mineração de dados, contendo ferramentas para lidar com problemas de classificação, regressão, entre outros. No contexto deste trabalho, o *software* foi utilizado para o teste de diversos algoritmos de classificação sobre conjunto de dados coletados das imagens.

Para o processo de aplicação e comparação dos algoritmos foi utilizado a aba *Experimenter* do *software*, que permite testar diversos algoritmos de aprendizado de máquina em sequência e, ao final, fazer uma comparação entre os resultados de cada algoritmo, tendo como base diversas métricas de avaliação.

2.9 Scikit-learn

Scikit-learn[2] é uma biblioteca da linguagem Python que oferece diversas ferramentas para a criação de modelos de aprendizado de máquina para resolução de problemas de classificação, regressão e outros. Neste projeto, essa biblioteca, juntamente de outras ferramentas presentes no Python, foram utilizadas para a aplicação dos algoritmos de classificação.

2.10 Trabalhos correlatos

Como citado na seção 2.4, nos últimos anos, as CNNs têm ganhado destaque na classificação de imagens por proverem bons resultados e não necessitarem que o programador realize uma seleção prévia dos atributos, já que esse tipo de algoritmo possui a capacidade de aprender a extraí-los automaticamente [4].

Em [4] foi utilizado um modelo de CNN chamado VGG16 para a classificação de imagens de satélite, utilizando as bandas vermelho, verde, azul e infravermelho próximo. Neste trabalho não foi realizada a segmentação das imagens, apenas a classificação das mesmas em múltiplas classes, com base nos tipos de cobertura presentes em cada recorte.

[2] <https://scikit-learn.org/stable/>

Os resultados, porém, foram satisfatórios, sendo obtida uma acurácia de 96.71% e um *F-Beta Score* de 92.69%.

Já em [9], o objetivo foi utilizar a técnica de imagem mosaico e de uma CNN para a classificação e segmentação de regiões da Amazônia em três tipos de cobertura: agricultura, pasto e floresta. As imagens foram capturadas do sensor LANDSAT-8/OLI., disponíveis no site da USGS [3] e as bandas utilizadas foram Vermelho (B4), Infravermelho próximo (B5) e Infravermelho de ondas curtas (B6). As imagens-mosaico citadas foram criadas por meio da junção de retalhos de imagem de cada uma das coberturas de solo. Dentre os modelos testados, o que apresentou melhor desempenho foi o que utilizou, somado à CNN, do método de otimização RMSProp, obtendo acurácia global de 97.467%.

No trabalho [8] foi desenvolvida uma metodologia de Classificação Supervisionada de Imagens de Satélite (ClasSIS) para a criação de mapas temáticos do uso e cobertura de terra em assentamentos localizados na Amazônia. As imagens foram obtidas pelo sensor TM do satélite Landsat 5 e selecionadas a partir do banco de imagens do INPE[4]. Foi utilizado o método de classificação *pixel* a *pixel* e os algoritmos testados foram: *Random Forest*, Máxima Verossimilhança (MAXVER), Máquina de Vetores de Suporte (SVM), *Classification And Regression Trees* (CART) e uma Rede Neural Artificial. O modelo de melhor desempenho foi o *Random Forest*, com uma acurácia de 98% e um Índice Kappa de 0.975; porém, concluiu-se que o algoritmo CART também obteve um desempenho semelhante.

[3] <https://earthexplorer.usgs.gov>
[4] <http://www.dgi.inpe.br/CDSR/>

3 PROCEDIMENTOS METODOLÓGICOS

Este capítulo descreve os procedimentos metodológicos necessários para o cumprimento dos objetivos deste trabalho. As etapas realizadas estão ilustradas na figura 11.

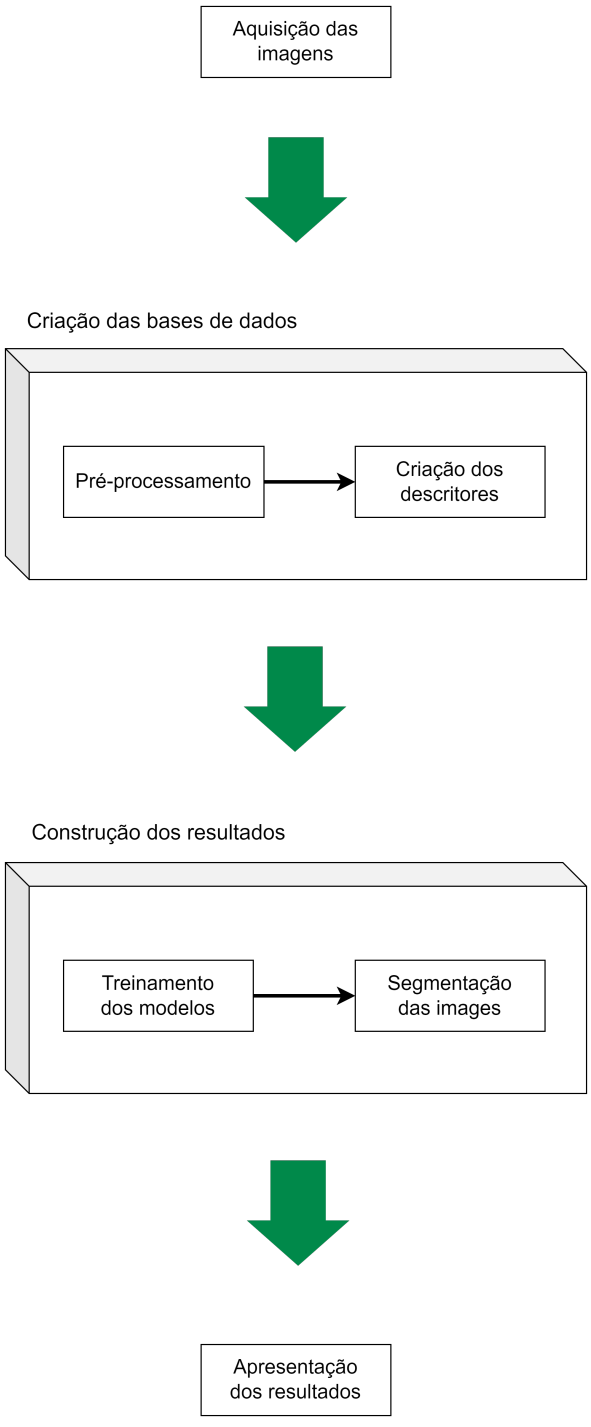

Figura 11 – Metodologia utilizada no desenvolvimento do trabalho

3.1 Aquisição das imagens

As imagens foram selecionadas pelo Prof. Osvaldo Coelho Pereira Neto do Departamento de Geologia e Geomática do CCE (UEL), sendo recortes da mesma localização nas seguintes datas: 13/06/2021, 14/07/2021, 20/05/2022, 20/06/2022, 21/07/2022 e 21/08/2022. As imagens são referentes aos meses de inverno, onde há menor predominância de nuvens. Cada uma das imagens possui formato .tif e são formadas pelas bandas espectrais do Verde, Vermelho e Infravermelho Próximo (NIR). Além disso, a resolução das imagens é de 16,5 metros.

Após a seleção, foi extraído um recorte menor, de aproximadamente 227 pixels de largura e 244 pixels de altura, de cada uma das imagens selecionadas. Após selecionados, os recortes foram classificados manualmente com a supervisão do professor Osvaldo para serem utilizados no aprendizado supervisionado de algoritmos de aprendizado de máquina. Na figura 12 é possível visualizar o recorte do dia 21/08/2022 e a sua classificação manual.

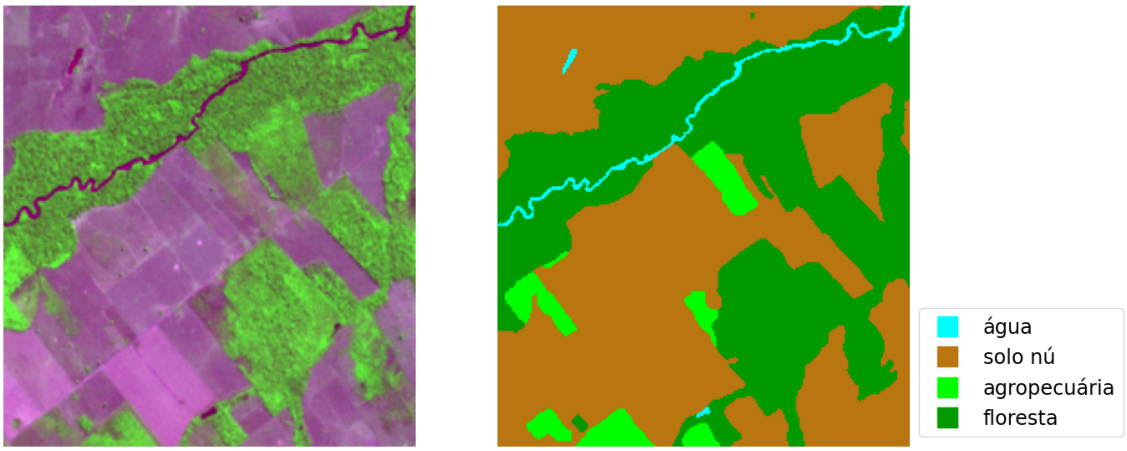

Figura 12 – À esquerda o recorte da imagem original do dia 21/08/2022 e à direita a sua classificação manual

3.2 Pré-processamento das imagens

Com o objetivo de testar os resultados da aplicação do filtro gaussiano nas imagens, foram selecionadas as máscaras de tamanho 3, 5, 7, 9 e 11 a serem utilizadas na criação de novas versões das imagens originais. Para a realização desse processo foi desenvolvido o seguinte algoritmo:

1. São carregadas na memória cada uma das imagens originais
2. Para cada uma das imagens é criada uma nova versão com a aplicação dos filtros:
    a) Gaussiano com máscara de tamanho 3x3

b) Gaussiano com máscara de tamanho 5x5

c) Gaussiano com máscara de tamanho 7x7

d) Gaussiano com máscara de tamanho 9x9

e) Gaussiano com máscara de tamanho 11x11

Desse modo, foram definidos seis conjuntos de imagens distintos, sendo eles: o conjunto de imagens "original" e os conjuntos de imagens "filtradas_3x3", "filtradas_5x5", "filtradas_7x7", "filtradas_9x9" e "filtradas_11x11",

3.3 Criação das bases de dados

A partir do que foi definido na seção 3.2 foi feita a criação de seis bases de dados distintas, uma para cada um dos conjuntos de imagens criados. Para tal, foi desenvolvido um algoritmo para a leitura das imagens de um conjunto e criação de um arquivo do tipo ".csv" (*Comma-separated values*) contendo os dados das imagens.

Dessa maneira, com o auxílio da linguagem Python e a biblioteca OpenCV foram executados, para cada conjunto de imagens, os seguintes passos:

1. São carregadas na memória as imagens manualmente classificadas e suas respectivas versões naquele conjunto de imagens
2. Para os pixels de cada imagem:
   a) São armazenadas as bandas B (espectro_verde), G (espectro_nir) e R (espectro_vermelho) como atributos na base de dados
   b) Com base nos valores das bandas BGR são criados e armazenados como atributos os seguintes descritores: R-G, R-B, G-B, H, S, V, L, A, B, NDVI e dist_euclidiana(BGR). Sendo o NDVI calculado pela seguinte equação:

$$NDVI = \frac{G(espectro\_nir) - R(espectro\_vermelho)}{G(espectro\_nir) + R(espectro\_vermelho)} \tag{3.1}$$

   Já o último descritor é calculado por meio da distância entre o pixel sendo analisado e o pixel centroide (média dos canais RGB) da imagem
   c) É definida a sua classe com base nas cores RGB do mesmo pixel na imagem manualmente classificada
3. Os dados coletados são então utilizados para a criação do arquivo .csv

Buscando manter uma imagem apenas para a avaliação dos resultados dos modelos, o recorte do dia 13/06/2021 foi excluído do processo de treinamento e utilizado apenas na validação dos modelos, por meio de uma avaliação visual do resultado da sua classificação.

Ao final, cada uma das bases de dados ficou composta por 279.350 instâncias, divididas entre as classes: “solo_nu”, “agropecuaria”, “floresta”, “agua”, como apresentado na figura 13.

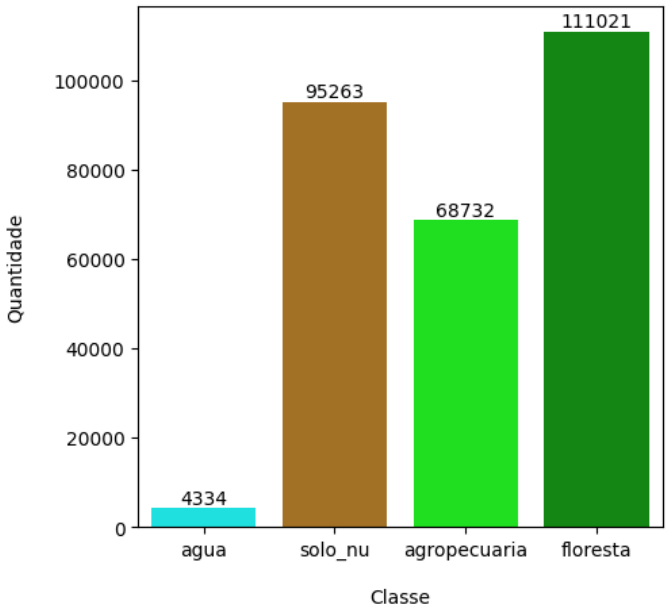

Figura 13 – Distribuição das instâncias entre as classes

3.4 Seleção de atributos

Para realizar a seleção de atributos utilizou-se a base de dados original como referência e os métodos do tipo *Filter* Coeficiente de Correlação de Pearson e Informação Mútua, devido à quantidade grande de atributos na base de dados, que tornaria a utilização dos métodos do tipo *Wrapper* muito custosa. Na tabela 2, que exibe a correlação entre os atributos, foi possível visualizar que diversos deles estavam bastante correlacionados, indicando que alguns poderiam ser removidos visando diminuir o tempo de processamento e melhorar os resultados dos algoritmos, como explicado na seção 2.5. Além disso, também foi calculada o valor de informação mútua de cada atributo com relação à classe do problema e o ranking gerado pode ser visualizado na tabela 3.

A partir dessas informações, buscou-se remover os atributos bastante correlacionados com a seguinte estratégia: Tomar como base um dos quatro atributos com maior índice de informação mútua e remover aqueles que possuírem correlação forte (acima de 70%) com o mesmo. Então, seguindo a ordem do ranking de informação mútua, primeiramente analisou-se o atributo dist_euclidiana(BGR), que foi descartado por não possuir correlação forte com nenhum outro, não sendo, portanto, interessante para o objetivo desta etapa. Restando os atributos espectro_vermelho(R), NDVI e espectro_verde(B) a escolha foi feita com base na correlação com a classe do problema, que levou a seleção da

variável NDVI.

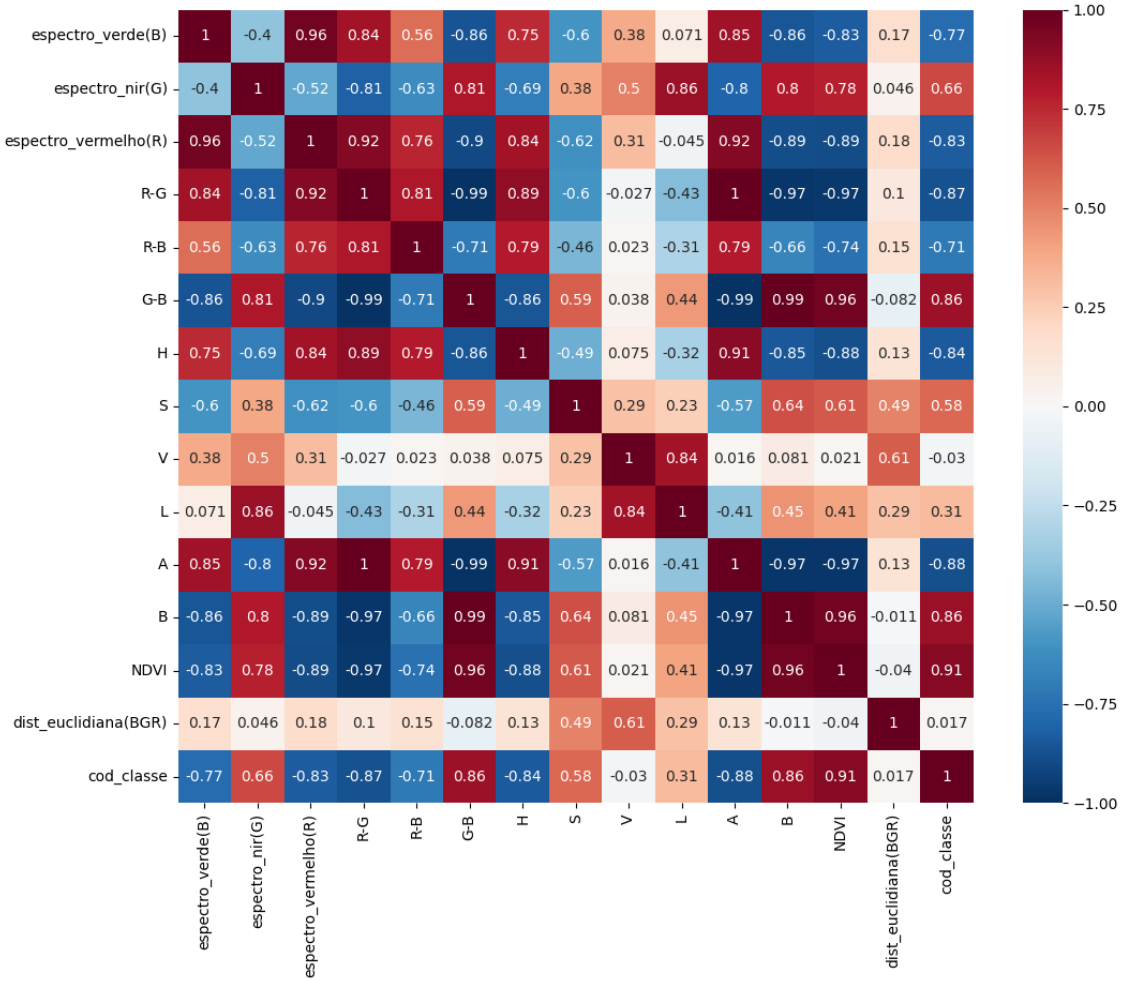

Tabela 2 – Correlação entre os atributos da base de dados

Atributo	Score
dist_euclidiana(BGR)	0.884383
espectro_vermelho(R)	0.840252
NDVI	0.831829
espectro_verde(B)	0.810697
R-G	0.703240
G-B	0.691974
A	0.690619
H	0.683583
B	0.639521
S	0.516800
R-B	0.502883
espectro_nir(G)	0.485906
V	0.310479
L	0.156021

Tabela 3 – Ranking de informação mútua de cada atributo

Assim, foram removidos atributos que possuem correlação forte com o NDVI, reduzindo a base de dados aos seguintes descritores: S, V, L, NDVI e dist_euclidiana(BGR).

3.5 Treinamento dos modelos

Inicialmente, utilizou-se do software Weka para a aplicação dos algoritmos de aprendizado de máquina, porém, devido às funcionalidades oferecidas pelas bibliotecas do Python, que facilitam a apresentação dos resultados e manipulação dos dados, optou-se pela sua utilização. Para a realização dos testes foram selecionados os seguintes algoritmos: *Random Forest*, SVM, *Gaussian Naive Bayes* e KNN.

Assim sendo, foi realizado o treinamento e teste com cada uma das bases de dados criadas na seção 3.3. Dessa forma, para cada algoritmo, foram criadas a versão "original", treinada com a base de dados original, e as versões filtradas "3", "5", "7", "9" e "11", identificadas pelo tamanho da máscara utilizada no algoritmo de filtragem.

Visando garantir uma melhor confiabilidade nos resultados, os testes foram executados utilizando validação cruzada com 5 *folds*. As métricas de comparação selecionadas foram: acurácia, precisão, sensibilidade, f-score, coeficiente kappa e MCC.

Após a execução de todos os testes, foram calculadas a média dos valores das métricas obtidas em cada uma das 5 execuções da validação cruzada. A partir daí, foi possível realizar uma comparação entre o desempenho dos algoritmos nas diferentes bases de dados.

3.6 Testes de segmentação de imagens

Após o treinamento dos modelos, também optou-se por verificar seus resultados na classificação de novos dados, de forma a identificar como o desempenho se alterava conforme a base de dados utilizada. Para isso, foi utilizado os modelos criados a partir da base de dados original e das bases "filtrada(3)" e "filtrada(11)" para a classificação dos pixels do recorte do dia 13/06/2021 (figura 14), que não foi aplicado no processo de treinamento. As bases de dados em questão foram selecionadas por serem bastante distintas dentre as criadas para este trabalho, permitindo analisar os resultados com grandes variações nos dados de entrada.

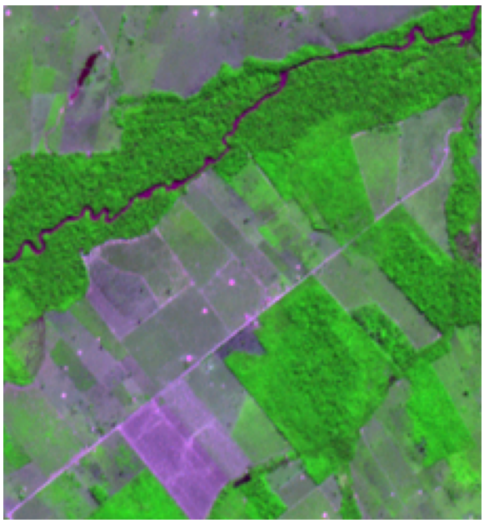

Figura 14 – Recorte do dia 13/06/2021, selecionado para o teste de segmentação

3.7 Segmentação por partes

Conferindo as classificações, percebeu-se que a maior dificuldade dos algoritmos foi em distinguir os pixels de solo nu e agropecuária, devido à proximidade entre as intensidades dos canais RGB desses pixels na imagem. Para contornar esse problema, foi proposta uma estratégia de classificação por partes, que consiste na composição de dois modelos diferentes na classificação dos pixels, aplicados em sequência sobre a imagem (como ilustrado na figura 15), sendo eles:

- Modelo 1: treinado para classificação de todas as classes, mas de onde são aproveitadas apenas as classificações de pixels referentes a corpos d'água e floresta
- Modelo 2: treinado apenas com pixels de solo nu e agropecuária para a classificação dessas regiões, sendo estes pixels suavizados com filtro gaussiano com máscara de tamanho 3x3

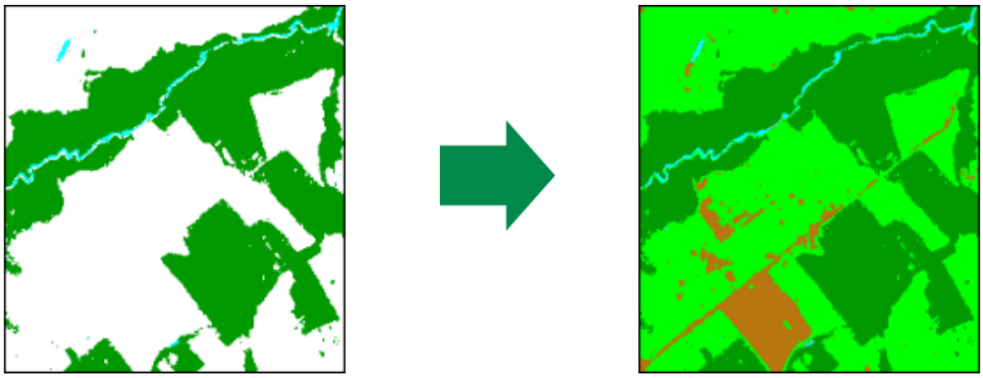

Figura 15 – Ilustração do processo de segmentação por partes

A partir disso, foram realizados outros testes da segmentação que consistiam na aplicação dessa estratégia para os quatro algoritmos selecionados, de forma que, para cada um deles, o primeiro modelo da composição fosse aquele treinado com a base de dados original, como apresentado na seção 3.5, e o segundo fosse resultado de um novo treinamento do algoritmo, utilizando apenas pixels de solo nu e agropecuária.

Dessa forma, buscou-se verificar se a realização de um treinamento isolado nos pixels de solo nu e agropecuária facilitaria os modelos na aquisição dos padrões de diferenciação entre as classes.

4 RESULTADOS E ANÁLISE

Neste capítulo são apresentados os resultados deste trabalho e uma análise dos mesmos, com base nos procedimentos executados no capítulo 3.

4.1 Resultados com *Random Forest*

O *Random Forest* obteve ótimo e consistente desempenho entre as bases de dados para as classes floresta, solo_nu e agropecuária. Já para os pixels de água, notou-se uma grande queda conforme maior a máscara do filtro aplicado nas imagens, característica que refletiu não apenas nas métricas focadas nesta classe, mas também nos coeficientes kappa, MCC e na acurácia geral. O melhor resultado obtido foi para a base de dados original, com uma acurácia de 93,7% e coeficiente kappa e MCC de 0,905.

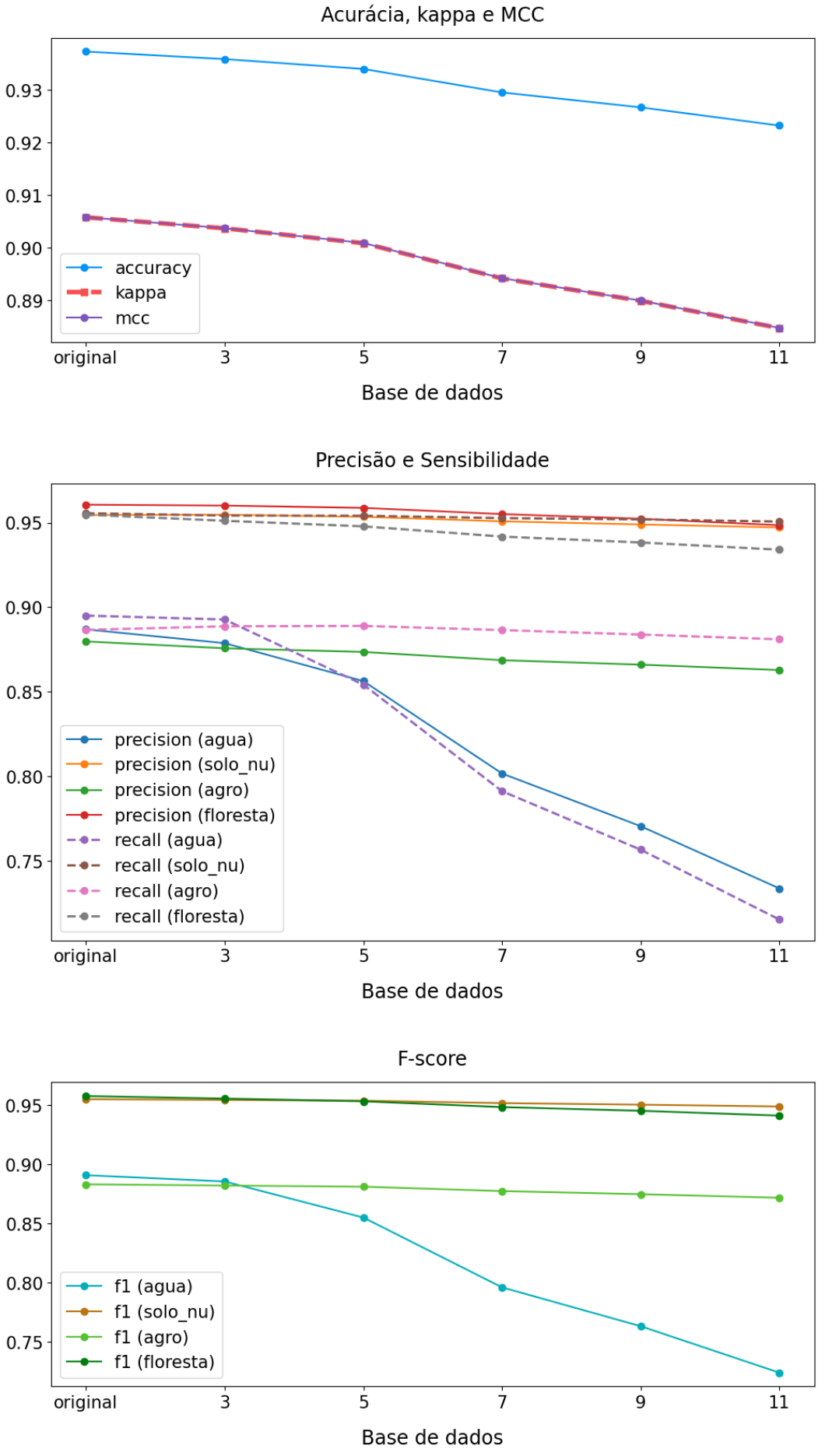

Figura 16 – Métricas obtidas pelo algoritmo *Random Forest* em cada base de dados

4.2 Resultados com KNN

O KNN se assemelha ao *Random Forest* no que tange a classificação dos pixels de floresta, solo_nu e agropecuária. Para os pixels de água notou-se uma pequena melhora de desempenho com a aplicação do filtro com máscara de tamanho 3x3. No entanto, essa melhora não se refletiu de maneira significativa nas outras métricas, onde ainda se percebeu uma grande queda causada pela aplicação dos filtros. O melhor resultado obtido foi para a base de dados original, com uma acurácia de 93,6% e coeficiente kappa e MCC de 0,904.

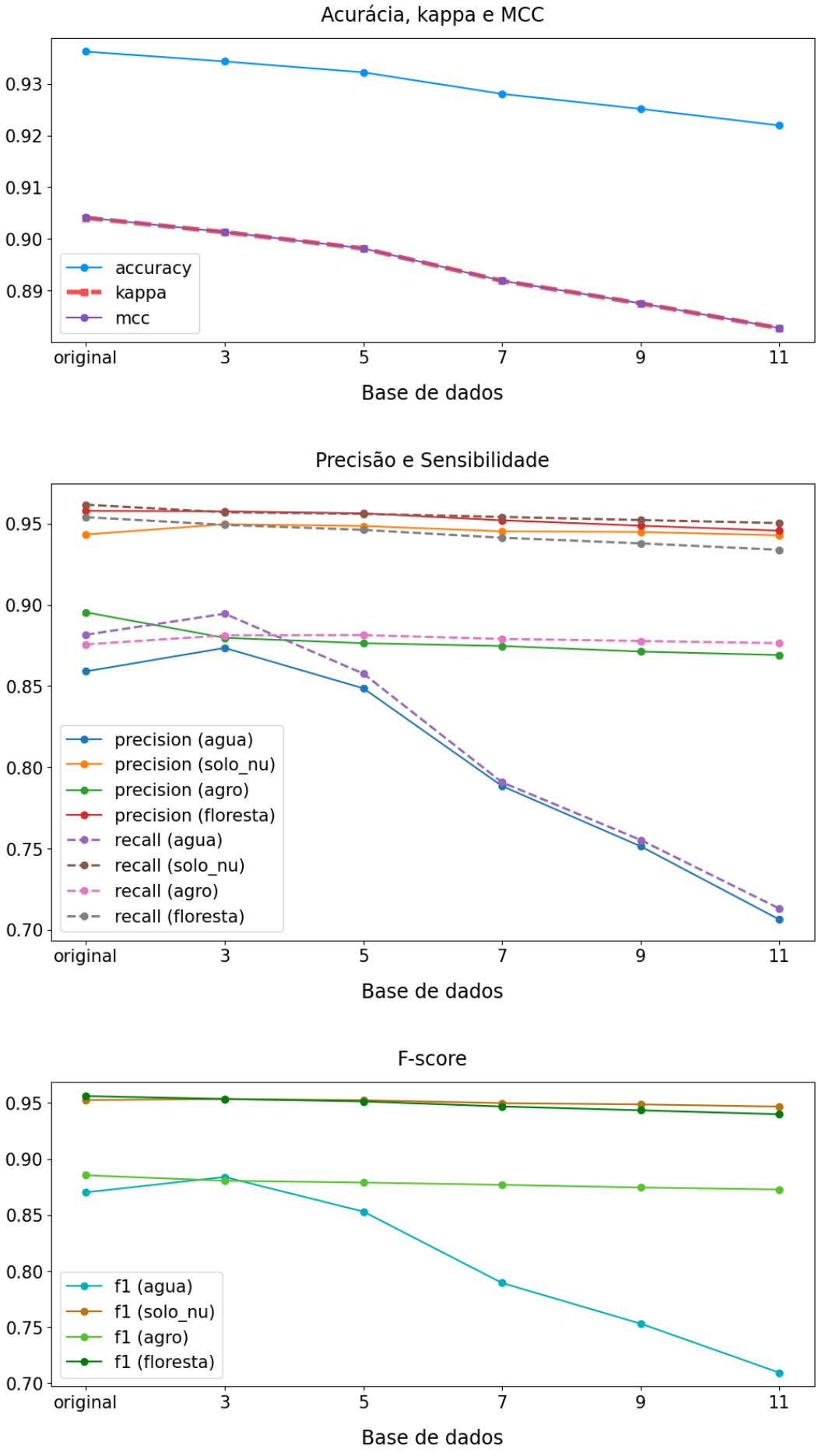

Figura 17 – Métricas obtidas pelo algoritmo KNN em cada base de dados

4.3 Resultados com *Gaussian Naive Bayes*

Para o *Gaussian Naive Bayes* , diferentemente dos algoritmos anteriores, os valores de precisão e f-score para os pixels de água e agropecuária foram muito menores do que o das outras classes, indicando que o algoritmo teve maior dificuldade de classificar essas amostras. Além disso, percebe-se que com a aplicação dos filtros esse algoritmo classificou alguns pixels erroneamente como pertencentes da classe água, fato que pode ser percebido pelo crescimento na sensibilidade e decaimento da precisão representados na figura 18. O melhor desempenho foi com a base de dados original, com uma acurácia de 84,9%, coeficiente kappa de 0,776 e MCC de 0,779, resultados bastante inferiores se comparados com o *Random Forest* e o KNN.

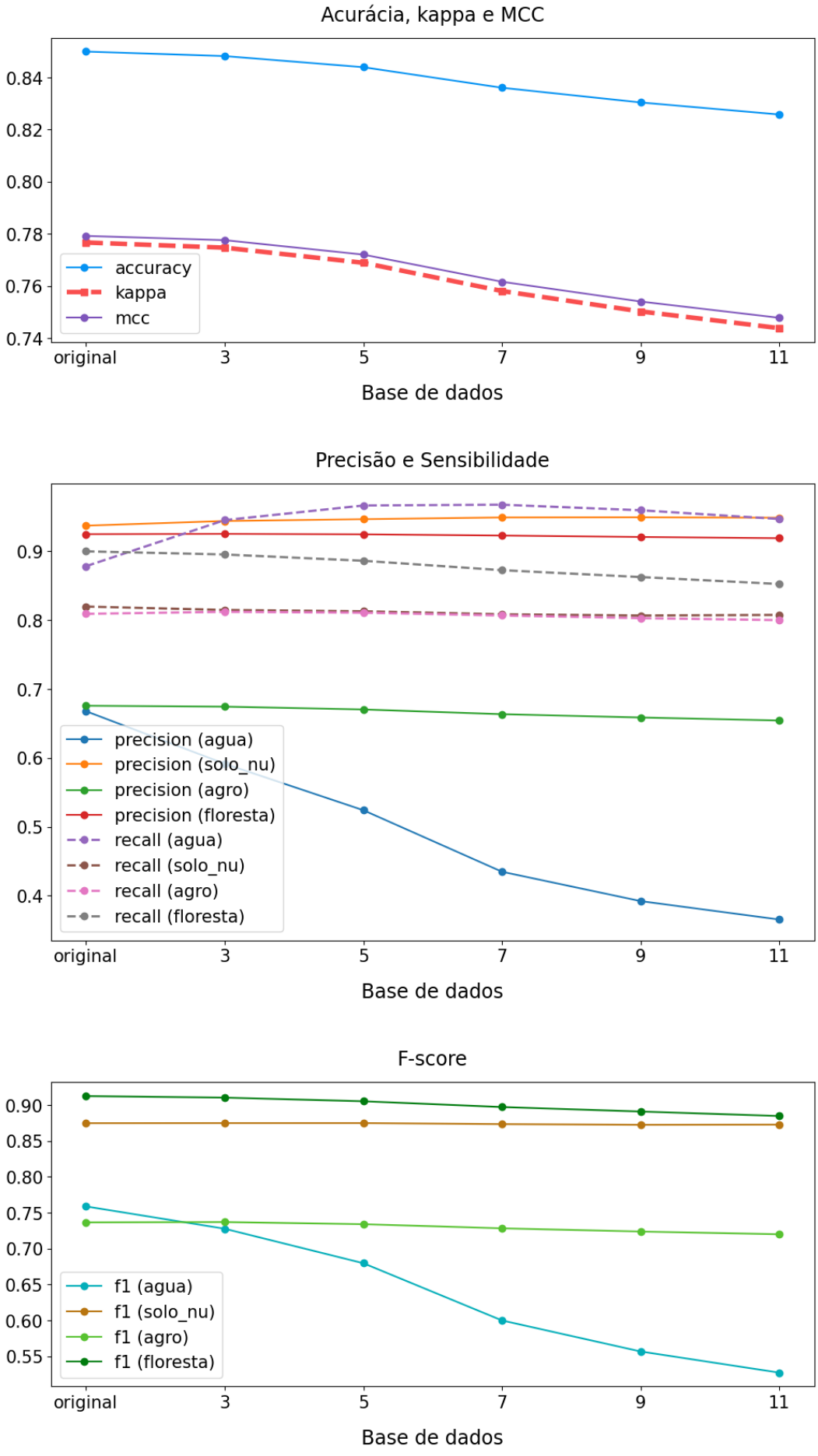

Figura 18 – Métricas obtidas pelo algoritmo *Gaussian Naive Bayes* em cada base de dados

4.4 Resultados com SVM

Com o SVM, a respeito da classe água, percebeu-se uma pequena melhoria utilizando a base de dados filtrada com máscara de tamanho 3x3, resultado que refletiu em um aumento na acurácia, coeficiente kappa e MCC. Porém, este classificador não obteve bons resultados gerais se comparados aos outros algoritmos, de forma que o seu melhor resultado foi com a base de dados filtrada com máscara 3x3, atingindo uma acurácia de 89,1% e coeficiente kappa e MCC de 0,836.

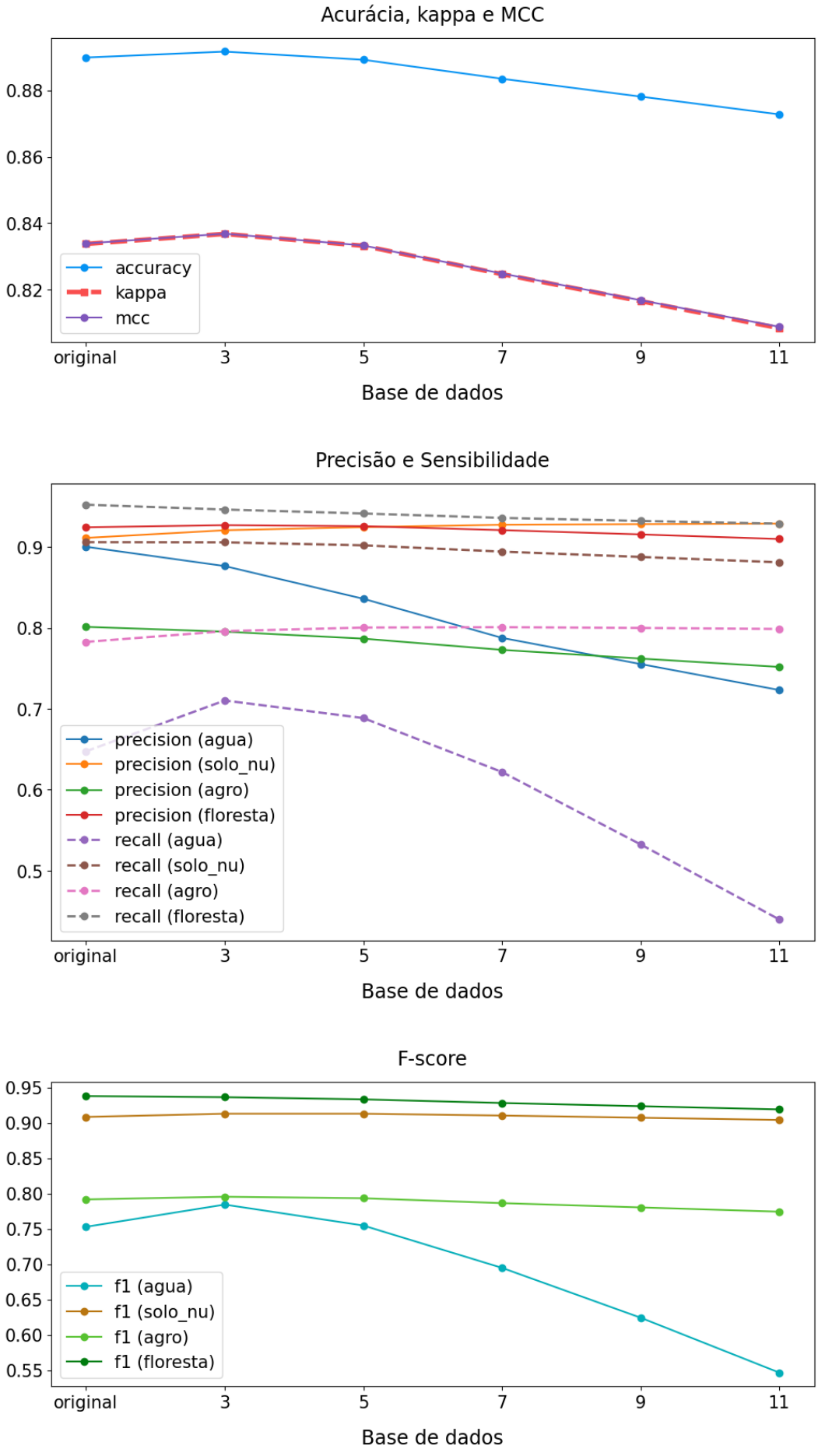

Figura 19 – Métricas obtidas pelo algoritmo SVM em cada base de dados

4.5 Comparação dos resultados do treinamento

Analisando os resultados, percebeu-se que, em geral, conforme maior o tamanho da máscara do filtro aplicado nas imagens, menores foram os valores das métricas, principalmente da acurácia, coeficiente kappa e MCC. Para os pixels de água, especificamente, também foi possível visualizar uma grande queda na precisão e sensibilidade (e, por consequência, no f-score) da classificação.

Tomando os melhores resultados de cada algoritmo, percebeu-se uma dominância do KNN e do *Random Forest*, que obtiveram, respectivamente, acurácia de 93,6% e 93,7% para a base de dados original. Além disso, os valores dos coeficientes MCC e kappa também foram altos, indicando que existe concordância entre as predições desses modelos e o valores esperados.

Contudo, para uma avaliação mais confiável dos modelos também se fez necessário medir a eficiência dos modelos na classificação de novos dados, verificando o valor de acurácia obtida na classificação do recorte do dia 13/06/2021.

4.6 Resultados dos testes de segmentação

Analisando os resultados visuais obtidos, como apresentados nas figuras 20, 21 e 22, constatou-se que conforme maior o tamanho da máscara utilizada, menos precisas foram as segmentações de todos os tipos de regiões. Além disso, comprovou-se o que foi dito na seção 4.5 acerca da classificação de instâncias da classe “água”, onde se percebeu uma queda no desempenho da classificação dos pixels desse tipo. Consultando as imagens foi possível verificar como a aplicação dos filtros prejudicou a segmentação dos corpos d’água em todos os algoritmos, algo que ocorreu devido ao formato delgado desse tipo de região nas imagens utilizadas, o qual torna sua estrutura bastante sensível à filtragem aplicada no pré-processamento.

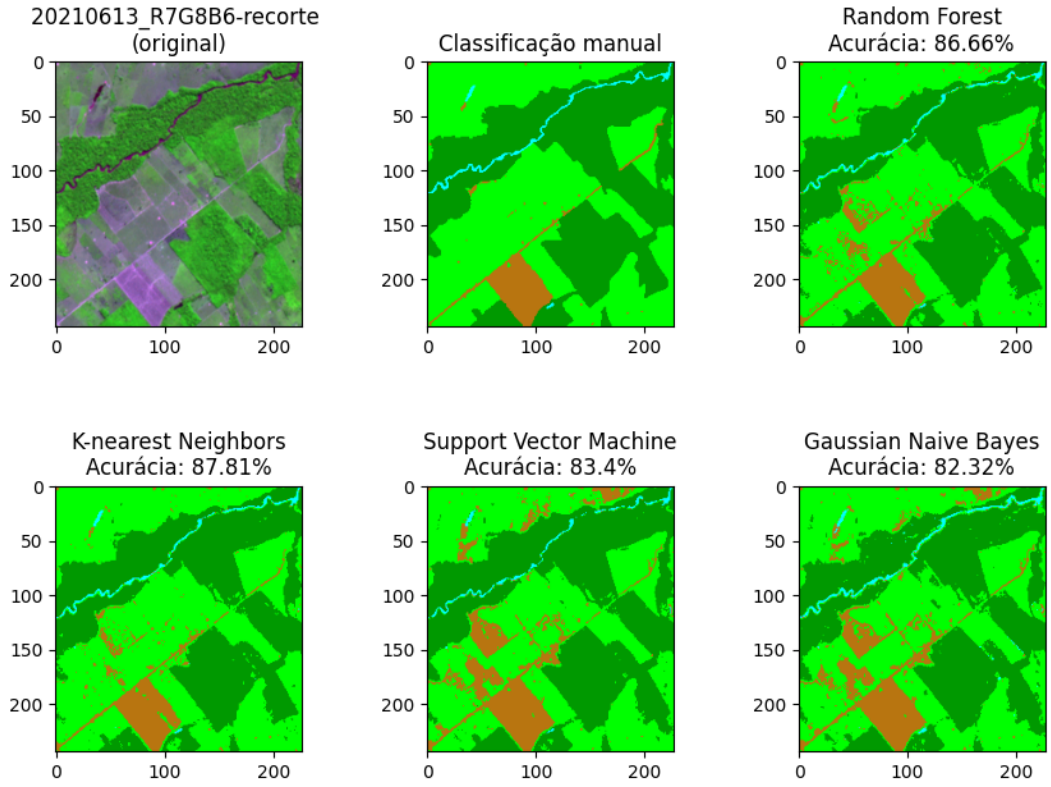

Figura 20 – Imagens segmentadas geradas pelos algoritmos treinados com a base de dados original

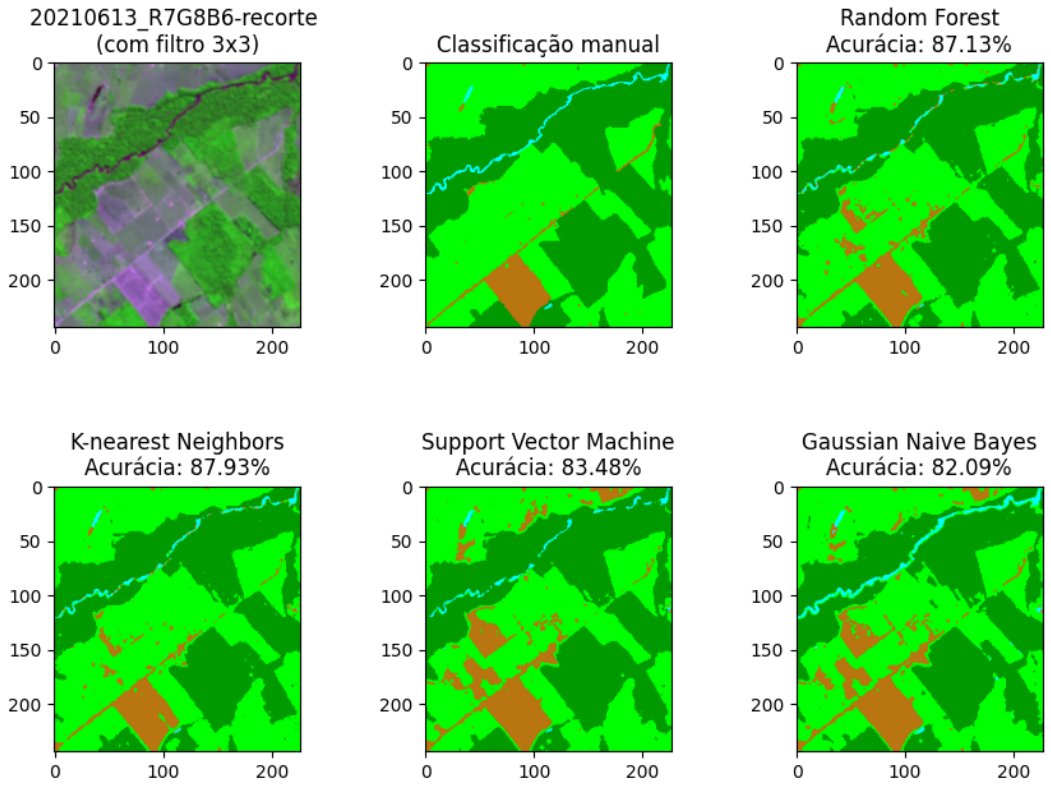

Figura 21 – Imagens segmentadas geradas pelos algoritmos treinados com a base de dados "filtrada(3)"

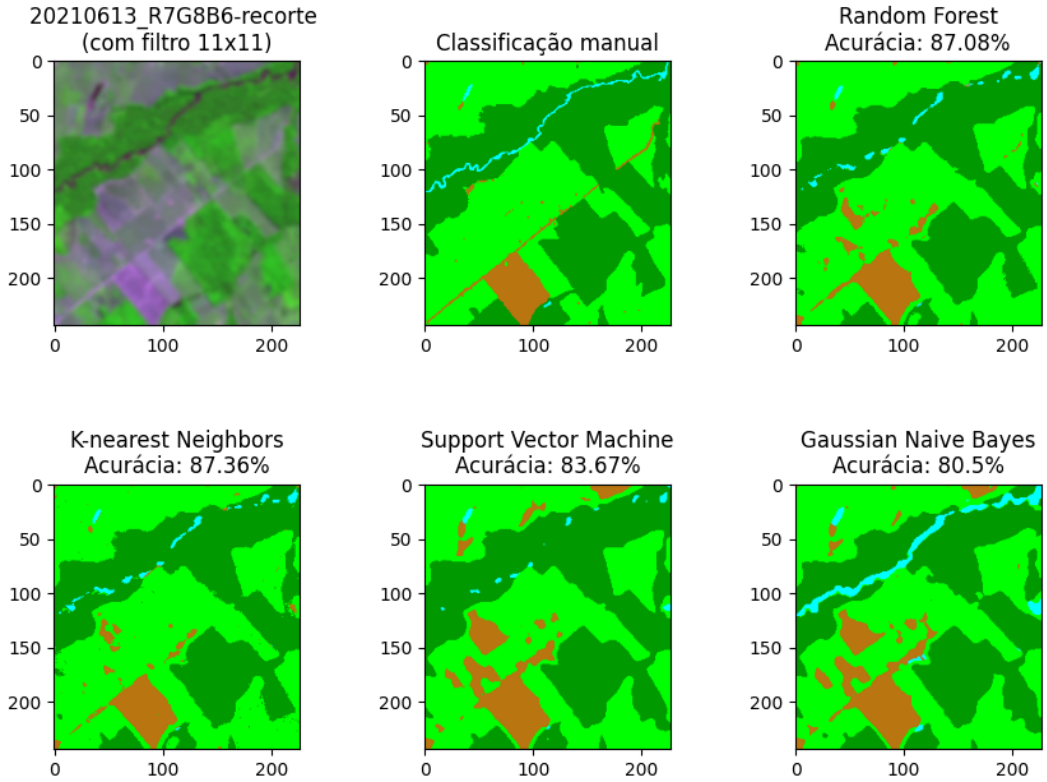

Figura 22 – Imagens segmentadas geradas pelos algoritmos treinados com a base de dados "filtrada(11)"

Nos modelos com o algoritmo *Gaussian Naive Bayes*, as regiões com água tiveram uma área segmentada maior do que a esperada, informando que esse algoritmo classificou alguns pixels erroneamente como pertencentes a ela, fato esse que corrobora com a queda na precisão e aumento da sensibilidade observadas no gráfico da figura 18. Já para o restante dos algoritmos, esse tipo de região teve a sua área reduzida conforme maior o tamanho da máscara utilizada no filtro de pré-processamento, indicando que muitos pixels pertencentes a corpos d'água não foram classificados corretamente.

Dessa forma, foi possível identificar que a aplicação do filtro com tamanho de máscara maiores que 3 não se provou muito interessante para as imagens aqui selecionadas, devido à resolução espacial grande de 16,5 metros. Assim, é possível que a utilização dessa técnica seja interessante para o mapeamento de grandes regiões, como as áreas florestais, onde os pixels representam uma área de poucos metros e um simples elemento destoante, como uma árvore de uma cor específica, pode prejudicar a segmentação de uma região.

Por outro lado, a aplicação dos filtros com máscara de tamanho 3x3 se mostrou promissora para a classificação dos pixels de solo nu e agropecuária, onde se percebeu a eliminação de alguns ruídos na segmentação dessas regiões, destacada pela melhoria de acurácia nas segmentações realizadas pelos algoritmos *Random Forest* e KNN para a base de dados filtrada com máscara de tamanho 3x3.

4.7 Resultados da segmentação por partes

Por fim, como citado na seção 3.7 foi percebido que a maior dificuldade dos modelos foi a de diferenciar pixels de solo nu e agropecuária e, buscando contornar a situação, foi proposta uma estratégia diferente na segmentação das imagens, utilizando dois modelos distintos. A métricas obtidas pelo Modelo 2 de cada algoritmo (treinado apenas para a classificação de pixels de solo nu e agropecuária) podem ser visualizadas na figura 23. Aqui, percebeu-se que os melhores algoritmos foram o *Random Forest* e o KNN, sendo que o primeiro obteve uma leve vantagem, com uma acurácia de 95,1% e coeficientes kappa e MCC de 0,9.

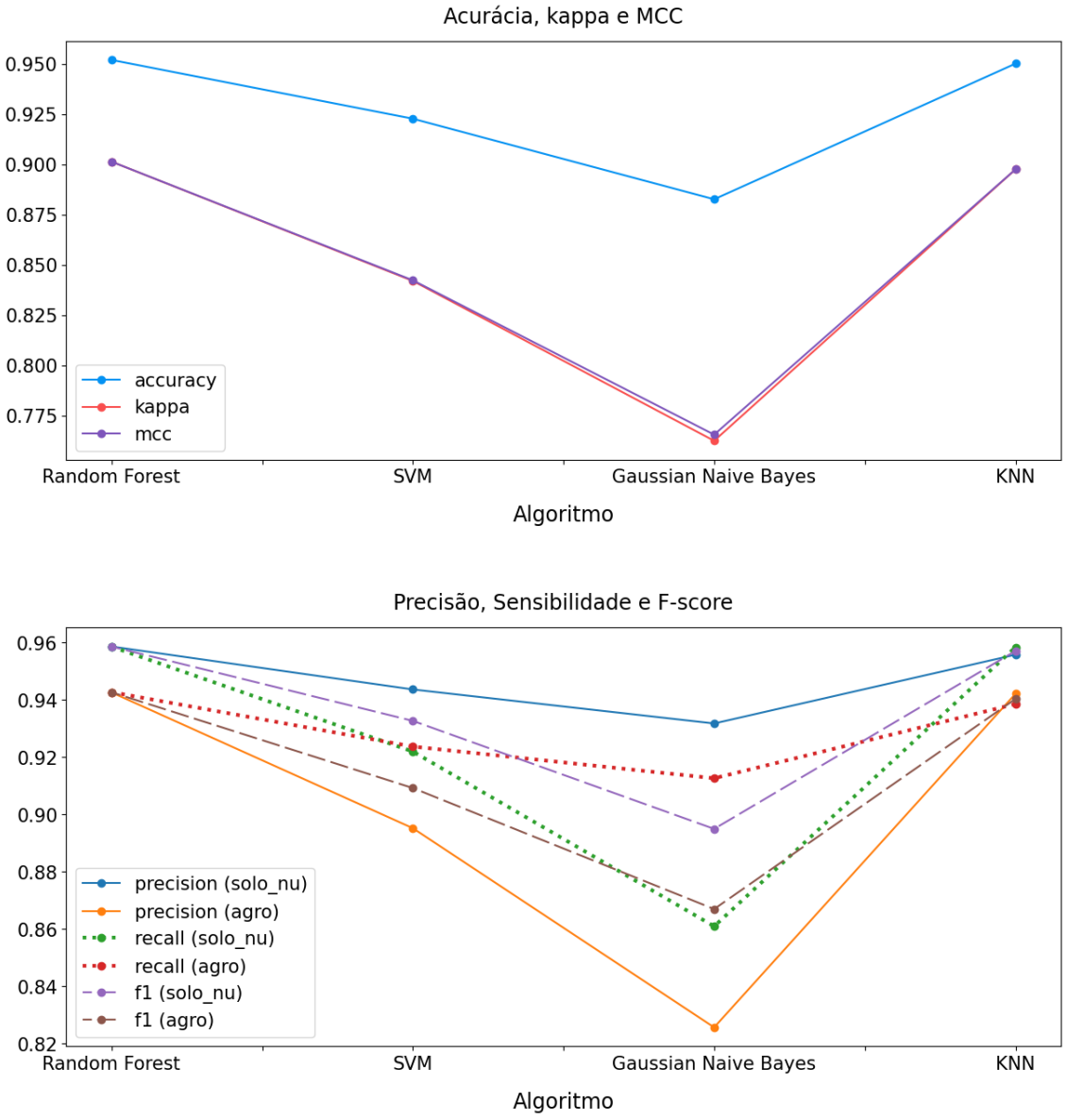

Figura 23 – Métricas obtidas pelos algoritmos treinados para classificação de solo nu e agropecuária utilizando a base de dados "filtrada(3)"

Analisando as segmentações geradas, apresentadas na figura 24, percebeu-se que essa estratégia foi a que gerou melhores resultados para todos os algoritmos, com exceção do *Gaussian Naive Bayes* que sofreu uma queda na acurácia do teste. Combinando a predição de dois modelos foi possível aproveitar do aumento da acurácia para as classes de solo_nu e agropecuária, resultantes da aplicação do filtro com máscara de tamanho 3x3, mantendo a boa classificação dos corpos d'água para a base de dados original.

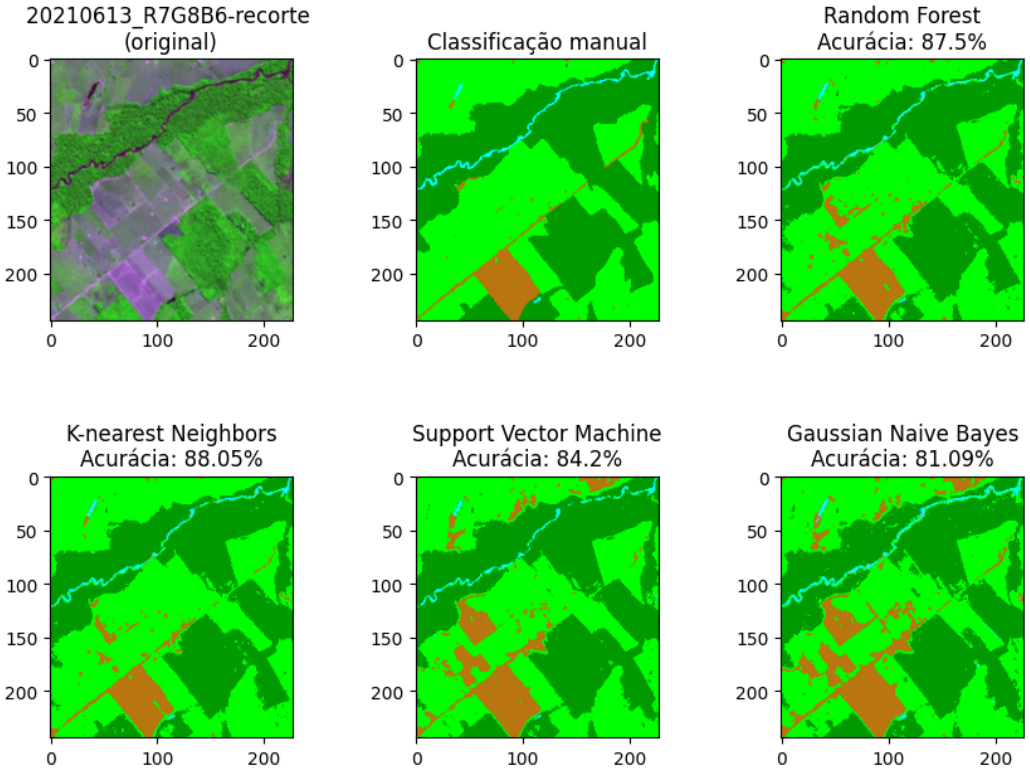

Figura 24 – Imagens segmentadas geradas pelos algoritmos utilizando a estratégia de segmentação por partes

Com essa estratégia mantiveram-se os algoritmos *Random Forest* e KNN como os melhores classificadores, ambos apresentando um valor de acurácia semelhante no treinamento, como apresentado na tabela 4.

Algoritmo	Acurácia (Modelo 1)	Acurácia (Modelo 2)
Random Forest	93,6%	95,1%
KNN	93,7%	95%

Tabela 4 – Comparação entre acurácia dos modelos com *Random Forest* e KNN obtidas no treinamento

Porém, como pode ser observado na tabela 5, a utilização do *Random Forest* resultou em um tempo de execução bem mais baixo para o teste de segmentação, de forma que a vantagem na acurácia do KNN não foi considerada relevante.

Algoritmo	Tempo de execução	Acurácia no teste de segmentação
Random Forest	3,9 segundos	87,5%
KNN	7,1 segundos	88,05%

Tabela 5 – Comparação entre os resultados dos *Random Forest* e KNN nos testes de segmentação

Em suma, concluiu-se que a estratégia de segmentação por partes utilizando o *Random Forest* obteve bons resultados para a classificação de todas as classes e, principalmente, na segmentação de florestas, significando que o modelo criado pode atingir bons resultados se aplicado em projetos relacionados ao controle de desmatamento, onde se é necessário apenas detecção do que é e o que não é vegetação.

5 CONCLUSÃO

O objetivo deste trabalho foi de construir um software automatizado para a classificação de imagens de satélite da região da floresta amazônica. Para isso, foram testados quatro diferentes algoritmos com diferentes estratégias de classificação/segmentação das imagens.

Após a análise realizada no capítulo 4, chegou-se a conclusão de que os melhores resultados foram obtidos utilizando a estratégia de classificação por partes com *Random Forest*. Isso se deve ao fato desse algoritmo ter apresentado uma boa acurácia, tanto no treinamento quanto nos testes, e um tempo de execução na classificação bem menor do que o do KNN. Além disso, também foi possível perceber indícios que a aplicação do filtro gaussiano com máscara de tamanho 3x3 pode melhorar os resultados obtidos.

Para trabalhos futuros, espera-se a utilização da metodologia aqui desenvolvida na resolução de problemas práticos, como, por exemplo, o monitoramento do desmatamento ou das transformações de terreno de uma região ao longo de um período. Além disso, também deve ser possível aprimorar os resultados aqui encontrados por meio da utilização de algoritmos de classificação mais robustos, como Redes Neurais Convolucionais. Outra situação possível de ser explorada é a de testar a viabilidade da aplicação do filtro gaussiano em imagens de satélite onde os pixels representam uma área menor, de forma que possam ser retirados ruídos que representam apenas pequenos elementos na paisagem.

REFERÊNCIAS

[1] QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. *Rita*, v. 13, n. 2, p. 11–42, 2006.

[2] LOPES, J. M. B. Cor e luz. *Texto elaborado para a disciplina de Computação Gráfica*, p. 34, 2013.

[3] IBRAHEEM, N. A. et al. Understanding color models: a review. *ARPN Journal of science and technology*, Citeseer, v. 2, n. 3, p. 265–275, 2012.

[4] RAKSHIT, S.; DEBNATH, S.; MONDAL, D. Identifying land patterns from satellite imagery in amazon rainforest using deep learning. *arXiv preprint arXiv:1809.00340*, 2018.

[5] A Amazônia em números. 2013. Imazon Website. Disponível em: <https://imazon.org.br/imprensa/a-amazonia-em-numeros/>. Acesso em: 10.1.2023.

[6] O que é? Amazônia Legal. 2008. Ipea Website. Disponível em: <https://www.ipea.gov.br/desafios/index.php?option=com_content&id=2154:catid=28>. Acesso em: 11.1.2023.

[7] ANDRADE, R. et al. Evaluation of semantic segmentation methods for deforestation detection in the amazon. *ISPRS Archives; 43, B3*, Katlenburg-Lindau: Copernicus Publications, v. 43, n. B3, p. 1497–1505, 2020.

[8] MONTEIRO, F. P. et al. Classis: uma metodologia para classificação supervisionada de imagens de satélite em áreas de assentamento localizados na amazônia. Universidade Federal do Pará, 2015.

[9] OLIVEIRA, J. P. D. et al. Segmentação semântica de tipos de uso de solo na amazônia utilizando aprendizado profundo. In: *GeoInfo*. [S.l.: s.n.], 2020. p. 198–203.

[10] MENESES, P. R.; ALMEIDA, T. d. Introdução ao processamento de imagens de sensoriamento remoto. *Universidade de Brasília, Brasília*, 2012.

[11] INPE. *CBERS 04A*. 2019. Website do INPE. Disponível em: <http://www.cbers.inpe.br/sobre/cbers04a.php>. Acesso em: 6.1.2023.

[12] JESUS, E. O.; JR, R. C. A utilização de filtros gaussianos na análise de imagens digitais. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 3, n. 1, 2015.

[13] RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. [S.l.]: Pearson, 2021. (Pearson series in artificial intelligence). ISBN 9781292401133.

[14] BARBOSA, M. *ABORDAGEM BASEADA NA EXTRAÇÃO DE ATRIBUTOS DO PIXEL PARA QUANTIFICAR A SEVERIDADE DA FERRUGEM ASIÁTICA EM IMAGENS DE FOLHA DE SOJA*. 2021. Disponível em: <www.tcpdf.org>.

[15] PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[16] GOLDBERGER, J. et al. Neighbourhood components analysis. In: SAUL, L.; WEISS, Y.; BOTTOU, L. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2004. v. 17. Disponível em: <https://proceedings.neurips.cc/paper/2004/file/42fe880812925e520249e808937738d2-Paper.pdf>.

[17] ISLAM, M. J. et al. Investigating the performance of naive- bayes classifiers and k- nearest neighbor classifiers. In: *2007 International Conference on Convergence Information Technology (ICCIT 2007)*. [S.l.: s.n.], 2007. p. 1541–1546.

[18] CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, Elsevier, v. 40, n. 1, p. 16–28, 2014.

[19] BROWNLEE, J. *How to Choose a Feature Selection Method For Machine Learning*. 2019. Machine Learning Mastery Website. Disponível em: <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/>. Acesso em: 6.1.2023.

[20] CALKINS, K. G. *Correlation Coefficients*. 2005. Andrews University Website. Disponível em: <https://www.andrews.edu/~calkins/math/edrm611/edrm05.htm>. Acesso em: 7.2.2023.

[21] BROWNLEE, J. *Machine Learning Mastery with Weka: Analyze Data, Develop Models, and Work Through Projects*. [S.l.]: Machine Learning Mastery, 2016.