

UNIVERSIDADE FEDERAL DE SÃO CARLOS– UFSCAR  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA– CCET  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA– DEE  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA– PPGEE

**Tayna Bertacine de Almeida**

**Detecção de Ninhos de formigas do  
gênero *Atta* utilizando visão  
computacional e quadrrrotores**

São Carlos  
2023



**Tayna Bertacine de Almeida**

**Detecção de Ninhos de formigas do  
gênero *Atta* utilizando visão  
computacional e quadrirrotores**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Ciências Exatas e de Tecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Sistemas Elétricos e Eletrônicos

Orientador: Prof. Dr. André Carmona Hernandes

São Carlos

2023





**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

Centro de Ciências Exatas e de Tecnologia  
Programa de Pós-Graduação em Engenharia Elétrica

---

**Folha de Aprovação**

---

Defesa de Dissertação de Mestrado da candidata Tayná Bertacine de Almeida, realizada em 04/07/2023.

**Comissão Julgadora:**

Prof. Dr. Andre Carmona Hernandes (UFSCar)

Prof. Dr. Marcelo Becker (USP)

Prof. Dr. Roberto Santos Inoue (UFSCar)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia Elétrica.



*Este trabalho é dedicado à minha família e a todos  
que acreditaram que eu era capaz e não me deixaram desistir.*



---

# Agradecimentos

Agradeço primeiramente à Deus, que além de ter permitido que tudo isso acontecesse ainda me deu força e sabedoria para concluir mais essa etapa.

A toda a minha família, pelos incentivos aos meus estudos e por torcerem sempre pelo meu sucesso.

Ao meu orientador, Prof. Dr. André Carmona Hernandes, pela orientação, dedicação, paciência e confiança.

Ao Prof. Dr. Marcelo Becker e ao Prof. Dr. Roberto Santos Inoue, pela contribuição com críticas e sugestões valiosas.

Aos(Às) professores(as) e funcionários(as) do Departamento de Engenharia Elétrica (DEE) e do Departamento de Computação (DC) da Universidade Federal de São Carlos (UFSCar) pelas disciplinas ministradas e por todo o suporte necessário durante esse período.

Aos meus colegas da pós-graduação e ao Grupo *Mobile Robotics Laboratory*, pela troca de ideias, ajuda e apoio.

Um agradecimento especial ao Fábio Mariotto de Azevedo por toda a ajuda com conceitos de *Data Science* e apoio nos momentos em que eu pensei em desistir (que não foram poucos).

Aos meus amigos e amigas, pelo apoio nos momentos difíceis e por terem compartilhado momentos de lazer necessários.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte financeiro e concessão da bolsa CAPES-DS durante parte do meu mestrado.



*“Porque o SENHOR dá a sabedoria,  
e da sua boca vem o conhecimento e o entendimento.”*

*Provérbios 2:6*



---

# Resumo

Nos últimos 40 anos, a produção e produtividade agropecuária brasileira sofreram aumentos significativos. O agronegócio tem sido reconhecido como um forte aliado no crescimento do Produto Interno Bruto brasileiro, o que resultou em investimentos e avanços tecnológicos no setor agrícola. Com esses avanços tecnológicos, surgiu-se a agricultura de precisão. A agricultura de precisão auxilia os agricultores no gerenciamento de seus insumos e produtos a fim de otimizar sua produção e produtividade. Como integrante dos avanços em agricultura de precisão, as Aeronaves Não Tripuladas vem ganhando cada vez mais espaço, dado a sua flexibilidade de uso e diversidade de aplicações. Paralelamente, a utilização de visão computacional e aprendizado de máquina trazem uma tecnologia de análise que otimizam ainda mais os processos, como por exemplo, de mapeamento de pasto e detecção de pragas. E quando se fala em pragas de plantações, as formigas cortadeiras são lembradas. As perdas diretas e indiretas que as formigas cortadeiras podem causar chegam a atingir bilhões de dólares em todo o mundo. A formiga cortadeira, conhecida como Saúva Parda, é um dos insetos que mais causa danos nas pastagens do sudeste do Brasil, chegando a competir com o gado por capim. O controle de pragas é de suma importância para o crescimento da produção agrícola, sendo assim, o presente trabalho apresenta a proposta de desenvolvimento de um algoritmo capaz de detectar formigueiros.

Com base nos recentes avanços tecnológicos e redução de custos, a detecção proposta foi realizada utilizando um veículo aéreo não tripulado com uma câmera RGB convencional integrada e um algoritmo baseado em visão computacional e aprendizado de máquina. O algoritmo de classificação proposto foi *Random Forest* e apresentou acurácia de aproximadamente 70%. A aplicação de visão computacional e quadrotoros na detecção de formigueiros representou uma boa oportunidade para impulsionar a agricultura de precisão e o controle de pragas no campo.

**Palavras-chave:** Sensoriamento Remoto. Agricultura de Precisão. Robótica Aérea. Visão Computacional. Formigueiro.



---

# Abstract

In the last 40 years, Brazilian agricultural production and productivity have had significant increases. Agribusiness has been strongly recognised in the growth of the Brazilian Gross Domestic Product, which has resulted in investments and technological advances in the agricultural sector. With these technological advances came precision agriculture. As part of the advances in precision agriculture, Unmanned Aircraft are gaining more and more space, given their flexibility of use and diversity of applications. At the same time, the use of computer vision and machine learning brings analysis technologies that further optimise processes, such as pasture mapping and pest detection. And when it comes to crop pests, leaf-cutter ants come to mind. The direct and indirect damage that leaf-cutting ants can cause amounts to billions of dollars worldwide. The leaf-cutting ant, known as *Sauva*, is one of the insects that causes most damage to pastures in southeastern Brazil, even competing with cattle for grass. Pest control is of paramount importance for the growth of agricultural production, therefore, this article presents the proposal for the development of an algorithm capable of detecting anthills. Based on recent technological advances and cost reduction, the proposed detection was performed using an unmanned aerial vehicle with an integrated conventional RGB camera and an algorithm, based on computer vision and machine learning. The proposed classification algorithm was Random Forest and it presented an accuracy of approximately 70%. The application of computer vision and quadrotors in the detection of ant nests represented a good opportunity to boost precision agriculture and pest control in the field.

**Keywords:** Remote Sensing. Precision Agriculture. Aerial Robotics. Computer Vision. Ant Nest.



---

## Lista de ilustrações

Figura 1 – Indicadores de produção e área plantada de 1975 a 2017. . . . .	23
Figura 2 – Crescimento da produção agrícola . . . . .	24
Figura 3 – Crescimento da produção pecuária . . . . .	24
Figura 4 – Morfologia das formigas do gênero <i>Atta</i> . . . . .	38
Figura 5 – Morfologia das formigas do gênero <i>Acromyemex</i> . . . . .	39
Figura 6 – Morfologia externa dos ninhos . . . . .	40
Figura 7 – Drone de asa fixa . . . . .	42
Figura 8 – Drone de asa rotativa rotor único . . . . .	44
Figura 9 – Drone de asa rotativa multi rotor . . . . .	44
Figura 10 – Visada para baixo . . . . .	45
Figura 11 – Dimensões das imagens obtidas pela drone . . . . .	46
Figura 12 – Esquema de um sistema de Visão Computacional . . . . .	48
Figura 13 – Modelo de Pinhole . . . . .	49
Figura 14 – Tipos de distorção de imagem . . . . .	50
Figura 15 – Triângulo dos parâmetros de fotografia . . . . .	50
Figura 16 – Efeitos das alterações dos parâmetros na fotografia . . . . .	52
Figura 17 – Funcionamento <i>Rolling Shutter</i> . . . . .	54
Figura 18 – <i>Rolling Shutter X Global Shutter</i> . . . . .	54
Figura 19 – Processo de adição das cores RGB. . . . .	57
Figura 20 – Diagramas do cubo representativo de RGB. . . . .	58
Figura 21 – Diagramas do cone representativo de HSV. . . . .	59
Figura 22 – Faixa de valores de cada variável CIELAB . . . . .	60
Figura 23 – Diagrama da esfera representativa de CIELAB . . . . .	61
Figura 24 – Descritor de textura HOG . . . . .	66
Figura 25 – Funcionamento de um LBP . . . . .	67
Figura 26 – Descritor de textura LBP . . . . .	67
Figura 27 – Média de <i>Skewness</i> . . . . .	72
Figura 28 – Média de <i>Kurtosis</i> . . . . .	73

Figura 29 – Funcionamento de uma árvore de decisão . . . . .	75
Figura 30 – Matriz de confusão teórica . . . . .	79
Figura 31 – Matriz de confusão exemplo . . . . .	80
Figura 32 – Casos em que os formigueiros se assemelham ao solo. Marcados por um especialista, os formigueiros estão identificados em azul e, em amarelo estão os que não se sabem se é somente solo ou formigueiro . . . . .	86
Figura 33 – Casos ainda em que os formigueiros se assemelham a montes de cupim. Marcados por um especialista, os formigueiros estão identificados em azul e o cúnzeiro em amarelo. . . . .	87
Figura 34 – Formigueiros Oclusos. Em (a) Oclusão por sombra. Em (b) Oclusão parcial pela vegetação. . . . .	87
Figura 35 – Efeitos da cromaticidade. . . . .	88
Figura 36 – Esquema da análise fatorial . . . . .	94
Figura 37 – Gráfico de interações $n\_seg$ x $comp$ . . . . .	95
Figura 38 – Gráfico de interações $comp$ x $n\_seg$ . . . . .	95
Figura 39 – IDE Demeter . . . . .	96
Figura 40 – Balanceamento da base de dados classificada manualmente . . . . .	99
Figura 41 – Aplicação dos descritores . . . . .	101
Figura 42 – Exemplo de árvore gerada pelo modelo . . . . .	113
Figura 43 – Histograma espaço de cor RGB . . . . .	113
Figura 44 – Histograma espaço de cor HSV . . . . .	114
Figura 45 – Histograma espaço de cor LAB . . . . .	114
Figura 46 – Drone modelo FIMI X8 SE . . . . .	133

---

## Lista de tabelas

Tabela 1 – Comparação dos resultados da Análise Bibliométrica realizada em anos diferentes . . . . .	30
Tabela 2 – Estado da arte . . . . .	35
Tabela 3 – Espécies de formigas e suas diferenças morfológicas . . . . .	38
Tabela 4 – Diferenças na morfologia externa dos ninhos . . . . .	40
Tabela 5 – Comparação dos sensores CCD e CMOS . . . . .	53
Tabela 6 – Aquisição de imagens . . . . .	92
Tabela 7 – Média aritmética condicional de "n_seg" . . . . .	94
Tabela 8 – Média aritmética condicional de "comp" . . . . .	94
Tabela 9 – Comando de classificação . . . . .	97
Tabela 10 – Resumo das versões dos códigos de <i>Random Forest</i> . . . . .	108
Tabela 11 – Matriz de confusão <i>v0</i> . . . . .	110
Tabela 12 – Matriz de confusão <i>v1</i> . . . . .	110
Tabela 13 – Matriz de confusão <i>v2</i> . . . . .	111
Tabela 14 – Matriz de confusão <i>v3.0</i> . . . . .	111
Tabela 15 – Matriz de confusão <i>v3.1</i> . . . . .	112
Tabela 16 – Matriz de confusão <i>v4.0</i> . . . . .	115
Tabela 17 – Matriz de confusão <i>v4.1</i> . . . . .	116
Tabela 18 – Resumo dos resultados . . . . .	117
Tabela 19 – Matriz de confusão versão extra . . . . .	118
Tabela 20 – Análise desvio padrão . . . . .	119



---

## Lista de siglas

**AM** Aprendizado de Máquina

**AP** Agricultura de Precisão

**APP** Área de Preservação Permanente

**CCD** *Charge Coupled Device*

**CEPEA** Centro de Estudos Avançados em Economia Aplicada

**CIVE** *Colour Index of Vegetation Extraction*

**CMOS** *Complementary Metal Oxide Semiconductor*

**CNA** Confederação da Agricultura e Pecuária do Brasil

**CNN** Redes Neurais Convolucionais

**DECEA** Departamento de Controle do Espaço Aéreo

**DFT** Transformada Discreta de Fourier

**DOF** Profundidade de Campo

**ExG** *Excess Green*

**ExR** *Excess Red*

**FFT** Transformada Rápida de Fourier

**FOV** **Field of view**

**HOG** Histograma de Gradientes Orientados

**IDE** Ambiente de Desenvolvimento Integrado

**IR** Infravermelho

**PIB** Produto Interno Bruto

**LPB** *Local Binary Pattern*

**ME<sub>x</sub>G** *Medified Excess Green Index*

**NDI** *Normalised Difference Index*

**RL** Reserva Legal

**RNA** Redes Neurais Artificiais

**SLIC** *Simple Linear Iterative Clustering*

**SP** *Super Pixel*

**UAS** *Unmanned Aerial Systems*

**UAV** *Unmanned Aerial Vehicle*

**UFSCar** Universidade Federal de São Carlos

**VANT** Veículo Aéreo Não Tripulado

**VEG** *Vegetation Index*

**VLOS** Operação em Linha de Visada Visual

---

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>23</b>
<b>2</b>	<b>ESTADO DA ARTE E REFERÊNCIAL TEÓRICO . . . . .</b>	<b>29</b>
<b>2.1</b>	<b>Análise bibliométrica . . . . .</b>	<b>29</b>
<b>2.2</b>	<b>Estado da arte . . . . .</b>	<b>30</b>
2.2.1	Detecção de formigueiro (Satélite) . . . . .	30
2.2.2	Mapeamento de desfolha (Satélite) . . . . .	31
2.2.3	Detecção de cupinzeiro (UAV) . . . . .	32
2.2.4	Quantificação de formigueiro (UAV) . . . . .	33
2.2.5	Detecção de formigueiros (Dados espectrais) . . . . .	34
2.2.6	Utilização de Câmeras Multi/Hiperespectrais . . . . .	35
2.2.7	Utilização de método de Visão Computacional e Inteligência Artificial .	36
<b>2.3</b>	<b>Estudo sobre formigas . . . . .</b>	<b>37</b>
2.3.1	Tipos de formigas . . . . .	37
2.3.2	Morfologia externa dos formigueiros . . . . .	39
2.3.3	Melhor momento para detecção . . . . .	40
<b>2.4</b>	<b>Estudo do Drone . . . . .</b>	<b>41</b>
2.4.1	Classificação dos drones . . . . .	42
2.4.2	Visada para baixo . . . . .	45
<b>2.5</b>	<b>Estudo de Visão Computacional . . . . .</b>	<b>47</b>
2.5.1	Aquisição . . . . .	48
2.5.2	Segmentação . . . . .	55
2.5.3	Extração de características . . . . .	57
<b>2.6</b>	<b>Estudo Aprendizado de Máquina . . . . .</b>	<b>69</b>
2.6.1	Aprendizagem supervisionada . . . . .	70
2.6.2	Aprendizagem não-supervisionada . . . . .	70
2.6.3	Descritores Estatísticos . . . . .	70
2.6.4	Base de dados . . . . .	73

2.6.5	Algoritmos baseados em Árvore de Decisão . . . . .	74
2.6.6	Métricas de avaliação de performance . . . . .	79
<b>3</b>	<b>FORMULAÇÃO DO PROBLEMA . . . . .</b>	<b>83</b>
<b>3.1</b>	<b>Possibilidade de entrar no mercado . . . . .</b>	<b>83</b>
<b>3.2</b>	<b>Tamanho a ser buscado . . . . .</b>	<b>85</b>
<b>3.3</b>	<b>Dificuldades enfrentadas . . . . .</b>	<b>85</b>
3.3.1	Altura de voo . . . . .	85
3.3.2	Horários de voo . . . . .	86
3.3.3	Condições climáticas . . . . .	88
3.3.4	Morfologia do formigueiro e preferência de localização . . . . .	89
<b>3.4</b>	<b>Sumário das decisões . . . . .</b>	<b>89</b>
<b>4</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>91</b>
<b>4.1</b>	<b>Aquisição de dados . . . . .</b>	<b>91</b>
<b>4.2</b>	<b>Determinação dos parâmetros do <i>Super Pixel</i> . . . . .</b>	<b>92</b>
4.2.1	Análise fatorial . . . . .	93
<b>4.3</b>	<b>Classificação Manual . . . . .</b>	<b>96</b>
4.3.1	Demeter . . . . .	96
4.3.2	Tratamento dos dados . . . . .	97
4.3.3	Análise gráfica . . . . .	98
<b>4.4</b>	<b>Implementação dos Descritores . . . . .</b>	<b>99</b>
4.4.1	Seleção dos descritores . . . . .	100
4.4.2	Aplicação dos descritores . . . . .	101
<b>4.5</b>	<b>Modelo de classificação . . . . .</b>	<b>102</b>
4.5.1	Versão zero ( <i>v0</i> ) . . . . .	103
4.5.2	Versão um ( <i>v1</i> ) . . . . .	104
4.5.3	Versão dois ( <i>v2</i> ) . . . . .	104
4.5.4	Versão três ( <i>v3.0</i> ) . . . . .	105
4.5.5	Versão três ponto um ( <i>v3.1</i> ) . . . . .	105
4.5.6	Versão quatro ( <i>v4.0</i> ) . . . . .	106
4.5.7	Versão quatro ponto um ( <i>v4.1</i> ) . . . . .	107
<b>5</b>	<b>RESULTADOS . . . . .</b>	<b>109</b>
<b>5.1</b>	<b>Versão zero (<i>v0</i>) . . . . .</b>	<b>109</b>
<b>5.2</b>	<b>Versão um (<i>v1</i>) . . . . .</b>	<b>109</b>
<b>5.3</b>	<b>Versão dois (<i>v2</i>) . . . . .</b>	<b>110</b>
<b>5.4</b>	<b>Versão três ponto zero (<i>v3.0</i>) . . . . .</b>	<b>111</b>
<b>5.5</b>	<b>Versão três ponto um (<i>v3.1</i>) . . . . .</b>	<b>112</b>
<b>5.6</b>	<b>Histogramas dos canais de cores . . . . .</b>	<b>112</b>

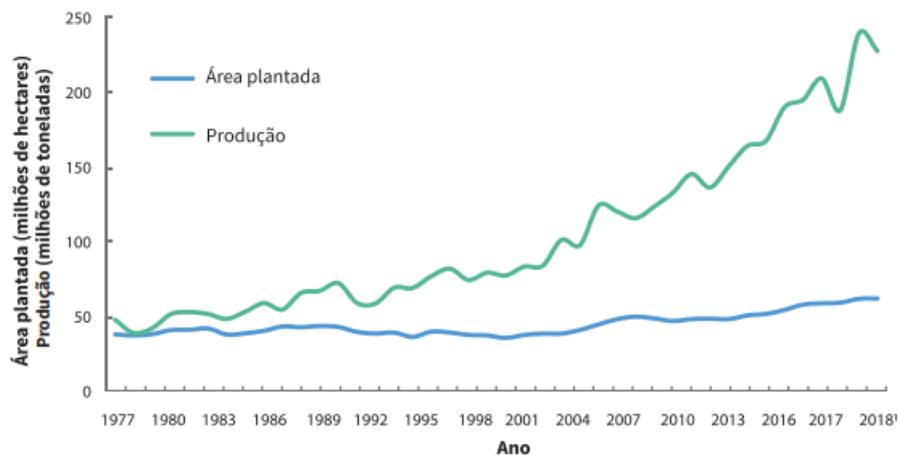
5.7	Versão quatro ponto zero ( <i>v4.0</i> ) . . . . .	115
5.8	Versão quatro ponto um ( <i>v4.1</i> ) . . . . .	116
5.9	Considerações gerais . . . . .	117
5.9.1	Versão extra ( <i>v.extra</i> ) . . . . .	118
6	CONCLUSÃO . . . . .	121
	REFERÊNCIAS . . . . .	123
	APÊNDICE A            MATERIAIS EM DETALHES . . . . .	133
A.1	Drone . . . . .	133
A.1.1	Especificações técnicas . . . . .	133
A.1.2	Autonomia da bateria . . . . .	134
A.1.3	<i>Range</i> de voo . . . . .	134
A.1.4	Estabilidade do <i>Gimbal</i> . . . . .	135
A.1.5	Câmera integrada . . . . .	135
A.2	Linguagem de Programação . . . . .	136
A.2.1	<i>Open Source</i> . . . . .	136
A.2.2	<i>Python</i> . . . . .	137
	APÊNDICE B            BASE DE DADOS E CÓDIGOS . . . . .	141
B.1	GitHub Projeto Tamanduá . . . . .	141



## Introdução

Nos último 40 anos, a produção e produtividade agropecuária brasileira sofreram aumentos significativos. Esse aumento transformou o status do Brasil de importador de alimentos para um provedor em potencial (EMBRAPA, 2018). No começo dessa janela de tempo, a agricultura brasileira era rudimentar, logo a escassez de tecnologia no campo afetava homens e mulheres gerando um baixo rendimento por hectare e baixa produção. De acordo com (EMBRAPA, 2018), nessa época, a produção era tão baixa que não atendia a demanda interna. A partir deste ponto, para aumentar a produtividade, o governo investiu em pesquisa e desenvolvimento, extensão e crédito rural. Entre os indicadores mais ilustrativos da trajetória recente da agricultura brasileira estão os números de produção e os índices de produtividade. Entre 1975 e 2017 a produção de grãos cresceu mais de seis vezes, enquanto a área plantada dobrou de tamanho (EMBRAPA, 2018). É possível notar esse crescimento na Figura 1.

Figura 1 – Indicadores de produção e área plantada de 1975 a 2017.

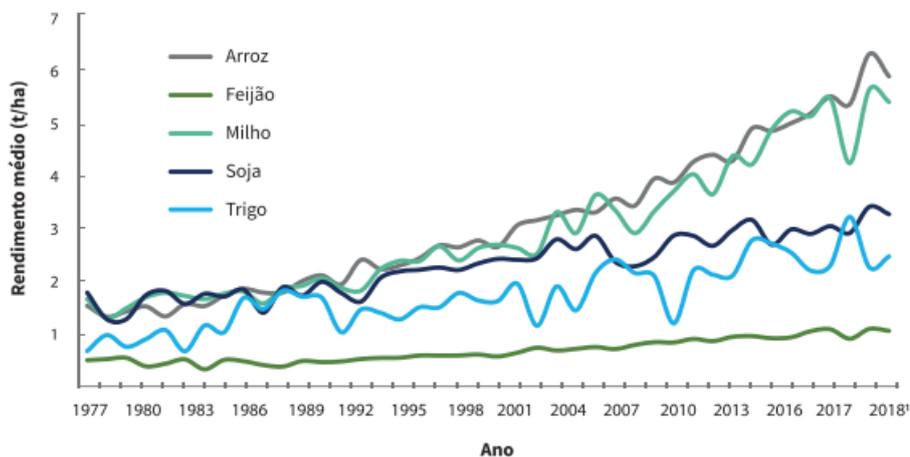


Fonte: (EMBRAPA, 2018)

O maior crescimento da produção em comparação à área pode ser visto por meio da evolução do rendimento médio (quilos por hectare) das lavouras de arroz, feijão, milho,

soja e trigo, no período de 1975 a 2017 (EMBRAPA, 2018). A Figura 2 abaixo mostra a evolução do rendimento médio:

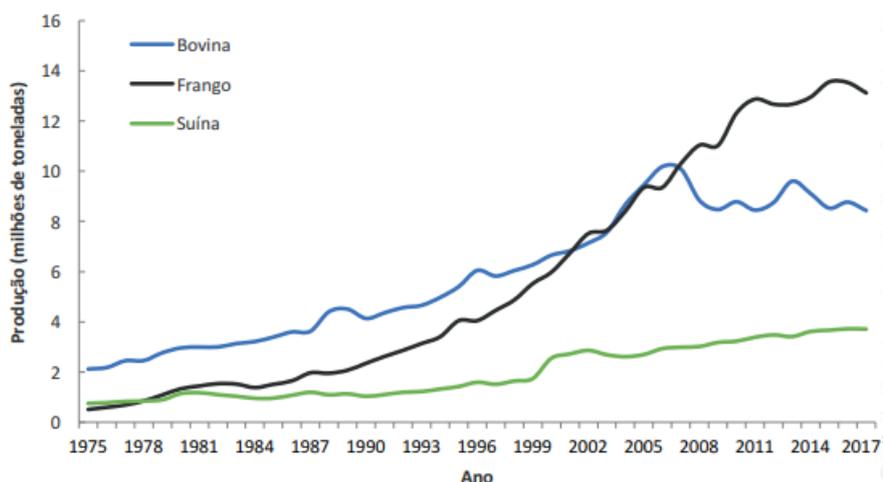
Figura 2 – Crescimento da produção agrícola



Fonte: (EMBRAPA, 2018)

O setor pecuário também sofreu incrementos de produção e produtividade, com destaque para a produção de carne bovina e carne de frango. O número de cabeças de gado no país mais que dobrou nas últimas quatro décadas, enquanto a área de pastagens teve pequeno avanço. O Brasil é atualmente o 2º maior produtor de carne bovina mundial, estando atrás apenas dos Estados Unidos (EMBRAPA, 2018).

Figura 3 – Crescimento da produção pecuária



Fonte: (EMBRAPA, 2018)

Foi a partir da década de 1990 que demandas crescentes e políticas macroeconômicas de estabilização, como controle de inflação e taxas de câmbios menores, impulsionaram ainda mais o crescimento do setor agrícola por meio das exportações (EMBRAPA, 2018).

O agronegócio tem sido reconhecido como um forte aliado no crescimento do Produto Interno Bruto (PIB) brasileiro. Segundo dados do Centro de Estudos Avançados em Economia Aplicada (CEPEA/USP) em parceria com a Confederação da Agricultura e Pecuária do Brasil (CNA), em 2019, a soma de bens e serviços gerados no agronegócio chegou a R\$ 1,55 trilhão ou 21,4% do PIB brasileiro (CNA, 2020). Dentre os segmentos, a maior parcela é do ramo agrícola, que corresponde a 68% desse valor (R\$ 1,06 trilhão), a pecuária corresponde a 32%, ou R\$ 494,8 bilhões.

A trajetória recente da agricultura brasileira é resultado de uma combinação de fatores. Dentre esses fatores, o que mais fez a diferença nestes últimos 50 anos foram os investimentos em pesquisa agrícola. Pode-se chamar de investimento agrícola os avanços nas ciências, tecnologias adequadas e inovações (EMBRAPA, 2018).

Como pode ser notado na Figura 1 acima, a área plantada não acompanhou o crescimento da produção, isso deve-se a pouca existência de terreno para o crescimento horizontal da agricultura no Brasil e no mundo. Em outras palavras, existem fortes razões para que o aumento da produção agrícola ocorra por meio da intensificação dos sistemas de produção e não pelo aumento da área utilizada para cultivo de plantas e criação de animais (EMBRAPA, 2018). A modernização recente da agricultura brasileira, é o principal contribuinte para que haja uma intensificação da produção agrícola sem necessidade de expansão territorial.

Com esses avanços tecnológicos, surgiu-se a agricultura de precisão (AP). Por agricultura de precisão pode-se entender como um conjunto de ferramentas e tecnologias que possibilita ao produtor conhecer toda a área para cultivo de maneira mais completa e trabalhar de maneira mais eficiente. O uso de AP agrega valor por meio do aumento de produtividade, da diminuição de falhas humanas e da redução do trabalho penoso e de riscos operacionais (RURAL, 2015).

A agricultura de precisão é uma abordagem holística e inovadora de sistemas que auxilia os agricultores no gerenciamento da variabilidade da safra e do solo, afim de diminuir custos, melhorar a qualidade e a quantidade do rendimento e aumentar a renda agrícola (PAUSTIAN; THEUVSEN, 2017). A tecnologia de AP é uma ferramenta de gestão para monitorar a eficiência dos insumos de recursos, enquanto reduz o uso de produtos químicos para evitar danos ambientais e produzir produtos de alta qualidade para satisfazer a crescente demanda por alimentos (MOURHIR et al., 2017) (ONYANGO et al., 2021). AP aplica práticas agrícolas tradicionais com novas tecnologias, práticas e impulsionadores econômicos para aumentar a sustentabilidade em um equilíbrio dinâmico (BALAFOUTIS et al., 2017). Estudos relataram resultados positivos da adoção de agricultura de precisão, incluindo economia nos fatores de produtividade (LOURES et al., 2020), aumento da produção e sustentabilidade ambiental (ULLAH et al., 2020), melhoria da segurança alimentar e vitalidade econômica da comunidade em regiões em desenvolvimento (YUE et al., 2016)

A agricultura de precisão tem como objetivo combinar a melhora da produtividade sem deixar a sustentabilidade de lado. O meio utilizado para atingir tal objetivo é justamente o desenvolvimento tecnológico, seja ele por meio de técnicas simples, como alocação de pastagem ou técnicas mais avançadas como sensoriamento remoto.

Feltran concluiu que, por meio de aplicações de AP, a recuperação de 12 milhões de hectares de pastagens degradadas, declaradas pelos produtores brasileiros no último Censo Agropecuário, permitiria ao Brasil elevar o seu rebanho bovino em 17,7 milhões de cabeças de gado – volume equivalente a 1,5 vez o rebanho do Uruguai (FELTRAN-BARBIERI; FÉRES, 2021).

O estudo realizado por Feltran também avaliou os resultados caso parte das pastagens degradadas fossem destinadas para cobrir o déficit de Áreas de Preservação Permanente (APP) e Reserva Legal (RL), cumprindo o que determina o Código Florestal. Nesse caso, seria possível adicionar uma capacidade suporte de 9 milhões de cabeças (ou 4,9% do rebanho bovino ou 70% do rebanho urugauio) e ainda garantir 12,7 milhões de hectares para restauração.

O sensoriamento remoto é geralmente considerado uma das tecnologias mais importantes para a agricultura de precisão. Ele é comumente utilizado para monitoramento de campos cultivados, proporcionando soluções eficazes (MULLA, 2013). O sensoriamento remoto pode monitorar muitas culturas e parâmetros da vegetação por meio de imagens. No passado, o sensoriamento remoto era frequentemente baseado em imagens de satélite ou imagens adquiridas usando aeronaves tripuladas (CIPRA; JAN, 1973) (MORA et al., 2017). No entanto, as imagens de satélite muitas vezes não são a melhor opção devido à baixa resolução espacial das imagens adquiridas e baixa resolução temporal. Além disso, muitas vezes, é necessário esperar longos períodos entre a aquisição e a recepção das imagens. Comparado com a utilização de aeronaves tripuladas, o processo geralmente resulta em custos elevados e, muitas vezes não é possível realizar vários voos para obter mais do que algumas imagens recortadas.

O uso de Veículos Aéreos Não Tripulados (VANTs) para sensoriamento remoto oferece a possibilidade de adquirir os dados de campo de uma forma fácil, rápida e econômica se comparado com os métodos citados anteriormente. A constante evolução das câmeras para obtenção de melhores resolução de imagem, em conjunto com o fato dos VANTs voarem em baixas altitudes também contribui para melhores resoluções nas imagens obtidas, além da alta resolução temporal. Além disso, os VANTs são muito mais simples de usar e também mais baratos do que as aeronaves tripuladas. Seu uso também é mais eficiente que os sistemas terrestres, pois promovem uma varredura de campo em um curto espaço de tempo (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

Sistemas Aéreos Não Tripulados (*Unmanned Aerial Systems - UAS*) são, agora, usados em aplicações de sensoriamento remoto para Agricultura de Precisão, servindo a uma infinidade de propósitos, como, por exemplo, monitoramento de saúde e detecção de

doenças, monitoramento de crescimento e estimativa de rendimento, manejo e detecção de ervas daninhas, entre outros. (ZHANG; KOVACS, 2012) (PURI; NAYYAR; RAJA, 2017) (MOGILI; DEEPAK, 2018) (YANG et al., 2017) (KULBACKI et al., 2018).

Visão computacional e aprendizado de máquina têm sido frequentemente utilizados na agricultura de precisão, juntamente com imagens de UAV (HUNG; XU; SUK-KARIEH, 2014) (PAPAGEORGIOU; MARKINOS; GEMTOS, 2011) (ZHENG; SONG; CHEN, 2013) (PÉREZ-ORTIZ et al., 2015). Já foi relacionado acima o uso da agricultura de precisão com a robótica área, agora agregam-se novas ferramentas que vem ganhando cada vez mais destaque, tanto no mercado, quanto no setor científico.

Visão computacional e aprendizado de máquina operam em conjunto, no desenvolvimento de algoritmos e realização de tratamentos, afim de extrair informações de imagens por meio dos *pixels* que a compõe (TRUCCO; VERRI, 1998).

A primeira definição de "*Machine Learning*" foi dada por Arthur Samuel em 1959 que era "o aprendizado de máquina como tecnologia que dá aos computadores a capacidade de aprender sem serem explicitamente programados".

No momento em que vivemos, em que cada vez mais o tempo é um fator determinante, o uso de *machine learning* e visão computacional se torna um aliado nos negócios. Por meio de aprendizado de máquina, o sistema auxilia os empresários na análise de dados para a tomada de decisões, de maneira mais rápida e eficiente, visando sempre o que atende as necessidades específicas de cada caso (NOTÍCIAS... , 2019).

Segundo uma pesquisa divulgada em 2019 pela IDC Global (IDCGLOBAL, 2021), o mercado de *Machine Learning* teria um crescimento anual de 46,2%, chegando a U\$ 52 bilhões no ano de 2021. Tendo em vista que o desenvolvimento dessa pesquisa científica visa auxiliar no desenvolvimento de uma nova ferramenta, a tendência de mercado também deve ser considerada quando as escolhas são feitas.

A fim de verificar o interesse na área de *remote sensing*, agricultura de precisão e *machine learning*, foi realizada uma análise bibliométrica que será abordada em sessões seguintes.

As formigas cortadeiras dos gêneros *Atta* e *Acromyrmex* são consideradas pragas para as plantações. Esse atributo das formigas cortadeiras é resultado de seus hábitos alimentares, que consistem em cortar e transportar, para seus ninhos, os fragmentos de plantas para o cultivo do fungo simbiótico *Leucoagaricus gongylophorus* utilizado em sua alimentação (ARENAS; ROCES, 2017).

Formigas cortadeiras podem causar perdas direta e indiretamente, que podem atingir bilhões de dólares em todo o mundo (MUTANGA; DUBE; GALAL, 2017). Um único ninho do gênero *Atta* pode cortar cerca de 5 kg de material vegetal por dia (JAFFE, 2010), 89 ninhos do gênero *Atta* podem coletar 825 kg de biomassa vegetal por hectare (COSTA et al., 2008).

Dentre os 11 tipos diferentes de formigas cortadeiras encontradas no Brasil, uma das

que mais chamam atenção são as Saúvas Pardas (*Atta capiguara*). Tal espécie de formiga cortadeira é um dos insetos que mais causam danos nas pastagens do sudeste do Brasil, competindo com o gado pelo capim e reduzindo consideravelmente a capacidade dos pastos, quando não controlada (BATISTA; FILHO; LOUREIRO, 1985).

Segundo Amante, a presença de 10 colônias de *A. capiguara* por hectare, consome aproximadamente 52,5 kg de capim por dia, o que equivale à ração diária de 3 bois em regime de pasto (AMANTE, 1967b). Com base nesses dados, um boi come cerca de 17,5kg de pasto por dia, e um formigueiro adulto degrada 5,25kg de pasto por dia, um formigueiro adulto "come" o equivalente a 0,3 boi por hectare. Esse valor é o número de cabeças de gado perdida por hectare para cada uma colônia presente no hectare .

De acordo com o Censo Agropecuário de 2017, o número de cabeças de gado por hectare é igual a 1,15. Então um formigueiro adulto comer o equivalente a 0,3 cabeça de gado por hectare implica em uma queda de produção de quase um quarto.

Considerando o número de cabeças de gado perdida por hectare sendo 0,3, o número de formigueiro por hectare igual a um, a cotação do arroba no valor de R\$291,60 e o peso mínimo do boi sendo 200kg (SAÚDE, 2021), estima-se um prejuízo de, no mínimo, R\$1166,40 por hectare.

De posse dessas informações, o presente trabalho foi conduzido no intuito de se estudar e implementar um algoritmo de detecção de ninhos de formiga da espécie *Atta* em ambientes pastoril, utilizando visão computacional de imagens RGB feitas por drones do tipo quadricóptero.

Este trabalho encontra-se organizado da seguinte maneira: O segundo capítulo apresenta uma revisão da literatura, afim de fornecer ao leitor uma base teórica sobre o assunto e as técnicas que serão aplicadas. O terceiro capítulo aborda as dificuldades enfrentadas e os motivos de decisões tomadas. O quarto capítulo apresenta os materiais e métodos utilizado, para que o trabalho possa ser reproduzido por outros pesquisadores. O quinto capítulo trás os resultados obtidos e uma análise dos mesmos . E o sexto e último capítulo apresenta a conclusão desse estudo.

---

## Estado da Arte e Referencial Teórico

### 2.1 Análise bibliométrica

Com o intuito de verificar o estado da arte e realizar uma revisão da literatura, foram feitas duas análises bibliométricas em anos diferentes. A análise bibliométrica consiste em uma busca sistemática na base de dados de artigos científicos. Dentre as duas bases mais famosas, *Scopus* e *IEEE*, optou-se por realizar essa busca na base *Scopus*, considerando que a mesma apresentou uma quantidade maior de resultados.

A análise consistiu na realização de duas pesquisas com termos distintos, afim de que uma fosse mais abrangente e a outra mais restrita e direcionada. As palavras chaves utilizadas para a primeira pesquisa foram: "*Remote Sensing*", "*Observation*", "*Monitoring*", "*Mapping*", "*Detection*", "*Prague*", "*Pragues*", "*Pest*" e "*Pests*". Já as palavras chaves da segunda pesquisa foram: "*Remote Sensing*", "*Observation*", "*Monitoring*", "*Mapping*", "*Detection*", "*Ant Mound*", "*Termite Mound*", "*Ant Nest*" e "*Termite Nest*". Nas duas pesquisas as áreas de estudo foram delimitadas em Ciência da Computação e Engenharia, foi excluído o termo "gen\*" e delimitado para que apenas textos em língua inglesa fossem considerados.

As primeiras análises foram realizadas em dezembro de 2020 e todos os periódicos encontrados, com as delimitações citadas acima, foram considerados para as etapas seguintes. As buscas retornaram 1418 e 11 resultados, respectivamente.

As segundas análises foram realizadas em maio de 2023 e da mesma maneira, todos os periódicos encontrados, com as delimitações citadas acima, foram considerados para as etapas seguintes. Essas buscas retornaram 2285 e 14 resultados, respectivamente.

Tabela 1 – Comparação dos resultados da Análise Bibliométrica realizada em anos diferentes

Data da busca	Quantidade de artigos (Busca mais abrangente)	Quantidade de artigos (Busca mais focada)
Dezembro 2020	1418	11
Mai 2023	2285	14

Fonte: Autor.

Com base na observação dos resultados obtido, é possível constatar que, apesar o alto interesse na área de *remote sensing* para fins agrícolas, pouco se estuda a aplicação do mesmo na detecção de ninho de formiga. Nota-se também que o interesse geral da área aumentou aproximadamente 61% no período de dois anos e meio, já os estudos mais específicos cresceram apenas, aproximadamente, 27% no mesmo período.

Assim, supõe-se que, da mesma maneira que há muito a ser estudado sobre este tema, não é uma tema fácil de ser desenvolvido.

## 2.2 Estado da arte

Com base na análise bibliométrica realizada foi possível observar a presença de uma lacuna na intersecção dos assunto "Sensoriamento Remoto", "Agricultura de precisão" e "Ninho de formiga". Nessa seção serão abordados os poucos estudos encontrados nessa área, a fim de apresentar o estado da arte de maneira um pouco mais detalhada.

### 2.2.1 Detecção de formigueiro (Satélite)

Em 2004 Vogt realizou uma pesquisa que tinha como objetivo a detecção de formigueiros utilizando imagem de satélite. As imagens foram obtidas por meio da *Digital Globe Geographic Information Products* e possuíam características pancromáticas e de resolução 0,65m. A câmera utilizada na aquisição das imagens não foi especificada pelo autor. O processamento dessas imagens foi dividido em duas etapas principais. Etapa 1: Trabalhadores examinaram os lotes em busca dos montes de formigas e os montes encontrados foram medidos e georreferenciados. Os dados dessa etapa foram obtidos em um período de duas semanas. Etapa 2: Os contornos dos gráficos foram sobrepostos aos dados das imagens e examinados visualmente a fim de encontrar detalhes que se assemelhavam a formigueiros, cada monte suspeito foi marcado e os dados de sua localização foram sobrepostos a imagem para verificar a precisão da detecção. Para prever a probabilidade de observar um monte, um modelo foi desenvolvido utilizando a atividade do monte para efeito de classificação fixa, cobertura de vegetação, área e altura do monte como cova-

riáveis. A fim de obter uma estimativa das diferenças na refletividade entre os montes e a superfície do solo/vegetação, foi aplicada uma equação que aumentava o contraste entre os *pixels* das imagens. No pós processamento, os montes apareceram nas imagens como pontos escuros contra um fundo claro, ou vice versa. O modelo proposto apresentou uma média de 46,9% de montes detectados. Algumas considerações foram feitas ao final do estudo, e foram elas: Montes maiores eram mais prováveis de serem identificados; fatores como tipo e umidade do solo podem influenciar na detecção; o horário do dia que a aquisição dos dados foi feita influencia nas sombras presentes nas imagens (o que pode gerar falso positivo); a fotointerpretação não é a melhor técnica para esse tipo de aplicação, devido a sua subjetividade como uma ferramenta analítica; e é indispensável ter um conhecimento prévio da aparência dos formigueiros, e uma compreensão de como as mudanças nas condições ambientais podem afetar a atividade de construção dos formigueiros. Por fim, o autor concluiu que a falsa detecção de montes menores faz com que as imagens de satélite existentes na época do estudo sejam inadequadas para aplicação onde se faz necessária uma alta precisão de amostragem. Essa falsa detecção pode ser proveniente da falta de resolução nas imagens, como também da falta de treinamento do modelo, fazendo necessário estudos adicionais para verificação da utilização de outras técnicas de processamento de imagem e avanços na tecnologia na área de sensores espaciais (VOGT, 2004a).

### 2.2.2 Mapeamento de desfolha (Satélite)

No estudo direcionado por Santos em 2020, ele buscava realizar, por meio de imagem de satélite, o mapeamento da desfolha causadas por formigas. As imagens utilizadas por ele foram obtidas por meio de um satélite S2 tipo nível 1C e adquiridas com um instrumento multiespectral. A câmera do satélite não foi especificada pelo autor. O processamento das imagens foi dividido em 5 etapas. Etapa 1: Os ninhos foram localizados e medidos manualmente, enquanto o georreferenciamento foi feito com um GPS. Na Etapa 2 foi realizado o pré processamento, em que de cada imagem foram extraídos os valores de refletância dos *pixels* com e sem a presença de ninhos. Para fins de efeitos topográficos, as imagens passaram por um processo de correção e quatro índices vegetativos foram calculados a partir dos dados de refletância das bandas. No último passo da etapa 2, as imagens com contorno georreferenciado foram recortadas resultando em novas imagens de refletância da copa dos eucaliptos (para cada uma das bandas espectrais e índices vegetativos). Etapa 3: Nessa etapa foi realizada correções e processamentos por meio de sete softwares. Etapa 4: Foi realizada uma análise multivariada de variância permutativa por meio de análise dos valores de refletância para cada uma das bandas espectrais, a fim de verificar se havia diferença entre a refletância dos *pixels* com a presença ou ausência de formigueiros. Etapa 5: Na última etapa foi realizada a predição, e para a predição das áreas atacadas foram utilizadas quatro bandas multiespectrais e quatro índices vegeta-

tivos em um algoritmo de regressão de mínimos quadrados parcial discriminante e para prever as áreas atadas (ou não), as variáveis preditivas foram avaliadas pela importância da variável na projeção. Como resultado, a previsão do modelo apresentou um erro amostral médio de 15,4% quando utilizados 4 bandas e 4 índices de vegetação e com base nas curvas de refletância do dossel desfolhado, a caracterização espectral mostrou que o modelo performou bem a estimativa da presença de formigas cortadeiras nas plantações de eucalipto (SANTOS et al., 2020).

### 2.2.3 Detecção de cupinzeiro (UAV)

Sandino tinha como objetivo detectar montes de cupim, e para isso, diferente de Vogt e Santos que utilizaram imagens de satélites, ele utilizou imagens adquiridas por meio de drone. O VANT utilizado por ele era um hexa-rotor DJI S800. A câmera utilizada na aquisição das imagens foi uma câmera hiperespectral, usada a fim de gerar informações sobre o espectro de cada um dos *pixels* da região alvo do estudo; e uma câmera digital, usada para capturar imagens de alta resolução a fim de identificar por fotointerpretação os cupinzeiros e salvar suas coordenadas GPS. O método de processamento das imagens foi dividido em quatro etapas. Etapa 1: Após a aquisição dos dados foi realizada uma ortorretificação dos dados hiperespectrais, em seguida, foram extraídas várias regiões de interesse por meio de verificação das coordenadas das imagens digitais. Cada região de interesse foi processada e os métodos utilizados foram: filtragem espacial e espectral, recuperação de refletância, recuperação e classificação de materiais por Máquina de vetores de suporte (SVM), *Linear Spectral Unmixing* e *Principal Component Analysis*. Etapa 2: Nessa etapa foi realizado um pré processamento para obtenção dos cubos hiperespectrais e implementação de técnicas de correção de radiância e ortorretificação. Etapa 3: Baseado em uma biblioteca, que possuía um conjunto de assinaturas de materiais, foram executados os algoritmos proposto. Após a comparação dos resultados dos três algoritmos, apenas o classificados SVM foi utilizado nos testes. Etapa 4: A etapa de detecção foi realizada com base nas análises de formato e de conteúdo das camadas de cores que foram geradas em etapas anteriores. O processo é iniciado pela manipulação das camadas de cores binárias. A fim de atenuar os ruídos, foram aplicadas filtragens e técnicas morfológicas, com isso, descobriu-se que é possível identificar os cupinzeiros a partir de outros objetos, calculando a distância entre suas camadas. Uma imagem temporária é calculada (por meio de uma operação lógica *AND* entre duas camadas) com o objetivo de verificar se a região de interesse inclui algum monte de cupim, se a intensidade média, definida previamente por uma equação, não for atendida, a imagem será descartada e classificada como "nenhum cupinzeiro encontrado". O algoritmo proposto por Sandino apresentou uma acurácia de 68,0%, e baseado nisso algumas considerações foram feitas: Em decorrência da adição de um procedimento de filtragem dos contornos, os resultados tornaram-se dependentes dos valores da resolução espacial e propriedades de distorção

das imagens; em decorrência do processo de ortorretificação, pequenos cupinzeiros não foram identificados; entretanto, independente das propriedades de localização e rotação, os cupinzeiros de tamanho grandes e médios foram identificados com sucesso. A principal contribuição do algoritmo proposto é a integração de métodos de processamento de imagem hiperespectral para classificar materiais com o processamento de imagens RGB para segmentar os cupinzeiros (SANDINO; WOOLER; GONZALEZ, 2017)

#### 2.2.4 Quantificação de formigueiro (UAV)

Também em 2004, Vogt realizou um estudo de quantificação de formigueiro utilizando aeronaves tripulada (Cessna 172 *aircraft*). Foi instalado e integrado um sistema de coleta de dados que possuía uma antena GPS e um receptor de 12 canais. Para as taxas de aceleração e rotação dos eixos das câmeras foi utilizada uma Unidade de Medição Inercial. O sistema de câmeras consistia em quatro câmeras digitais monocromáticas (uma para cada canal RGB e uma Infra-vermelha). O método de processamento de imagem utilizado foi dividido em cinco etapas. Etapa 1: Essa primeira etapa consistia em gerar dois ortomosaicos georreferenciados, um em RGB e outro infravermelho. Etapa 2: Foi aplicado um único processamento de imagem nos mosaicos, um histograma linear de 2% que teve como finalidade aumentar o contraste. Etapa 3: O modelo utilizado para análise nessa etapa foi a fotointerpretação. Etapa 4: Trabalhadores realizaram uma busca por formigueiros em todas as áreas de estudo, os formigueiros encontrados foram medidos, classificados como ativo ou inativo, georreferenciados e cada um recebeu um número único. Etapa 5: Os dados foram examinados afim de verificar se a aproximação normal da distribuição abinomial foi atendida e as análises foram conduzidas assumindo distribuição normal dos valores médios previstos. Afim de obter uma tendência linear na probabilidade de detectar um monte, foram adicionadas, individualmente ao modelo, as variáveis de: tamanho, altura e volume monte e cobertura de vegetação. O maior porcentual de detecção foi de 75% no mês de maio, com imagens infravermelho e resolução de 0,1m. A variação de todas as combinações de cores, resolução e data revelou tendências não detectáveis. Da mesma maneira as interações entre área dentro da pastagem, atividade do monte, resolução da imagem e cor não revelou nenhuma interação em potencial. Com base nos resultados obtidos, o autor concluiu que as diferenças sazonais no tamanho médio e condições dos formigueiros são fatores importantes a serem considerados, o mês de realização da detecção impacta na porcentagem de ninhos detectados com sucesso e devido a muitos fatores influenciarem que a detecção (como tipo de solo e vegetação) e essas características variarem entre os locais, o valor preditivo das regressões geradas são limitadas a um único local (VOGT, 2004b)

### 2.2.5 Detecção de formigueiros (Dados espectrais)

Por fim, Wu 2014 baseou sua pesquisa na detecção de formigueiros utilizando dados espectrais. Ele fez uso de um gerador de imagens hiperespectrais que tem faixa espectral de 400 a 1000nm, com resolução espectral de 3 a 10nm. Seu sistema de detecção baseado em informações espectrais, coletou os dados (imagens) para determinar se as assinaturas espectrais poderiam ser definidas, com o objetivo de distinguir os montes de outras características ao redor dele. Sua metodologia foi dividida em três etapas. Etapa 1: Solo do ninho, areia e grama foram colocados separadamente no gerador de imagem hiperespectral afim de obter os dados espectrais de cada material. As refletâncias obtidas foram comparadas no campo. Etapa 2: Para determinar a relação entre a refletância do solo do ninho e o teor de umidade do solo foram realizadas várias etapas de pesagem, aquecimento e pulverização do solo. Etapa 3: Para caracterizar as diferenças entre solos irregulares e granulares, suas reflexões espectrais foram testadas várias vezes usando o gerador de imagens hiperespectral. A variação nos valores de refletância foram analisadas usando o Modelo Linear Geral. Quando os resultados foram significativos, comparações múltiplas de média foram realizadas por análise *post-hoc* de Tukey HSD. Uma regressão linear foi utilizada para analisar a correlação entre o valor da refletância e o teor de água no solo. Os resultados obtidos também foram divididos em etapas. A etapa 1 consistiu em comparar a refletância do solo do ninho com a refletância dos solos ao redor. O espectro de refletância do solo do ninho e do solo comum foram semelhantes, a refletância da areia e da grama eram significativamente diferentes e as diferenças desses dois para o solo do ninho e solo comum eram evidentes. Com base nessas informações foi definida uma banda espectral para distinguir ninhos dos substratos circundantes. Na etapa 2 foi verificada a relação entre a refletância e a umidade do solo. As porcentagens de refletância do solo de ninho seco foram aproximadamente 10% a 15% maiores do que as das amostras úmidas. Na etapa 3 foi comparado os agregados de solo de ninho e agregados de solo comum. O solo do ninho apresentou partículas menores, mais lisas e mais uniformes do que o solo ao redor e compara a refletância do solo do ninho granular com a do solo comum com torrões, observou-se que a refletância do solo com torrões era 5% a 10% maior do que a do solo do ninho (WU et al., 2014).

Na Tabela 2 abaixo, pode-se observar de maneira resumida as diferenças entre os cinco trabalhos apresentados acima.

Tabela 2 – Estado da arte

Foco da detecção	Equipamento utilizado	Câmera utilizada	Resultados	Autor, Ano
Formigueiro	Satélite	N/D	Acurácia de 46.9%	Vogt, 2004
Desfolha	Satélite	Instrumento multiespectral	Erro médio amostral de 15.4%	Santos, 2020
Cupinzeiro	UAV	Hiperespectral e Digital	Acurácia de 68%	Sandino, 2017
Formigueiro	Aeronave tripulada	Digitais monocromaticas e Infravermelha	Acurácia de 70% no mês de Maio	Vogt, 2004
Formigueiro	Dados Espectrais	Gerador de imagens multiespectrais	Acurácia não apresentada	Wu, 2014

Fonte: Autor

### 2.2.6 Utilização de Câmeras Multi/Hiperespectrais

Com esse estudo do estado da arte pode-se observar um uso frequente de câmeras multi/hiperespectrais. Sendo assim, vale estudá-las também, afim de entender o motivo dessas escolhas.

O funcionamento básico por trás dos sistemas HSI (Hiperespectrais) e MSI (Multiespectrais) é o conceito de assinatura espectral. De maneira simplificada, todos os materiais refletem, emitem, transmitem ou absorvem radiação eletromagnética com base na sua estrutura física, composição química e comprimento de onda proveniente da radiação. Portanto, a quantidade de radiação eletromagnética refletida, emitida, transmitida ou absorvida de cada material varia com o comprimento de onda da radiação. Desse modo, a assinatura espectral desse material será a curva resultante da refletância ou emissividade plotada em uma faixa de comprimentos de onda (WITTENSTEIN, 2007).

As câmeras HSI e MSI conseguem produzir imagens em que cada pixel têm informação espectral relevante para detalhar a cena registrada. Porém, elas se diferem em alguns aspectos, como por exemplo, número de bandas, largura de cada banda espectral e se as bandas são vizinhas em uma região ou se possuem intervalos (WITTENSTEIN, 2007).

- HSI: Coleta pelo menos 100 bandas espectrais, pode ser dividido em grupos de largura de banda (as bandas não são necessariamente vizinhas), possuem capacidade ampla de discriminação espectral;

- MSI: Coleta menos de 20 bandas espectrais, geralmente as bandas não são vizinha, fornece bandas que detectam informações em combinações específicas das regiões desejáveis do espectro.

Entre as vantagens do uso de câmeras multi/hiperespectrais temos que as câmeras hiperespectrais e multiespectrais capturam cenas em muitos comprimentos de ondas e em amplas faixas de espectro, até mesmo em bandas invisíveis aos olhos humanos, como por exemplo a banda infravermelha (IR), e elas fornecem informações muito completas sobre as cenas capturadas e isso inclui informações sobre objetos que podem estar ocultos

ou até mesmo objetos que não são visíveis ao olho humano. Essa capacidade permite um entendimento muito profundo de vários objetos da cena/imagem (MOGHADAM; SHARMA; HEFEEDA, 2021).

Já as desvantagens ficam por conta de problemas de interpretação computacional dos dados multi/hiperespectrais. Devido ao grande número de canais e a esparsa distribuição dos dados, torna-se difícil a modelagem a medida que a dimensionalidade do espaço aumenta (SIGNORONI et al., 2019), paralelamente o conjunto dos dados multi/hiperespectrais possuem tamanhos grandes, e isso faz com que o volume de dados acumulados aumente muito rápido dificultando o processo de aquisição, armazenamento e transmissão dos dados (SHEN; JIANG; PAN, 2018) e considerando que um processo automatizado pode ser executado nas imagens, o desempenho final do sistema que faz uso dessas imagens HSI e MSI dependem da qualidade dos sistemas de processamento (*hardware* e *software*) dos dados coletados (WITTENSTEIN, 2007), além disso, trabalhar com dados HSI e MSI requer tratamento cuidadoso das informações obtida, pois devido a grande quantidade de dados, as assinaturas espectrocópicas podem ficar ocultas.

### 2.2.7 Utilização de método de Visão Computacional e Inteligência Artificial

Baseado na análise bibliométrica realizada foi possível notar o crescente interesse na utilização de Visão Computacional e Inteligência Artificial, entretanto, ainda não aplicados a detecção de formigas.

Analisando os resultados obtidos utilizando as palavras chaves "*Prague*", "*Pragues*", "*Pest*", "*Pests*" ou "*Detection*" e "*Remote Sensing*", "*Observation*", "*Monitoring*" ou "*Mapping*", foram encontrados 1558 artigos. Sendo os mais proeminentes Barbedo 2019, Mogili 2018, Lee 2010, Li 2019 e Yuan 2016 .

Barbedo é um *review* sobre o uso de VANTs na agricultura. Ele tem como base 100 artigos já publicados e pesquisas próprias, nele é abordado os desafios já enfrentados e as principais lacunas que ainda existem, além de sugestões para pesquisas futuras (BARBEDO, 2019).

Mogili faz uma revisão sobre a implementação de VANTs para monitoramento de safra e pulverização de pesticidas (MOGILI; DEEPAK, 2018).

Lee apresenta uma revisão de tecnologias de sensoriamento remoto e discute como elas são usadas na agricultura de precisão (LEE et al., 2010).

Li apresenta um estudo de reconhecimento de imagem que faz utilização de VANTs para captura das imagens RGB e para o processamento faz uso do algoritmo de *superpixel* combinado com método de equalização de histograma (LI et al., 2019).

Yuan propõe um método de detecção de pragas florestais que faz uso de VANT para aquisição de imagens RGB, de *superpixel* como método de processamento e *random forest*

como método de classificação (YUAN; HU, 2016).

## 2.3 Estudo sobre formigas

Um estudo sobre as formigas foi realizado a fim de compreender os tipos de formigas cortadeiras, e a relação entre o gênero da formiga e a morfologia externa dos ninhos.

Os ninhos das formigas cortadeiras são constituídos por duas partes, uma parte interna onde são encontradas as câmaras e os canais e uma parte externa proveniente do solo retirado na construção das escavações (LUCIA, 2011). Para o presente estudo a parte externa é mais analisada, dado que a detecção do ninho será realizada por meio de imagens feitas com um drone.

### 2.3.1 Tipos de formigas

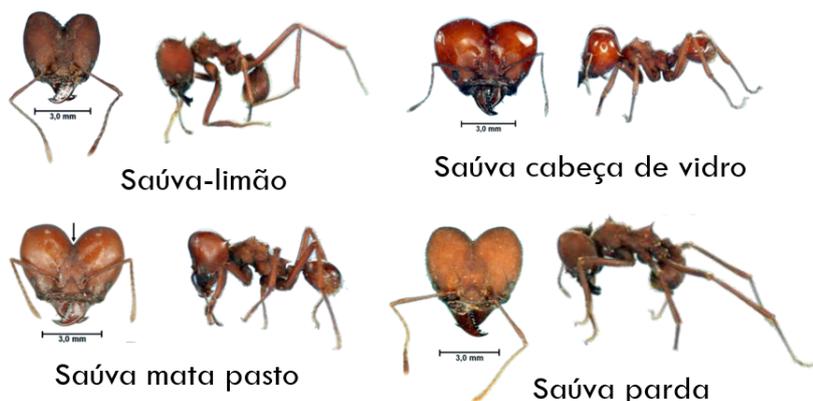
Diferente dos demais gêneros das formigas da tribo *Attini*, os gêneros *Atta* e *Acromyrmex* são consideradas as verdadeiras formigas cortadeiras, pois todas as suas espécies utilizam partes frescas (verdes) das plantas para produzir o fungo da qual se alimentam (LUCIA, 2011).

Apesar de ambas serem consideradas verdadeiras formigas cortadeiras, elas se diferem morfológicamente em alguns aspectos. Essas diferenças são importantes na identificação de cada uma para realização do manejo adequado. Na Tabela 3, podem ser encontradas algumas dessas diferenças e seus gêneros e espécies correspondentes. E nas Figura 4 e Figura 5 podem ser visualizadas essas diferenças morfológicas

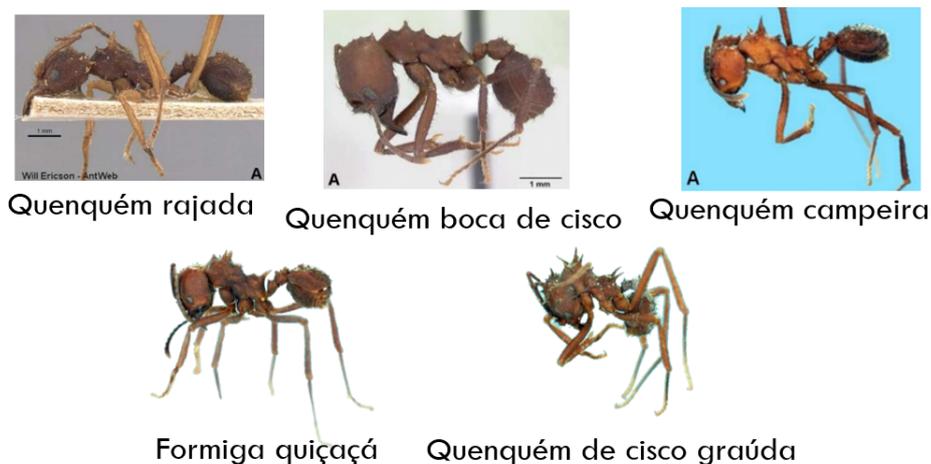
Tabela 3 – Espécies de formigas e suas diferenças morfológicas

Espécie	Nome popular	Diferenças
<i>Atta sexdens</i>	Saúva limão	Cabeça sem brilho e com pelos avermelhados
<i>Atta laevigata</i>	Saúva cabeça de vidro	Cabeça brilhante (aparência de vidro)
<i>Atta bisphaerica</i>	Saúva mata pasto	Cabeça dividida em dois hemisférios e mais opaca em relação a <i>Atta laevigata</i>
<i>Atta capiguara</i>	Saúva parda	Cabeça semelhante a da <i>Atta sexdens</i> ; Caracteres morfológicos de difícil reconhecimento;
<i>Acromyrmex aspersus</i>	Quenquém rajada	Mandíbula alongada
<i>Acromyrmex balzani</i>	Quenquém boca de cisco	Mandíbula curta; Olhos não salientes
<i>Acromyrmex laticeps</i>	Quenquém campeira	Mandíbula alongada; Tubérculos do gárter dispersos
<i>Acromyrmex rugosus</i>	Formiga quiçaçá	Mandíbula alongada; Tubérculos do gárter em 4 séries
<i>Acromyrmex subterraneus</i>	Quenquém de cisco graúda	Mandíbula alongada; Olhos grandes

Fonte: Adaptado de LUCIA, 2011

Figura 4 – Morfologia das formigas do gênero *Atta*

Fonte: Autor.

Figura 5 – Morfologia das formigas do gênero *Acromyrmex*

Fonte: Autor.

Além das diferenças morfológicas de seus corpos, elas também possuem diferenças com relação a morfologia externa de seus ninhos.

### 2.3.2 Morfologia externa dos formigueiros

A parte externa do ninho não é tão visível no caso das formigas do gênero *Acromyrmex*. A morfologia externa do ninho das *Acro.* variam muito de acordo com a espécie. Algumas espécies fazem seus ninhos superficiais cobertos por galhos, folhas, palha ou outro tipos de resíduos vegetais, enquanto outras constroem subterraneamente (sendo cobertos por terra solta) ou de maneira que não seja perceptível a terra escavada. Essa maneira "escondida" de construir e manter seus ninhos, dificulta a localização e controle desse gênero (LUCIA, 2011).

Essa parte externa é bem visível para as formigas do gênero *Atta* e é conhecida como murundu, ou monte de terra solta, e ela se caracteriza por possuir uma quantidade elevada de orifícios que levam a parte interna (LUCIA, 2011).

A forma do murundu é uma das primeiras características observadas quando pretende-se identificar as diferentes espécies. Além da forma do monte de terra solta, outro parâmetro observado é a localização do ninho (LUCIA, 2011). Na Tabela 4 abaixo pode-se identificar a preferência de localização e deposição de terra solta para algumas espécies de formigas cortadeiras do gênero *Atta*. E na Figura 6 podem ser visualizadas essas diferenças na morfologia externa dos ninhos.

Tabela 4 – Diferenças na morfologia externa dos ninhos

<b>Espécie (Nome Popular)</b>	<b>Local</b>	<b>Deposição</b>
<i>Atta sexdens</i> (Saúva limão)	Áreas sombreadas	Terra solta depositada de maneira irregular (com aparência de vulções)
<i>Atta laevigata</i> (Saúva cabeça de vidro)	Áreas sombreadas e abertas	Monte de terra geralmente convexo
<i>Atta bisphaerica</i> (Saúva mata pasto)	Áreas abertas (pastagem e canaviais)	Superfície de terra solta com formato irregular e aparência de compactada
<i>Atta capiguara</i> (Saúva parda)	Áreas abertas (pastagem e canaviais)	Composta por vários montes, sendo o maior uma zona morta e os menores (rosetas) zonas vivas

Fonte: Adaptado de LUCIA, 2011

Figura 6 – Morfologia externa dos ninhos



Saúva limão



Saúva cabeça de vidro



Saúva mata pasto



Saúva parda

Fonte: Autor.

### 2.3.3 Melhor momento para detecção

Segundo James, o mês mais propício para detecção dos ninhos é Maio. Entretanto, a análise realizada por ele foi feita no estado do Mississippi - EUA e, segundo suas próprias conclusões, muitos fatores podem influenciar a detecção dos montes, como o tipo de solo,

vegetação e época de chuva. Esses fatores variam entre os locais, então os resultados obtidos por ele estão limitados a um único local (VOGT, 2004b).

Além disso, o tamanho dos ninhos não é constante, isso quer dizer que, mesmo em um período do ano favorável, se for um ninho jovem, o período chega a ser irrelevante comparado as outras dificuldades enfrentadas na detecção.

Na literatura não foi determinado nenhum horário fixo que seja o melhor para a detecção. Entretanto, algumas técnicas podem ajudar a encontrar esse horário em testes experimentais, e são elas:

- Horário de atividade do ninho: Segundo Caldato, os horários de maior atividade das formigas apresentam dois picos: Um no início do dia e outro no final do dia (CALDATO et al., 2016). Há a possibilidade, que será estudada, de que essa atividade das formigas ajudem a determinar um melhor horário para detecção. Entretanto não se sabe ainda como, dado que as formigas são muito pequenas e, dependendo da altura de voo, não se tem resolução de imagem para visualizar as mesmas;
- Horário de melhor condição climática: É possível que horários do dia em que há maior incidência de luz permitam uma resolução melhor de imagem, possibilitando uma melhor detecção. Além de que esse horário mais iluminado sofre menos com o problema de cromaticidade (que será abordado em sessões seguintes).

Em Lucia 2011, foi encontrada uma relação entre a idade do formigueiro e a quantidade de olheiros que ele possui, ou a idade do formigueiro com a quantidade de câmaras (LUCIA, 2011). Entretanto, a quantidade de câmaras não é um parâmetro a ser considerado nesse estudo, dado que só se sabe a sua quantidade quando o formigueiro é escavado, e essa prática não será aplicada.

Em relação a quantidade de olheiros, não se sabe ainda se o algoritmo terá capacidade de detectar pequenos montes para fins de se estimar a idade do formigueiro. Então uma idade/tamanho também não foi definido de acordo com a literatura, o que nos leva a concluir que esse parâmetro será obtido por meio de testes experimentais.

Não foi encontrado na literatura um melhor momento para detecção. Entretanto, o que se sabe é que, quanto antes os ninhos forem encontrados e realizado o manejo/controle, menor será o prejuízo econômico.

## 2.4 Estudo do Drone

Drone é o termo popularizado para as aeronaves não-tripuladas, devido o barulho que fazem. De forma ampla, a legislação vigente (ICA 100-40)((ICA100-40, 2020)) define qualquer aeronave não tripulada como :

Qualquer aparelho que possa sustentar-se na atmosfera, a partir de reações do ar que não sejam as reações do ar contra a superfície da terra, e que se pretenda operar sem piloto a bordo.

### 2.4.1 Classificação dos drones

Ainda pela legislação, podemos classificar as aeronaves não tripuladas pelo peso de decolagem: classe C1 - para aeronaves com mais de 150 kg, C2 - para aeronaves entre 150 kg e 25 kg, e C3 para aeronaves com menos de 25 kg.

Cada classe apresenta restrições de registro e acesso ao espaço aéreo. A classe que será focada nessa dissertação de mestrado será a classe C3, respeitando os limites de teto de vôo (400 pés) e mantendo a aeronave em linha de visada. Essa restrição é importante, pois, assim, basta uma autorização no sistema SARPAS do Departamento de Controle de Espaço Aéreo (DECEA) para que o voo possa ser realizado.

Dentro da classe C3, podemos separar as aeronaves não tripuladas pelo método de sustentação. As aeronaves de asa fixa, como o próprio nome diz, possui estruturas de asas fixas, que são responsáveis por gerar a sustentação de toda a aeronave no ar, semelhante aos aviões comerciais. Essas aeronaves podem ter envergaduras de alguns centímetros até alguns metros, podendo ter um ou mais hélices que servem como propulsores. Já as aeronaves de asa rotativa são sustentados pela força de empuxo geradas por uma ou mais hélices, sendo o *design* com um rotor de sustentação semelhante ao helicóptero, e, um exemplo, com 4 rotores para sustentação é o quadrirrotor.

#### 2.4.1.1 Asa fixa

Os drones de asa fixa possuem um design semelhante ao das aeronaves habituais, sendo a sustentação dada pela velocidade relativa nas asas, e a força tratora vindo de 1 ou mais propulsores.

Figura 7 – Drone de asa fixa



Fonte: (BIOTEC, 2022)

Possuir asa fixa não é a única diferença entre esses drones e os de asa rotativa. Drones de asa fixa possuem um método de decolagem diferente. Eles necessitam de uma pista de decolagem para realização do lançamento e pouso manual por meio de um sistema de im-

pulsão (DILEEP et al., 2020) (TSOUROS; BIBI; SARIGIANNIDIS, 2019) (BARSANTE, 2020).

Algumas vantagens do drone de asa fixa são: Alta resistência, alta velocidade, capacidade de maior transporte de carga útil (TSOUROS; BIBI; SARIGIANNIDIS, 2019). Devido a presença das asas fixas que geram sustentação vertical, esses drones possuem maior estabilidade de voo e percorrem maiores distâncias (BARSANTE, 2020).

Já as desvantagens são: Sua limitação devido a necessidade de local amplo e aberto para decolagem e é mais caro que os outros tipos de drone (TSOUROS; BIBI; SARIGIANNIDIS, 2019) (DILEEP et al., 2020) (BARSANTE, 2020).

Esse tipo de drone é utilizado em 22% das aplicações em agricultura de precisão (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

#### 2.4.1.2 Asa rotativa

Os drones de asa rotativa tem a sustentação por um por mais hélices, como exemplos os helicópteros, quadrirrotores, hexarotores, entre outros.

Diferente dos drones de asa fixa, os drones de asa rotativa realizam pouso e decolagem vertical. Esse tipo de decolagem e pouso apresenta uma vantagem por permitir um voo estável em um só lugar, mantendo a manobrabilidade. Outra vantagem dos drones de asa rotativa é o fato deles apresentarem menor custo se comparado com os de asa fixa (TSOUROS; BIBI; SARIGIANNIDIS, 2019). Entretanto, eles não podem voar em velocidades muito altas ou permanecer no ar por muito tempo (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

Os drones de asa rotativa são divididos em outros dois grupos, os de rotor único e os multi-rotores.

##### □ Rotor Único

Apesar da classificação chamar de drone de rotor único, esses drones possuem dois rotores. Um rotor é para sustentação da aeronave (rotor superior), e o outro é para dar direção (rotor de causa) (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

Figura 8 – Drone de asa rotativa rotor único



Fonte: (FARMFOR, 2021)

Drones de rotor único são mais eficientes que os de hélice. De acordo com as regras aerodinâmicas, pás maiores de rotor "precisam" de um giro menor para fornecer o mesmo deslocamento/sustentação que pás menores precisariam (DILEEP et al., 2020).

Outras vantagens do drone de rotor único são: sua possibilidade de voos mais altos e maior autonomia de voo. Entretanto, as desvantagens são: Custo mais elevado para compra, necessidade de manutenção regular e maior instabilidade e complexidade de pilotagem (BARSANTE, 2020).

Esse tipo de drone é utilizado em apenas 4% das aplicações em agricultura de precisão (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

#### ❑ Multi-rotor

Devido a maior quantidade de rotores, esses drones se apresentam mais estáveis que os de rotor único (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

Figura 9 – Drone de asa rotativa multi rotor



Fonte: (STOCK, 2022)

Outras vantagens desses drones são: facilidade de compra em decorrência de serem mais comuns no mercado, possuem boa estabilidade quando parados no ar, possuem melhor controle sobre as posições de onde as fotos são tiradas, isso garante resultados mais precisos (DILEEP et al., 2020) (BARSANTE, 2020).

As desvantagens são: Menor resistência, menor velocidade de voo (DILEEP et al., 2020) e quanto mais rotores o drone possuir, menor será a sua autonomia de voo (sem bateria adicional), dado que quanto mais rotores, mais energia elétrica é necessária para mante-los funcionando (BARSANTE, 2020).

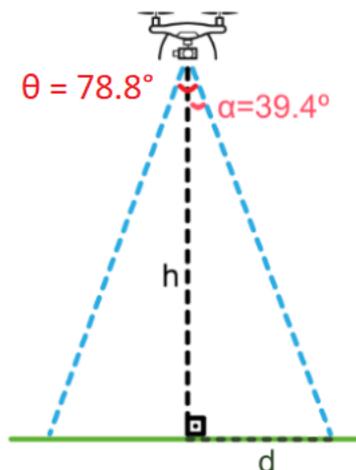
A nomenclatura das subdivisões dos drones multi-rotor é dada de acordo com a quantidade de rotores.

Esse tipo de drone é utilizado em 72% das aplicações em agricultura de precisão (TSOUROS; BIBI; SARIGIANNIDIS, 2019).

## 2.4.2 Visada para baixo

Por meio do modelo de *Pinhole*, é possível utilizar trigonometria para chegar uma estimativa do tamanho, em centímetros, dos formigueiros. Primeiramente é preciso encontrar o valor de  $d$  da Figura 10. Para encontrar o valor de  $d$  em centímetros é necessário considerar que o drone esteja voando paralelamente ao chão, a fim de que sua altura  $h$  forme um ângulo de 90 graus com o solo. Outra consideração necessária é, o ângulo de abertura da lente da câmera foi dado nas especificações técnicas como sendo igual a  $\theta$  (também pode ser encontrado como FOV (*Field Of View*)), com essas informações e uso de trigonometria básica, pode-se calcular o ângulo  $\alpha$  de acordo com a fórmula  $\alpha = \theta/2$ . Fazemos um exemplo, supondo  $\theta = 78.8$  graus, isso resulta em um valor de  $\alpha = 39.4$  graus.

Figura 10 – Visada para baixo



Fonte: Autor.

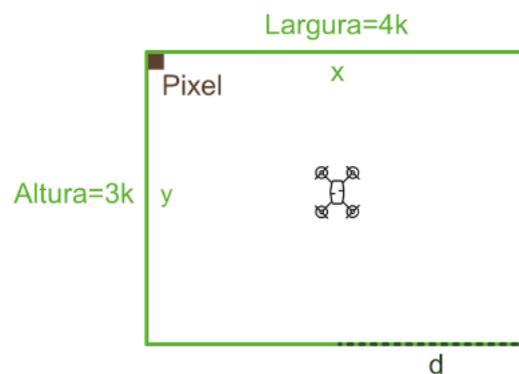
Tendo estabelecido essas considerações, fez-se uso do cálculo da tangente sob o ângulo  $\alpha$ , de acordo com a fórmula abaixo:

$$\tan \alpha = \frac{d}{h} \quad (1)$$

O valor de  $h$  é a altura de voo do drone e pode ser obtido tanto durante o voo, por meio da interface de controle, quanto pelas propriedades da imagem obtida. Continuando o exemplo, foi adotado um valor de  $h$  igual a 3,8m (380cm).

$$\begin{aligned} \tan 39,4 &= \frac{d}{380} \\ 0,82 &= \frac{d}{380} \\ d &= 311,6 \end{aligned}$$

Figura 11 – Dimensões das imagens obtidas pela drone



Fonte: Autor.

Sabe-se que as imagens obtidas possuem dimensões  $x$  de largura, em *pixels*, e  $y$  de altura, em *pixels*. Faz-se, então, uma regra de três, e encontra-se as resoluções de  $x$  ( $R_x$ ) e resoluções de  $y$  ( $R_y$ ) como descrito nas fórmulas abaixo

$$R_x = \frac{d}{x/2} \quad (2)$$

$$R_y = \frac{d}{y/2} \quad (3)$$

Seguindo o exemplo apresentado, se  $x = 4000$  pixels de comprimento e  $y = 3000$  pixels de altura, temos:

$$R_x = \frac{311,6}{4000/2}$$

$$R_x = \frac{311,6}{2000}$$

$$R_x = 0,1558cm$$

$$R_y = \frac{311,6}{3000/2}$$

$$R_y = \frac{311,6}{1500}$$

$$R_y = 0,2077cm$$

Com isso pode-se dizer que as dimensões, em centímetros, de um *pixel* é de aproximadamente 0,16cm de largura por 0,21cm de altura, a uma altura de voo de 380 cm.

Para obter o tamanho do formigueiro em centímetros basta multiplicar a quantidade de pixel que o formigueiro ocupa, pelo valor do pixel em centímetros.

## 2.5 Estudo de Visão Computacional

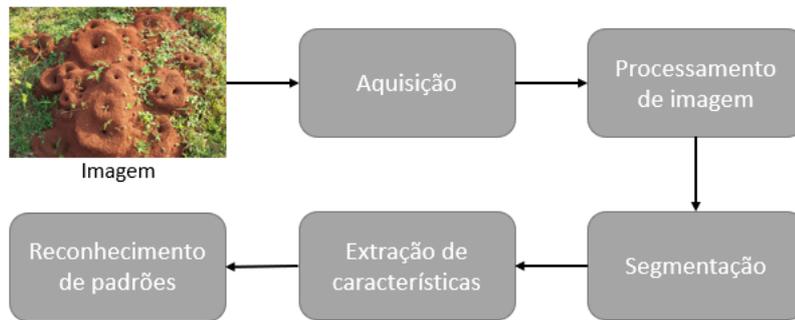
A Visão Computacional pode ser definida como a área de estudo que busca repassar para máquinas a capacidade de visão (BACKES; JUNIOR, 2019).

Segundo os autores Backes e Junior, um sistema de Visão Computacional é constituído por cinco fases, sendo elas:

- ❑ Aquisição: fase responsável por captar imagens. Dispositivos como *scanners*, filmadoras e máquinas fotográficas simulam as funções dos olhos;
- ❑ Processamento: responsável por “melhorar” a imagem, no caso, o saneamento de possíveis problemas existentes, como por exemplo, retirar ruídos, aumentar contraste, salientar bordas e suavizar a imagem. Essa fase de processamento pode ser um fim em si mesma ou ter continuidade ao fornecer uma imagem mais adequada para futuras fases. É uma etapa que abrange “pré-processamento”, como a rotação da imagem, até processamentos complexos como filtragens;
- ❑ Segmentação: fase responsável por particionar a imagem em regiões de interesse;
- ❑ Extração de características/análise de imagens: é a fase responsável em extrair um conjunto de características do objeto de interesse;
- ❑ Reconhecimento de padrões: fase responsável pela classificação ou agrupamento das imagens com base em seus conjuntos de características.

A Figura 12 apresenta um esquema simples de um sistema de Visão Computacional.

Figura 12 – Esquema de um sistema de Visão Computacional



Fonte: Adaptado de BACKES, 2019.

O número de fases pode variar de acordo com o autor, da mesma maneira que algumas fases podem ser anuladas dependendo do problema.

As fases desenvolvidas nesse projeto serão abordadas em mais detalhes as Seções abaixo.

### 2.5.1 Aquisição

A aquisição de imagem é a primeira etapa de sistemas de visão computacional. Sua função é converter os sinais eletrônicos do dispositivo de detecção, como por exemplo imagens de uma câmera, em representações numéricas, que serão utilizadas nas etapas seguintes (ZAREIFOROUGH et al., 2015) (FILHO; NETO, 1999).

É comumente chamado de "aquisição de imagem" o processo de conversão de uma cena real em 3D para uma cena digital em 2D.

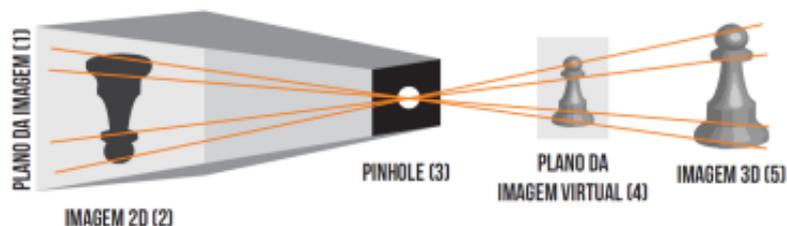
O primeiro passo dessa conversão é a redução de dimensionalidade, sendo os dispositivos mais usados os sensores com carga de estado sólido (CCD ou CMOS) (NARENDRA; HAREESH, 2010). Esses dois sensores de imagem serão tratados na seção 2.5.1.3.

O segundo passo dessa conversão é a digitalização. Na digitalização, o sinal analógico obtido da saída do sensor é submetido a uma discretização espacial e de amplitude, a fim de converter a amostra para o formato desejável para as etapas seguintes (FILHO; NETO, 1999).

#### 2.5.1.1 Modelo óptico das câmeras

No estudo de visão computacional, geralmente as câmeras são modeladas a partir do modelo de câmera *pinhole* (STURM, 2014). O modelo de *Pinhole* (traduzido como buraco de alfinete) é baseado no princípio de câmera escura, onde a luz do objeto entra por um pequeno orifício (3) (DIAS, 2015) (TONINI et al., 2020). A ilustração do modelo de *Pinhole* pode ser observada na Figura 13 abaixo.

Figura 13 – Modelo de Pinhole



Fonte: (DIAS, 2015)

Esse modelo leva em consideração uma projeção central, por meio do centro óptico da câmera (assumindo-se que o centro da câmera é o *pinhole*) e de um plano de imagem (TONINI et al., 2020). Em termos gerais, um objeto 3D (5) tem sua projeção 2D (2) em um plano de imagem (1) (DIAS, 2015).

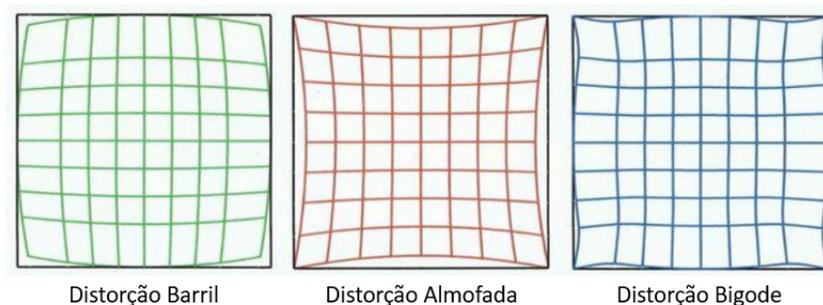
A principal função do orifício é captar a luz para dentro da câmera e projetar a imagem. Esse mecanismo de projeção é semelhante ao olho humano (DIAS, 2015).

Na câmera física, a imagem é formada espelhada e atrás do centro da câmera, entretanto, o plano da imagem é frequentemente representado na frente do centro da câmera (TONINI et al., 2020).

O modelo *pinhole* não considera que as lentes reais podem produzir distorções nas imagens geradas. Distorções são esticamentos não uniformes da imagem, em que o centro e as bordas são ampliados em proporções diferentes (DIAS, 2015). Dentre as distorções existentes, a mais comum é a distorção radial, causada por diferenças de refração da luz em conjunto com o formato esférico da lente (TONINI et al., 2020). Esta distorção ainda pode ser classificada em três tipos, que são: Barril, Almofada e Bigode.

- ❑ A distorção barril é normalmente encontrada em lentes grandes angulares, isso significa, lentes de curto alcance, mas com um grande campo de visão. A característica de identificação desse tipo de distorção é possuir o centro da imagem mais ampliado que as bordas (DIAS, 2015);
- ❑ A distorção almofada é normalmente encontrada em lentes teleobjetivas, isso significa, lentes de longo alcance. A característica de identificação desse tipo de distorção é possuir as bordas da imagem mais ampliadas que o centro (DIAS, 2015);
- ❑ A distorção bigode é uma mistura da distorção barril e almofada. A característica de identificação desse tipo de distorção é a parte central da imagem apresentar características de distorção barril e as extremidades apresentarem características de distorção almofada (DIAS, 2015).

Figura 14 – Tipos de distorção de imagem



Fonte: Autor.

Para corrigir essas possíveis distorções, se faz necessária uma etapa de calibração da câmera antes do uso. A etapa de calibração será mais detalhada na seção ??.

### 2.5.1.2 Parâmetros de fotografia

Os parâmetros de fotografia que serão abordados fazem parte do triângulo de exposição ilustrado na Figura 15. Em fotografia, exposição se refere a quantidade de luz que entra no sensor da câmera.

Figura 15 – Triângulo dos parâmetros de fotografia



Fonte: (MURARI, 2020)

Imagens muito claras, são imagens que estão superexpostas, ou seja, com muita luz. Já imagens muito escuras, são imagens subexpostas, ou seja, com pouca luz. Essa exposição pode ser controlada por meio de três parâmetros, que são: Abertura, Velocidade e ISO.

#### ❑ Abertura

A abertura da lente é um diafragma que encontra-se na própria lente ou atrás dela. Ele é composto por um conjunto de lâminas que abrem ou fecham de acordo com a configuração de abertura selecionada. O abrir e fechar desse diafragma afeta a quantidade de luz que passa pela lente. As configurações de abertura são chamadas

de "*f-stops*" e são expressados por meio de uma sequência numérica específica, como por exemplo, *F5.6*, *F7.1*, *F8*, *F11* e assim por diante. Configurações mais baixas de *f-stops* significam uma abertura maior do diafragma, permitindo a passagem de mais luz através da lente. Já as configurações de *f-stops* mais altas, significam uma menor abertura do diafragma, permitindo uma menor passagem de luz através da lente (WHAT... , 2019b).

O valor da abertura da lente também pode afetar a Profundidade de Campo (DOF). Profundidade de Campo refere-se a quanto a fotografia parece estar em foco. Ao utilizar grandes aberturas, apenas uma parte do objeto ficará em foco e o restante ficará desfocado, ou seja, quanto menor o valor do *f-stops*, mais desfocado o fundo vai ficar. Analogamente, ao utilizar pequenas aberturas toda a foto ficará em foco, ou seja, quanto maior o número do *f-stops*, mais focada a foto ficará como um todo (WHAT... , 2019a)

#### □ Velocidade do obturador

Outra forma de controlar a quantidade de luz que entra na câmera é controlar por quanto tempo o obturador da câmera ficará aberto. O tempo de abertura do obturador normalmente será uma fração que tem como unidade "segundos", entretanto pode-se usar velocidade de obturação de vários segundos. Quanto mais tempo o obturador permanecer aberto, ou seja, a velocidade de obturação mais lenta, mais luz entrará na câmera. Se o obturador da câmera permanecer aberto por menos tempo, ou seja, velocidade de obturação mais rápida, menos luz entrará na câmera (WHAT... , 2019b).

O valor da velocidade também pode afetar o Desfoque de Movimento. O desfoque de movimento refere-se ao desfoque, ou não, dos objetos em movimento no momento da fotografia. Quanto mais alta a velocidade, mais "congelado" o objeto (que estava em movimento) vai ficar. Já quanto menor a velocidade, mais borrado o objeto vai ficar (HOW... , 2019).

Quanto menor a velocidade do obturador mais chances tem da foto ficar inteira desfocada, por isso é recomendada a utilização de tripé, para que só o que realmente está em movimento, mantenha a aparência de movimento na foto (HOW... , 2019).

Vale salientar que para se usar uma velocidade rápida do obturador, deve haver alta incidência de luz. Devido ao obturador ficar aberto por um período mais curto, isso limita a quantidade de luz que entra na câmera. Velocidades lentas do obturador não precisam de tanta luz porque o obturador da câmera fica aberto por mais tempo, permitindo que mais luz entre na câmera por mais tempo (HOW... , 2019).

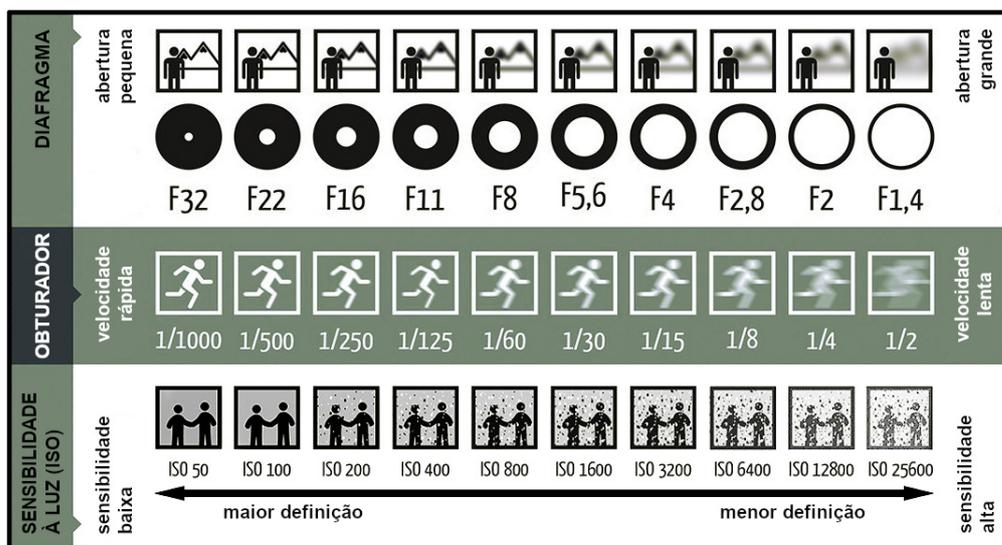
#### □ ISO

O valor de ISO determina a sensibilidade à luz do sensor da câmera. Quanto maior o número ISO, maior é a sensibilidade do sensor, isso significa que menos luz é necessária para produzir a exposição correta. Quanto menor o número ISO, menor é a sensibilidade do sensor, então mais luz é necessária para produzir a exposição correta. Portanto, em ambientes pouco iluminados, recomenda-se o uso de ISO mais alto, e para ambientes bem iluminados, ISO mais baixo (WHAT..., 2019b).

O valor do ISO também pode afetar o grão digital. Um problema que pode ocorrer quando se usa um valor alto de ISO é o surgimento de grãos digitais nas imagens. Quanto mais alto o ISO, a imagem começa a perder qualidade, e essa perda de qualidade é devido ao surgimento de grãos digitais, que é basicamente uma granularidade na imagem (ABERTURA..., 2018).

A Figura 16 resume de maneira ilustrativa o efeito que cada parâmetro pode causar na fotografia.

Figura 16 – Efeitos das alterações dos parâmetros na fotografia



Fonte: (HAMAGUCHI, 2013)

### 2.5.1.3 Sensores de imagem

Sensores são equipamentos que realizam conversões de diversos tipos de sinais, em sinais elétricos. Portanto, sensores de imagens realizam a conversão do sinal de luz (ou fótons) em sinal elétrico. No âmbito de sensores de imagens, são dois os tipos mais comuns: Sensor CCD analógico, que converte fótons em elétrons; e Sensor CMOS digital, que converte a intensidade do fóton em tensão/voltagem (RADHAKRISHNA; GOVINDH; VENI, 2021).

- ❑ O *Charge Coupled Device* (CCD) é um sensor de alta sensibilidade. Ela é composto por diodos individuais, organizados em linhas e colunas. Cada um desses diodos

são chamados de *pixel*. Quando um fóton atinge o *pixel*, os elétrons são gerados no dispositivo individualmente. Essa geração de elétrons é diretamente proporcional a intensidade do *pixel*. A quantidade de elétrons gerados é medida e a imagem é reconstruída (Radha2021);

- ❑ O *Complementary Metal Oxide Semiconductor* (CMOS) também é composto por diodos. Os fótons do objeto atingem o fotodiodo e ele converte em carga, então essa carga é aplicada ao capacitor que a converterá em tensão, e só aí que o conversor analógico/digital converterá as tensões em sinais digitais;

Na Tabela 5 abaixo é possível observar a comparação entre esses dois tipos de sensores.

Tabela 5 – Comparação dos sensores CCD e CMOS

CCD	CMOS
Imagem de alta qualidade	Imagem de baixa qualidade
Alta eficiência quântica	Baixo custo
Alto consumo de energia	Menor consumo de energia
Ruído muito baixo	Ruídos mais altos
Altamente não programável	Baixa capacidade de carga
Alta sensibilidade a luz	Baixa sensibilidade a luz
Captura de imagem restrita em 1/60s	Captura de imagem restrita em 1/30s
Não possui problema de desfoque	Possui Problema de desfoque
Maior velocidade	Menor velocidade

Fonte: Adaptado de MEHTA 2015.

Apesar dos sensores CCD apresentarem mais vantagens, isso não significa que os sensores CMOS sejam inferiores. Os CCDs já estão presentes nas câmeras a muito mais tempo, isso possibilitou uma tecnologia mais avançada. Já os CMOS estão se atualizando e com o avançar da tecnologia eles serão aperfeiçoados e corresponderão aos CCD em termos de resolução e qualidade geral (além das outras vantagens já citadas) (KOZACEK; GRAUZEL; FRIVALDSKY, 2018).

Em suma, os sensores possuem tendências diferentes e a tecnologia está se desenvolvendo rapidamente (MEHTA; PATEL; MEHTA, 2015). Ambos os sensores possuem vantagens de desvantagens, então a escolha do melhor sensor vai de acordo com a aplicação e as necessidades do usuário (RADHAKRISHNA; GOVINDH; VENI, 2021).

#### 2.5.1.4 Métodos de captura de imagem

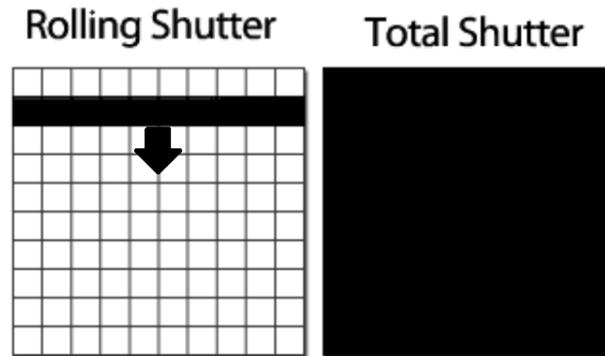
Os métodos de captura de imagens mais comuns são: *Rolling Shutter* e *Global Shutter*.

- ❑ *Rolling Shutter*

*Rolling Shutter* é um método de captura de imagem em que a imagem é capturada fazendo uma varredura dos fotodiodos em linhas ou colunas (*Rolling Shutter* da

Figura 17). Isso significa que todas as partes da cena não são registradas no mesmo instante. Esse tipo de "shutter" produz distorções de objetos em movimento rápido ou flashes de luz rápidos (KOZACEK; GRAUZEL; FRIVALDSKY, 2018);

Figura 17 – Funcionamento *Rolling Shutter*



Fonte: Autor.

A vantagem desse processo é que sensor pode continuar reunindo fótons durante a aquisição, o que resulta em maior sensibilidade;

Ele é mais encontrado em câmeras que possuem sensores CMOS.

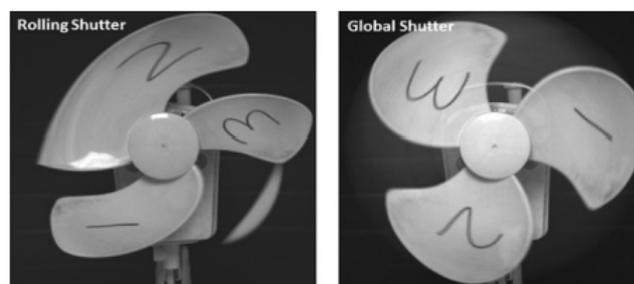
#### □ *Global Shutter*

*Global Shutter* é um método de captura de imagem em que a imagem é capturada fazendo uma varredura de todos os todos os fotodiodos simultaneamente (*Total shutter* da Figura 17). Isso significa que todas as partes da cena são registradas no mesmo instante. Esse tipo de "shutter" não produz distorções de objetos em movimento rápido (KOZACEK; GRAUZEL; FRIVALDSKY, 2018);

Ele é mais encontrado em câmeras que possuem sensores CCD.

Na Figura 18 abaixo é possível observar o efeito que o rolling shutter causa quando a cena tem presença de objeto em movimento rápido.

Figura 18 – *Rolling Shutter X Global Shutter*



Fonte: (KOZACEK; GRAUZEL; FRIVALDSKY, 2018)

As vantagens e desvantagens dos tipos de sensores já foram tratadas na seção anterior. A escolha do sensor/*shutter*, vai de acordo com a necessidade e aplicação.

## 2.5.2 Segmentação

A técnica de segmentação estudada e aplicada foi o *Super Pixel*.

### 2.5.2.1 *Super Pixel*

Por meio do *Super Pixel* o agrupamento dos *pixels* é feito em regiões atômicas que podem ser usadas como substitutas para a grade de *pixels*. Eles capturam redundâncias na imagem gerando uma estrutura que reduz significativamente a complexidade das tarefas de processamento de imagem que serão realizadas em sequência. Esses tipos de algoritmos tem sido amplamente utilizados em aplicações como localização de objetos e segmentação de imagens (FERREIRA et al., 2017).

Em geral, as técnicas de *superpixel* são baseadas em medidas que procuram semelhanças de cores e medidas da forma das regiões. O processo de segmentação incorpora bordas e mudanças abruptas de intensidade para delimitar regiões (NIXON; AGUADO, 2020).

Existem muitas abordagens para gerar *superpixel*, cada uma com suas próprias vantagens e desvantagens dependendo da aplicação. Foi realizado um estudo completo comparando diversas dessas abordagens do *superpixel* e propondo um novo método chamado *Single Linear Iterative Clustering* (SLIC) (ACHANTA et al., 2012) que será abordado na subseção 2.5.2.1.1.

A principal vantagem da utilização do *superpixel* é obter regiões que representem descrições significativas com muito menos dados ao invés de usar todos os *pixels* da imagem fazendo um processamento *pixel a pixel* (NIXON; AGUADO, 2020). A ideia é que ao reduzir o número de "primitivas", a redundância também é reduzida e conseqüentemente reduz a complexidade das tarefas de reconhecimento.

Outra vantagem de se utilizar o *superpixel* é que a substituição da estrutura rígida dos *pixels* delimitando regiões que mantêm significado na imagem, de modo que as regiões forneçam informações sobre a estrutura da cena, facilitando as outras tarefas de processamento (NIXON; AGUADO, 2020).

A quantidade de *superpixel* existentes na imagem, será de acordo com o parâmetro  $k$  inserido no algoritmo (ACHANTA et al., 2012). Sendo assim, não existe uma relação pré estabelecida entre altura de voo do drone e a quantidade de *superpixel* presentes na imagem, sendo este, um parâmetro de estudo.

O algoritmo abordado no presente projeto será o *SLIC* e ele será melhor detalhado na subseção abaixo.

### 2.5.2.1.1 SLIC

*Simple Linear Iterative Clustering* (SLIC) é um dos algoritmos de *superpixel* mais utilizado devido a sua simplicidade e baixo custo computacional (KIM, ).

A estratégia desse algoritmo consiste em agrupar *pixels* com base em semelhança de cores e proximidade espacial na imagem, de tal maneira que os *superpixel* tenham tamanho uniformes e que descrevam adequadamente a estrutura dos objetos na imagem (NIXON; AGUADO, 2020). (FERREIRA et al., 2017). Para isso, é utilizado um espaço de cinco dimensões ( $labxy$ ), em que duas dimensões são para as coordenadas  $x - y$  do *pixel* e três dimensões são para três intensidades de cor do *pixel*, considerando o espaço de cores CIELAB (Lab) (KIM, ).

A vantagem do SLIC com relação aos outros métodos de *superpixel* é que ele é mais rápido, mais eficiente em termos de memória, exhibe aderência aos limites da imagem e melhora o desempenho dos algoritmos de segmentação (ACHANTA et al., 2012).

O agrupamento dos *pixels* para geração do *superpixel* é uma adaptação do algoritmo *k-means* e será explicado melhor abaixo.

O algoritmo constrói *clusters* de *pixel* utilizando uma variação do algoritmo *k-means* que realiza, em um espaço reduzido, a busca proporcional à região do *superpixel*, ao contrário do algoritmo tradicional que realiza a comparação com todos os centros dos *clusters* (FERREIRA et al., 2017).

O único parâmetro que o algoritmo recebe (por padrão) é ' $k$ '. ' $K$ ' é o número desejado de *superpixel*. Dessa forma, para uma imagem em que  $N$  é o número de *pixels*, o tamanho aproximado de cada *superpixel* (para que eles tenham tamanhos aproximadamente iguais) é dado de acordo com a equação  $S = \sqrt{\frac{N}{k}}$  (ACHANTA et al., 2012).

Na etapa chamada de "atribuição", cada *pixel* é associado ao centro do *clusters* mais próximo. Essa é a etapa principal para acelerar o processamento do algoritmo, pois limitar o tamanho da região da pesquisa reduz, de maneira significativa, o número de cálculos de distância e conseqüentemente resulta em uma vantagem de velocidade, se comparado com os agrupamentos de *k-means* convencionais onde cada *pixel* é comparado com todos os centros de *clusters* (ACHANTA et al., 2012).

Após cada *pixel* ter sido associado ao centro do *cluster* mais próximo, na etapa seguinte é realizada uma atualização. Nessa etapa o centro do *cluster* é ajustado para serem o vetor médio  $[labxy]^T$  de todos os *pixels* pertencentes ao *cluster*. As etapas de atribuição e atualização podem ser repetidas iterativamente até o erro convergir (ACHANTA et al., 2012).

E, por último, é realizada uma etapa que reatribui os *pixels* "orfãos" a um *superpixel* próximo. Essa etapa é classificada como um pós processamento que reforça a conectividade (ACHANTA et al., 2012).

Esse agrupamento visa minimizar a soma das distâncias euclidianas entre cada centróide e os pontos a ele atribuídos (KIM, ).

O único parâmetro obrigatório que o algoritmo recebe é um valor para ' $k$ '. Entretanto, é possível ajustar um parâmetro de compactação que permite controlar a forma do *superpixel*, deixando-o mais próximo de losango ou mais amorfo (FERREIRA et al., 2017).

A variável ' $m$ ' é normalmente atribuída para esse fim. A relação entre o valor de ' $m$ ' e a regularidade do formato do *superpixel* é dada por:

- Quanto maior o valor de ' $m$ ', a proximidade espacial é mais importante, o que resulta em *superpixel* mais compactos (com tamanho e formas mais semelhantes);
- Já quanto menor o valor de ' $m$ ', os *superpixel* resultantes aderem melhor aos limites da imagem (menos compacto), porém sem tamanho e formas regulares (ACHANTA et al., 2012).

## 2.5.3 Extração de características

### 2.5.3.1 Espaços de cores

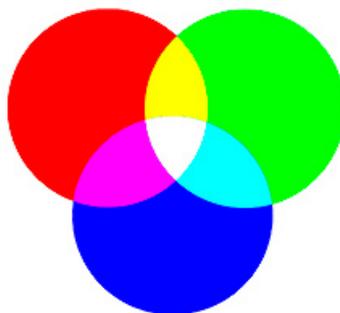
Um espaço de cor pode ser descrito como um método para se expressar a cor de um objeto usando algum tipo de notação ou modelo matemático.

Nas sessões abaixo serão tratados os espaços de cores RGB, HSV e CIELab.

#### 2.5.3.1.1 RGB

RGB é um espaço de cores formado pelas cores primárias: vermelho (*red*), verde (*green*) e azul (*blue*) e são medidos em valores que variam de 0 a 255. A mistura dessas cores formam todas as outras cores por meio de um processo aditivo (Figura 19) (GONZALEZ; WOODS, 2001).

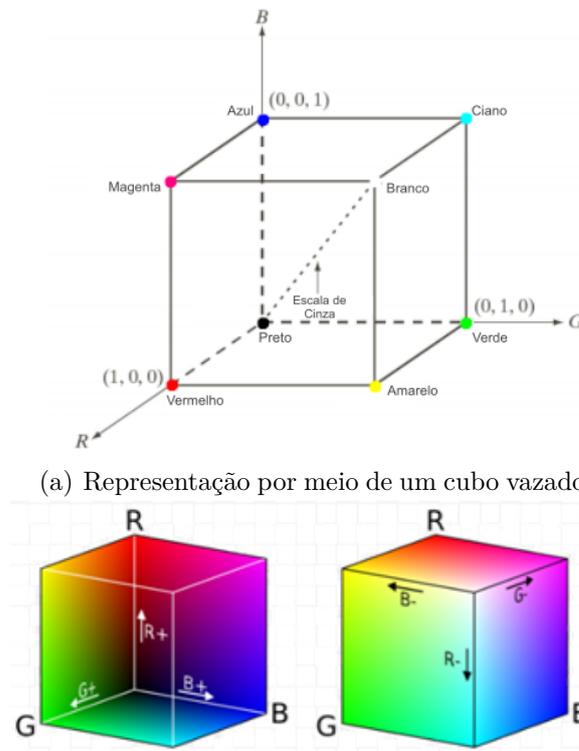
Figura 19 – Processo de adição das cores RGB.



Fonte: (GONZALEZ; WOODS, 2001)

O modelo pode ser representado por meio de um cubo (Figura 20), em que o preto se encontra na origem, o branco no extremo oposto e as cores primárias e secundárias nos outros vértices (GONZALEZ; WOODS, 2001).

Figura 20 – Diagramas do cubo representativo de RGB.



(a) Representação por meio de um cubo vazado

(b) Representação por meio de cubos preenchidos

Fonte: Adaptado de WOODS, 2001.

A principal vantagem que torna o espaço RGB o mais utilizado é a sua simplicidade (MAHESWARI; KORAH, 2016). Entretanto, estudos provaram que converter valores RGB em tons de cinza não resultou em uma boa segmentação porque os pixels de plantas e de solo possuem valores de tons de cinza semelhantes. Portanto, para uma melhor segmentação o espaço RGB é frequentemente convertido em espaços de cores alternativos (TIAN; SLAUGHTER, 1998).

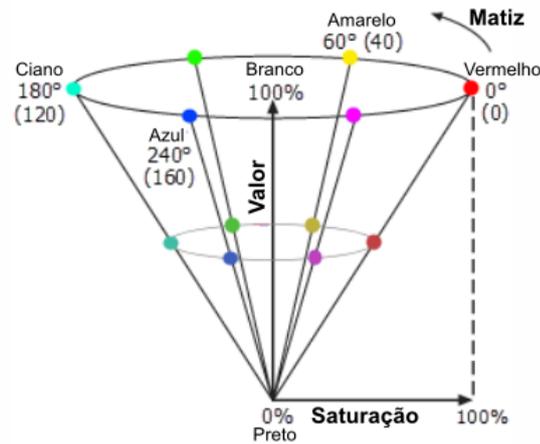
### 2.5.3.1.2 Conversão RGB/HSV

HSV é um espaço de cores formado por três variáveis, que são elas: Matiz (*Hue*), Saturação (*Saturation*) e Brilho/Valor (*Value*). Matiz descreve o pigmento de uma cor e é medido em graus (de 0 a 359), Saturação descreve o grau de pureza ou intensidade da cor e é medida em porcentagem, quanto mais pura a cor (mais próximo do 100), menos partículas de cinza ela possui e maior é sua intensidade. Já o brilho, descreve a quantidade de branco que uma cor contém e é medido em porcentagem (quanto mais brilho, mais branco possui e mais próximo de 100 é a porcentagem) (GANESAN et al., 2019).

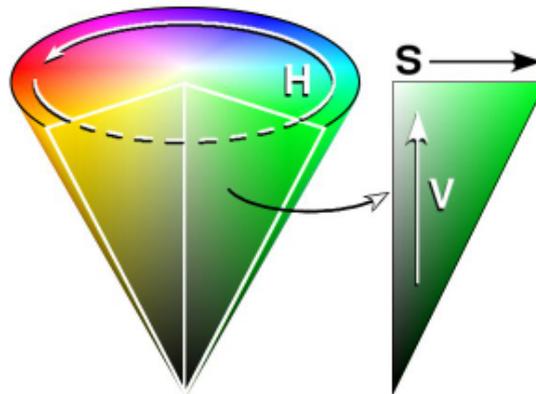
O modelo pode ser representado por meio de um cone, em que o raio é o valor da saturação (em porcentagem), a altura o valor do brilho (em porcentagem) e no ângulo o valor da cor (em grau). A explicação em coordenadas cilíndricas foi adotada para

facilitar o entendimento, entretanto o espaço de cor HSV é representado por um cone de coordenadas cartesianas e polares (Figura 21). A representação cilíndrica é para o espaço HSL, que não será abordado nesse estudo (GANESAN et al., 2019).

Figura 21 – Diagramas do cone representativo de HSV.



(a) Representação por meio de um cone vazio



(b) Representação por meio de um cone preenchido

Fonte: Adaptado de WOODS, 2001.

As fórmulas matemáticas da conversão de RGB para HSV podem ser encontradas nas equações abaixo:

$$H = \begin{cases} 60 \left( \frac{G-B}{Max-Min} \right) + 0, & \text{se } Max=R \text{ e } G \geq B \\ 60 \left( \frac{G-B}{Max-Min} \right) + 360, & \text{se } Max = R \text{ e } G < B \\ 60 \left( \frac{B-R}{Max-Min} \right) + 120, & \text{se } Max = G \\ 60 \left( \frac{R-G}{Max-Min} \right) + 240, & \text{se } Max = B \end{cases} \quad (4)$$

$$S = \begin{cases} \left( \frac{Max-Min}{Max} \right), & \text{se } Max > 0 \\ 0, & \text{se } Max=0 \end{cases} \quad (5)$$

$$V = Max \quad (6)$$

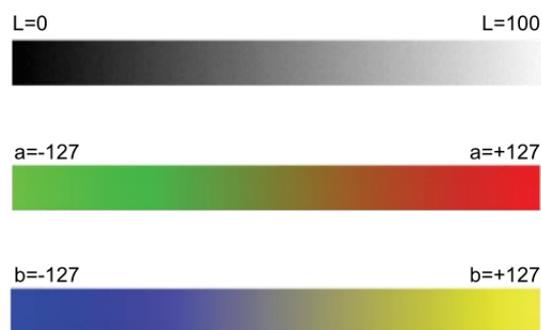
Dado que o espaço de cor HSV é composto por três variáveis distintas (que não são apenas três valores de cor, como no espaço RGB), uma das vantagens do HSV é a possibilidade de separar a cor da sua intensidade e brilho. Há outros espaços de cores que também permitem essa separação (da cor e intensidade/brilho), entretanto para o HSV existem muitos códigos disponíveis que realizam a conversão (RGB/HSV) de maneira simples de ser implementada. E por fim, o espaço HSV ainda é o que mais se assemelha a percepção humana das cores (GANESAN et al., 2019).

### 2.5.3.1.3 Conversão RGB/CIELAB(L\*a\*b)

CIELAB é um espaço de cores formado por três coordenadas, que são eles: Luminosidade (L), Alpha (A) e Beta (B). A coordenada  $L$  representa a luminosidade, já as coordenadas  $a$  e  $b$  são canais de faixas de cores, e são eles: O canal  $a$  que refere-se a faixa de cores de vermelho e verde e o canal  $b$  que refere-se a faixa de cores de azul e amarelo (GANESAN; RAJINI; RAJKUMAR, 2010).

A luminosidade (L) é medida em uma escala de de 0 a 100, em que 0 é o preto e 100 é o branco. A coordenada ' $a$ ' é medida entre -127 e +127, em que -127 representa o verde puro e o +127 representa o vermelho puro. A coordenada ' $b$ ' é também medida entre -127 e +127, em que o -127 representa o azul puro e o +127 representa o amarelo puro (GANESAN; RAJINI; RAJKUMAR, 2010). Essas faixas de valores podem ser observados na Figura 22 abaixo:

Figura 22 – Faixa de valores de cada variável CIELAB

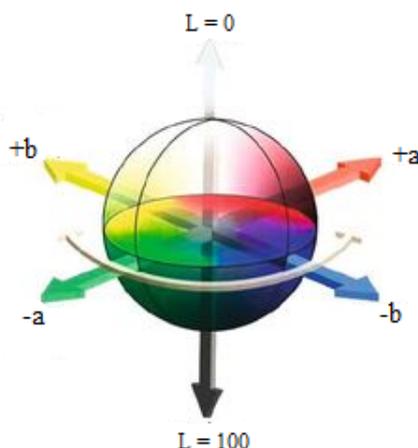


Fonte: Adaptado de GANESAN, 2010.

Ele é baseado no conceito de que as cores podem ser consideradas combinações entre vermelho e amarelo, vermelho e azul, verde e amarelo e verde e azul (GANESAN; RAJINI; RAJKUMAR, 2010).

O modelo pode ser representado por meio de uma esfera (Figura 23), em que os eixos  $X$  e  $Y$  são os valores das cores e o eixo  $Z$  o valor da luminosidade.

Figura 23 – Diagrama da esfera representativa de CIELAB



Fonte: Adaptado de GONZALEZ, 2011.

A fórmula matemática da conversão de RGB para CIELAB pode ser encontrada no *OpenCV* (OPENCV, 2021).

Segundo a Comissão Internacional de Iluminação, o CIELAB é o espaço de cores mais completo. Ele consegue descrever todas as cores do olho humano e foi criado para ser um modelo referência independente do dispositivo (GANESAN; RAJINI; RAJKUMAR, 2010). Isso quer dizer que suas cores não variam de um dispositivo para outro.

Outra vantagem do CIELAB é que a componente  $L$  pode ser usada para fazer correções de equilíbrio de cores, modificando as curvas de saída nos componentes ' $a$ ' e ' $b$ ' (GANESAN; RAJINI; RAJKUMAR, 2010).

### 2.5.3.2 Índices de cores/vegetativo

Cada *pixel* pode ser modificado por uma função matemática, assim, a cor de uma região de interesse pode ser acentuada, enquanto a região indesejada será atenuada. Algumas funções descrevem combinações entre os canais de cores e, portanto, são chamados de índices de cores.

Portanto, índices de cores são modelos matemáticos, ou algoritmos, que buscam atenuar ou destacar alguma característica desejada. Trabalhar com índices de cores significa a possibilidade de se realizar transformações matemáticas que realçam alguns espectros de cores em comparação a outros. Quando se fala em espectros de cores, nesse caso, os mais comuns de serem utilizados são as bandas espectrais do espaço de cor RGB. Isso significa que as equações matemáticas desenvolvidas trabalharam com as cores vermelho, verde e azul de maneiras distintas, realizando o realce ou suavização das mesmas (HAMUDA; GLAVIN; JONES, 2016).

Uma vantagem de se trabalhar com índices de cores é a redução de dimensionalidade causada por ele. O que antes existia em três dimensões distintas (R (vermelho), G (verde),

B (azul)) passarão a ser apenas uma dimensão (Índice de cor), que engloba e pondera de maneiras diferentes o RGB (que antes era trabalhado cada cor com uma dimensão).

### 2.5.3.2.1 ExG

O *Excess green* (ExG) é um método que apresenta bons resultados quando aplicado para separação de plantas do solo descoberto (HAMUDA; GLAVIN; JONES, 2016). Isso ocorre porque o ExG fornece um contraste claro entre as plantas e o solo, e isso fornece uma imagem de intensidade quase binária (MEYER; NETO, 2008). O índice ExG tem sido amplamente utilizado e tem apresentado um bom desempenho na separação de plantas e não plantas (MEYER; HINDMAN; LAKSMI, 1998) (HAMUDA; GLAVIN; JONES, 2016) (MEYER; NETO, 2008).

O balanço do índice de cor proposto por Woebbecke (WOEBBECKE et al., 1995) é de:  $2g - r - b$ . Em que  $g$ ,  $r$  e  $b$  são as coordenadas cromáticas:

$$\begin{aligned} r &= \frac{R'}{(R' + G' + B')} \\ g &= \frac{G'}{(R' + G' + B')} \\ b &= \frac{B'}{(R' + G' + B')} \end{aligned} \quad (7)$$

Em que  $R'$ ,  $G'$  e  $B'$  são os valores RGB normalizados entre 0 e 1, e são calculados de acordo com a equação abaixo:

$$\begin{aligned} R' &= \frac{R}{R_{\max}} \\ G' &= \frac{G}{G_{\max}} \\ B' &= \frac{B}{B_{\max}} \end{aligned} \quad (8)$$

Em que  $R$ ,  $G$  e  $B$  são os valores de *pixel* das imagens baseadas em cada canal e  $R_{\max}$ ,  $G_{\max}$  e  $B_{\max}$  vale 255 cada um.

### 2.5.3.2.2 ExR

O método *Excess Red* (ExR) foi inspirado no fato de que há 4% de azul e 32% de verde, em comparação com 64% de cones vermelhos na retina do olho humano. Com isso iniciou-se um estudo que comparou o método ExR e ExG em um experimento que tinha como finalidade segmentar regiões foliares do restante da imagem (fundo) (MEYER; HINDMAN; LAKSMI, 1998). O índice vermelho foi capaz de segmentar os *pixels* de planta dos *pixels* do fundo, entretanto não foi tão preciso quando o ExG. A fórmula desenvolvida e utilizada nos experimentos para o ExR é definida como:  $ExR = 1.3R - G$

### 2.5.3.2.3 NDI

No *Normalised Difference Index* (NDI) foi testado três métodos para distinguir o material vegetal do fundo do solo em uma imagem RGB (WOEBBECKE et al., 1993). Foi avaliada uma gama de índices de diferença com base nos canais  $R$ ,  $G$  e  $B$ , como por exemplo,  $G - R$ ,  $G - B$  e  $G - R/G + R$ , com o terceiro demonstrando a melhor separação da planta do fundo (WOEBBECKE et al., 1993). Este índice é aplicado a todos os *pixels* da imagem, fornecendo valores que variam entre 1 e +1, mas para exibir a imagem, esses valores devem variar entre 0 e 255. Portanto, o índice foi processado adicionando 1 a ele e depois multiplicado por um fator de 128 para fornecer um cinza (WOEBBECKE et al., 1993).

$$NDI = 128 \left( \left( \frac{G - R}{G + R} \right) + 1 \right) \quad (9)$$

### 2.5.3.2.4 CIVE

O *Colour INdex of Vegetation Extraction* (CIVE) foi proposto para separar as plantas verdes do fundo do solo, a fim de avaliar o estado de crescimento da cultura de soja e beterraba (KATAOKA et al., 2003). Seu cálculo é dado por:

$$CIVE = 0.441R - 0.881G + 0.385B + 18.78745 \quad (10)$$

### 2.5.3.2.5 VEG

O *Vegetative Index* (VEG) foi proposto para separar as *pixels* das plantas (cereais e ervas daninhas) dos *pixels* do solo (HAGUE; TILLET; WHEELER, 2006). Para alcançar a segmentação, uma imagem RGB foi convertida em tons de cinza usando a seguinte fórmula:

$$VEG = \frac{G}{R^a B^{(1-a)}} \quad (11)$$

onde  $a$  é um valor constante igual a 0,667. Haia descobriu que essa transformação demonstrou bom contraste entre planta e solo. Além disso, o VEG tem uma vantagem significativa porque é robusto à mudança de iluminação (HAGUE; TILLET; WHEELER, 2006).

### 2.5.3.2.6 Outros

Na literatura, é possível encontrar diversos índices de cores para as mais variadas aplicações. Entretanto, a maioria desses índices estão relacionados a segmentação de

plantas (HAMUDA; GLAVIN; JONES, 2016). Dentre esses diversos índices alguns outros chamaram atenção, foram eles:

- MExG - *Modified Excess Green Index*: Nele é utilizado um índice já conhecido, porém os pesos dos canais sofreram alterações. Seu cálculo é dado por:  $MExG = 1.262G - 0.884R - 0.311B$  (BURGOS-ARTIZZU et al., 2011);
- *Combined Indices 2*: Nele foi realizada uma soma ou média ponderada entre outros índices. Seu cálculo é dado por:  $COM2 = -0.36ExG + 0.47CIVE + 0.17VEG$  (GUERRERO et al., 2012)

Apesar desses índices serem utilizados com a finalidade de fazer a segmentação de plantas, foram as suas propostas que chamaram atenção. Eles realizaram uma leitura dos dados e propuseram novos índices, cada um adaptado da sua maneira.

Alunos do grupo de robótica *Mobile Robotics Laboratory* da Universidade Federal de São Carlos (UFSCar), propuseram outros dois índices vegetativos que foram utilizados no presente projeto, sendo eles  $mA$  e  $BmA$ . Ambos os índices trabalham e realizam operações matemáticas em cima do espaço de cor  $LAB$ , anteriormente citado nesse texto.

- $mA$ : O índice vegetativo  $mA$  é o negativo do canal de cor  $A$ ;
- $BmA$ : Já o índice vegetativo  $BmA$  é uma operação de subtração do canal  $B$  sob o canal  $A$ .

### 2.5.3.3 Textura

A textura de um objeto é uma característica visual importante durante o procedimento de classificá-lo e reconhecê-lo com algoritmos de visão computacional. Analisando texturas é possível distinguir regiões de uma imagem que apresentam padrões semelhantes, e, portanto, podem ser classificados como parte de um mesmo grupo.

Apesar de ser um termo amplamente utilizado na literatura e em diversas aplicações, o termo "textura" não possui uma definição única e precisa e nem uma aproximação matemática definida. Cada pesquisador define a textura de acordo com o interesse de suas áreas de pesquisa. Logo, diversas definições podem ser encontradas e elas variam de acordo com a aplicação desejada, não havendo consenso sobre uma única definição (TRAVAINI, 2015).

O estudo de texturas possui diversas abordagens, algumas delas são: Abordagens estatísticas, estrutural, espectral e descritores locais de textura.

- A abordagem estatística descreve a textura por meio de regras estatísticas que gerenciam a distribuição e relação entre os níveis de cinza e a interação, ou falta dela, é medida por meio de coeficientes probabilísticos (HARALICK, 1979);

- A abordagem estrutural procura buscar padrões bidimensionais que podem ser construídos utilizando-se um conjunto de regras. Para o funcionamento dessa abordagem, o ideal é a extração de padrões da textura grandes o suficiente para que possam ser segmentados individualmente e depois descritos. O método de descrição é realizado com estatísticas das propriedades extraídas (JULESZ, 1962) (HARALICK, 1979);
- A abordagem espectral trabalha com o espectro das frequências em imagens de textura. esses padrões de textura são facilmente distinguíveis com concentrações de alta energia no espectro (GONZALEZ; WOODS, 2008);
- Os descritores locais de textura são classificados como uma abordagem por micropadrões locais. Eles realizam a análise de propriedades da superfície do objeto analisando a vizinhança e codificando cada uma delas, gerando desse modo as relações das intensidades relativas entre os pixels em uma vizinhança e não sobre os valores de intensidade absolutos (THEODORIDIS et al., 2010).

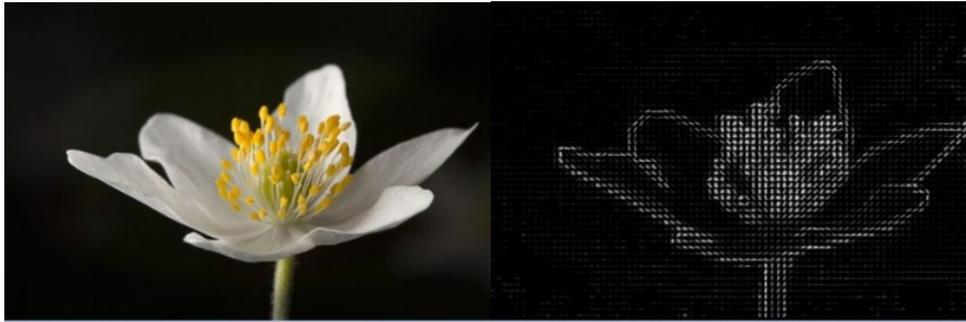
### 2.5.3.3.1 HOG

Todos os objetos em imagens possuem forma, cor e textura. Estas propriedades podem ser medidas e essas medidas são denominadas características da imagem. Tais características são normalmente agrupadas em um vetor de escalares, denominado descritor de imagem (SANTOS et al., 2012).

O Histograma de Gradientes Orientados (HOG) é baseado em descritores de recursos. Um descritor de recurso extrai as informações úteis da imagem e descartam as partes consideradas desnecessárias. Devido a complexidade computacional do cálculo, a imagem é dividida em pequenas regiões chamadas células, e para cada *pixel* dentro de cada célula, um histograma de direções de gradiente é compilado (DALAL; TRIGGS, 2005). O cálculo de cada histograma é feito por meio da componente horizontal e vertical da magnitude do gradiente e direção (SAID; ATRI; TOURKI, 2011). O descritor final é um vetor de dados, que é basicamente a concatenação desses histogramas gerados em cada *pixel* e depois em cada célula. (DALAL; TRIGGS, 2005).

No descritor HOG, a distribuição (histogramas) de direções de gradientes (gradientes orientados) são usados como recursos. Gradientes (derivadas de  $x$  e  $y$ ) de uma imagem são úteis porque a magnitude dos gradientes é grande em torno das bordas e cantos (regiões de mudanças abruptas de intensidade) e sabe-se que bordas e cantos contêm muito mais informações sobre a forma do objeto do que regiões planas (HISTOGRAM. . . , 2016).

Figura 24 – Descritor de textura HOG



(a) Imagem real RGB

(b) Imagem descrita por HOG

Fonte: (RATH, 2020)

### 2.5.3.3.2 LBP

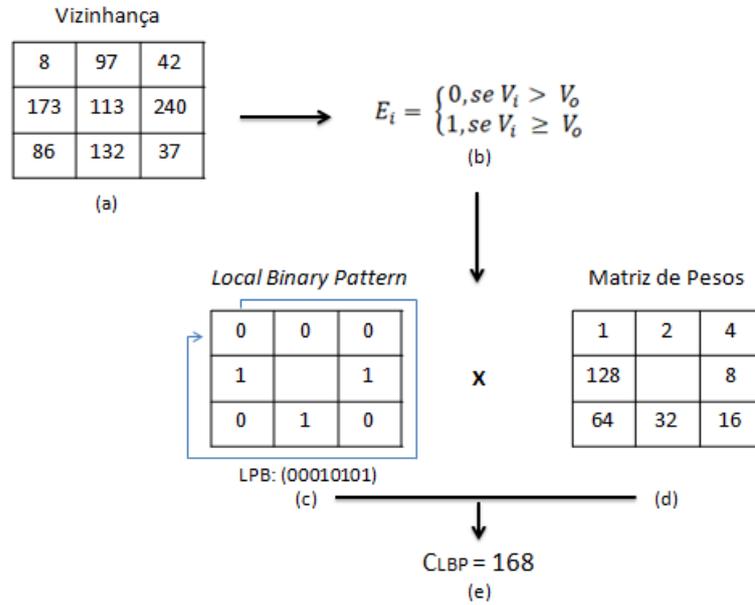
O *Local Binary Pattern* (LBP) é classificado como um descritor local de textura, isso significa que ele analisa a imagem em relação as intensidades relativas entre os *pixels* em uma pequena vizinhança (OJALA; PIETIKAINEN; HARWOOD:, 1996). Como se trata de relação entre intensidades, as análises são realizadas com as imagens em escala de cinza, sendo assim, converter a imagem colorida em escala de cinza é o primeiro passo para a utilização do descritor LBP. Essa comparação ocorre com o intuito de adquirir um código que caracterize a micro região selecionada.

Para a análise é considerada uma vizinhança de  $3 \times 3$  *pixels* ao redor de um *pixel* central que pode ser observado na 26a. Essa vizinhança é limitada pelo valor do *pixel* central, de acordo com a 26b, gerando uma matriz de valores binários encontrada na 26c. Cada *pixel* vizinho ( $V_i$ ) adquire o valor ( $E_i$ ) respeitando a seguinte condição:

- Se  $V_i$  for menor que o *pixel* central ( $V_o$ ), então é atribuído a ele o valor '0'
- Se  $V_i$  for maior ou igual ao *pixel* central ( $V_o$ ), então é atribuído a ele o valor '1'

Os valores dos *pixels* na vizinhança limitada são multiplicados por uma matriz peso (26d). Para obtenção do Código LBP ( $C\_LBP$ ) (26e) é necessária a soma dos valores conforme a equação abaixo.

Figura 25 – Funcionamento de um LBP



Fonte: Autor.

$$C_{LBP} = \sum_{i=1}^8 E_i * 2^{i-1} \tag{12}$$

Figura 26 – Descritor de textura LBP



(a) Imagem real RGB (b) Imagem real em escala de cinza (c) Imagem descrita por LBP

Fonte: Autor.

O descritor é o histograma gerado a partir das ocorrências dos Códigos LBP , formando um vetor de características. (RIBEIRO; PAIVA; JORGE, 2014)

### 2.5.3.3.3 FFT

A Transformada de Fourier é uma ferramenta matemática que realiza a transformação das variáveis no domínio do tempo, para o domínio da frequência. Toda ferramenta

matemática demanda um certo "esforço" para ser utilizada, quando ela é realizada por uma máquina, esse esforço é chamado de complexidade computacional. A complexidade computacional é a quantidade de operações aritméticas necessária para realizar um determinado cálculo e ela está diretamente relacionado com o tempo de processamento para a realização desses cálculos.

Em 1965, Cooley e Tukey (COOLEY; TUKEY, 1965) propuseram um algoritmo que produzia os mesmo resultados que a Transformada Discreta de Fourier (DFT), porém reduzia drasticamente a complexidade computacional de seu processamento. Esse novo algoritmo foi chamado de Transformada Rápida de Fourier (FFT). Enquanto a DFT apresenta uma complexidade computacional de  $N^2$ , a FFT apresenta complexidade de  $N \cdot \log_2 \cdot N$  (em que  $N$  é a quantidade de dados) (COOLEY; PETER; WELCH, 1967).

Tanto para o cálculo da DFT quanto da FFT a fórmula matemática utilizada é a mesma, a diferença está na sua decomposição e no uso de elementos trigonométricos que a torna menos complexa (COOLEY; PETER; WELCH, 1967).

Vale lembrar que a FFT é comumente utilizada para casos de aplicações 1D e quando trata-se de imagens estamos falando de algo em 3D. Para utilização da FFT em imagens a mesma deve ser convertida para escala de cinza, tornando-a bidimensional (2D) e ai então é aplicada a FFT de acordo com a equação abaixo.

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi(\frac{ux}{M} + \frac{vy}{N})] \quad (13)$$

para

$$u = 0, \dots, M - 1$$

e

$$v = 0, \dots, N - 1$$

Normalmente para aplicações em imagens, a FFT é utilizada combinada com funções Gaussinas e Filtros.

#### 2.5.3.3.4 Detecção de borda

Borda é o contorno entre um objeto e o fundo, indicando o limite entre objetos sobrepostos. Contorno é uma linha fechada formada pelas bordas de um objeto. Bordas são definidas como picos da magnitude do gradiente, ou seja, são variações abruptas que ocorrem ao longo de curvas baseadas nos valores do gradiente da imagem. Desde que uma borda é definida por uma mudança no nível de cinza, quando ocorre uma descontinuidade na intensidade, ou quando o gradiente da imagem tem uma variação abrupta, um operador que é sensível a estas mudanças operará como um detector de bordas (ALGORITMOS... , 1998).

□ Operador Roberts

- Fácil de implementar (Matriz da máscara 2x2);
  - Rápido para calcular a máscara de resposta;
  - Sensível aos ruídos da imagem.
- Operador de Sobel
- Menos sensível ao ruído do que o Roberts (devido a matriz da máscara ser 3x3);
  - Resultados mais precisos.
- Operador Robinson
- Utiliza um conjunto de oito máscaras;
  - Aumenta a precisão;
  - Requer mais esforço computacional devido ao tamanho das máscaras.
- Detector de bordas de Canny
- É um filtro de convolução (convolução de Gauss) que uniformiza o ruído e localiza as bordas;
  - A convolução é relativamente simples de ser implementada, porém cara computacionalmente (especialmente em 2D);
  - Algoritmo é mais eficiente no processamento de imagens com ruídos ou com bordas difusas.
- Operador de Marr-Hildreth
- Utiliza derivadas;
  - Uniformização com uma máscara de Gauss e depois calcula a derivada segunda;
  - Bordas são encontradas em todas as direções (devido a utilização da segunda derivada);
  - Pode marcar bordas em lugares que não são bordas.

## 2.6 Estudo Aprendizado de Máquina

O Dicionário de Cambridge define como *Machine Learning* (ML) "O processo em que os computadores mudam a maneira como realizam as tarefas, aprendendo com novos dados, sem que um ser humano precise dar instruções na forma de um programa".

Já o Dicionário *Collins* define *Machine Learning* como: "Um ramo da inteligência artificial em que um computador gera regras subjacentes ou com base em dados brutos que foram inseridos nele".

Em outras palavras, Aprendizado de Máquina (AM) é uma área de Inteligência Artificial que trabalha com automatização de modelos analíticos, a qual permite a criação de modelos que analisam conjuntos de dados e aprendem a reconhecer padrões/realizar previsões; AM é baseada na aplicação de algoritmos que dão aos modelos a capacidade de identificação e/ou previsão, a depender do tipo de aprendizado envolvido na aplicação.

Autores como O'Shea (O'SHEA; NASH, 2015), classificam o aprendizado de máquina em duas categorias distintas, são elas: aprendizagem supervisionadas e não supervisionadas.

### 2.6.1 Aprendizagem supervisionada

É realizada por meio da entradas de dados pré-rotulados. O objetivo desse tipo de aprendizado é, geralmente, prever ocorrências futuras. Além disso, é subdividido entre as técnicas de classificação e regressão as quais possuem características diferentes que devem ser observadas na estruturação do seu modelo de previsão.

- Classificação: é o método de previsão de uma classe discreta ou categorias.
- Regressão: é o método de previsão de um valor contínuo;

São exemplos de aprendizagem supervisionada: Árvores de Decisão e Redes Neurais Artificiais.

### 2.6.2 Aprendizagem não-supervisionada

O conjunto de dados fornecido não possui nenhum rótulo. Esse tipo de aprendizado trabalha com a busca e reconhecimento de padrões e relacionamentos. De um modo geral, ele é utilizado para encontrar peculiaridades implícitas nos dados. São exemplos de aprendizagem não supervisionada: Algoritmos de clusterização e Regras de Associação.

Entretanto, existe autores que, além das duas classificações acima, acrescentam uma terceira classificação chamada de Aprendizagem por reforço, que não será abordada nessa revisão bibliográfica.

### 2.6.3 Descritores Estatísticos

Descritores estatísticos fazem parte da chamada estatística descritiva. A estatística descritiva é uma das áreas da estatística que aplica técnicas para descrever e sumarizar um conjunto de dados. Algumas das medidas que são normalmente usadas para descrever um conjunto de dados são medidas de tendência central e medidas de variabilidade (MORETTIN; BUSSAB, 2013).

As medidas de tendências centrais que serão descritas abaixo são moda, mediana e média. Já as medidas de variabilidade são desvio padrão, assimetria e curtose.

### 2.6.3.1 Moda

A moda é definida como a realização mais frequente do conjunto de valores observados (MORETTIN; BUSSAB, 2013);

### 2.6.3.2 Mediana

A mediana é a realização que ocupa a posição central da série de observações, quando estão ordenadas em ordem crescente (MORETTIN; BUSSAB, 2013);

### 2.6.3.3 Média Aritmética

A média aritmética, é a soma das observações dividida pela quantidade delas (MORETTIN; BUSSAB, 2013). A média aritmética é calculada por:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{i=1}^N x_i \quad (14)$$

Seja  $n$  o número total de valores e  $x_i$  cada valor, em que  $i = 1, \dots, n$ .

### 2.6.3.4 Média Ponderada

A média ponderada, é a média aritmética com um coeficiente  $w_i$  ponderando os valores. A média ponderada é calculada por:

$$\frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} = \frac{w_1 x_1 + w_2 x_2 + \dots + w_n x_n}{w_1 + w_2 + \dots + w_n} \quad (15)$$

Podendo o valor de  $w_i$  ser sempre igual ou diferente.

### 2.6.3.5 Desvio Padrão

Desvio padrão indica uma medida de dispersão dos dados em torno da média amostral (média aritmética), ou seja, indica o quanto o conjunto de dados é uniforme (GOUVEIA, 2021).

O desvio padrão é calculo por:

$$DP = \sqrt{\frac{\sum_{i=1}^N (x_i - M_A)^2}{n}} \quad (16)$$

Em que  $x_i$  é o valor na posição  $i$  do conjunto de dados,  $M_A$  é a media aritmética dos dados e  $n$  é quantidade de dados.

- Baixo desvio padrão (valor mais próximo de zero) indica que os pontos dos dados tendem a estar próximos da média (mais homogêneo) (GOUVEIA, 2021);
- Alto desvio padrão indica que os pontos dos dados tendem a estar espalhados por uma ampla gama de valores (GOUVEIA, 2021).

### 2.6.3.6 Skewness (Assimetria)

É a medida de quanto a sua curva de frequência se afasta da posição simétrica ou a medida da falta de simetria em uma determinada distribuição de frequência (CUNHA, 2017). Uma distribuição simétrica é quando apresenta os mesmos valores para a moda, média e mediana (CUNHA, 2017) (OLIVEIRA., 2021).

*Skewness* pode ser calcula pelo Coeficiente de Pearson:

$$A_S = \frac{\bar{x} - M_O}{S} \quad (17)$$

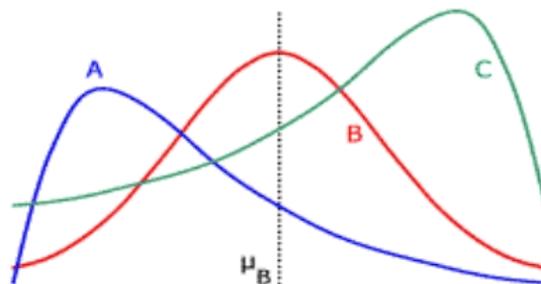
Em que  $\bar{x}$  é a média,  $M_O$  é a moda e  $s$  é o desvio padrão.

O valor final do coeficiente de Pearson ( $A_S$ ) indica a presença de simetria ou não, e onde encontra-se a assimetria

- $A_S = 0$  - Distribuição Simétrica;
- $A_S > 0$  - Distribuição Assimétrica Positiva (à direita);
- $A_S < 0$  - Distribuição Assimétrica Negativa (à esquerda).

Na Figura 27 pode ser observada em *A* uma distribuição assimétrica positiva, em *B* uma distribuição simétrica e em *C* uma distribuição assimétrica negativa

Figura 27 – Média de *Skewness*



Fonte: (KITAHARA, 2021).

### 2.6.3.7 Kurtosis (Curtose)

*Kurtosis* (Curtose) ou achatamento, é a medida da concentração (ou dispersão) dos valores dos dados em relação às medidas de tendência central de uma distribuição normal (CUNHA, 2017) (OLIVEIRA., 2021).

Curtose pode ser calculada pelo Coeficiente de Curtose:

$$K = \frac{Q_3 - Q_1}{2(P_{90} - P_{10})} \quad (18)$$

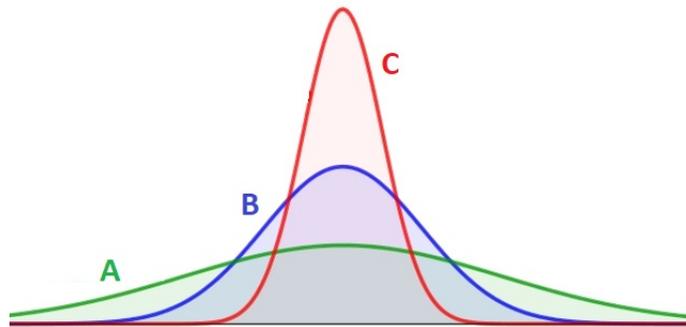
Em que  $Q_1$  é o primeiro quartil,  $Q_3$  o terceiro quartil,  $P_{90}$  percentil 90, e  $P_{10}$  percentil 10.

O valor final do coeficiente de curtose ( $K$ ) indica o tipo de distribuição dos dados.

- $K = 0.263$  - Distribuição Mesocúrtica;
- $K > 0.263$  - Distribuição Platocúrtica;
- $K < 0.263$  - Distribuição Leptocúrtica.

Na Figura 28 pode ser observada em *A* uma distribuição Platocúrtica, em *B* uma distribuição Mesocúrtica e em *C* uma distribuição Leptocúrtica.

Figura 28 – Média de *Kurtosis*



Fonte: Autor.

#### 2.6.4 Base de dados

Um banco de dados, ou base de dados, é uma coleção de dados relacionados. Os dados são fatos que podem ser gravados e que possuem um significado implícito. Apesar da definição de banco de dados ser muito genérica, seu uso é geralmente mais restrito (ELMASRI; NAVATHE, 2015).

As bases de dados possuem as seguintes propriedades implícitas (ELMASRI; NAVATHE, 2015):

- Um banco de dados representa alguns aspectos do mundo real, sendo chamado, às vezes, de minimundo ou de universo de discurso. As mudanças no minimundo são refletidas em um banco de dados.
- Um banco de dados é uma coleção lógica e coerente de dados com algum significado inerente. Uma organização de dados ao acaso (randômica) não pode ser corretamente interpretada como um banco de dados.;
- Um banco de dados é projetado, construído e povoado por dados, atendendo a uma proposta específica. Possui um grupo de usuários definido e algumas aplicações preconcebidas, de acordo com o interesse desse grupo de usuários;
- Um banco de dados pode ser de qualquer tamanho e de complexidade variável;
- Um banco de dados pode ser gerado e mantido manualmente ou pode ser automatizado (computadorizado).

Em outras palavras, um banco de dados possui algumas fontes das quais os dados são derivados, alguns níveis de interação com os eventos do mundo real e um público efetivamente interessado em seus conteúdos (ELMASRI; NAVATHE, 2015).

A importância da base de dados para o Aprendizado de Máquina, é que é dela que vão sair todas as informações/dados que alimentarão os modelos. A geração da base varia de acordo com o tipo de aplicação e manipulação dos dados. Para casos de aplicação em visão computacional, as bases são, geralmente, alimentadas com imagens. Já para casos de previsão do tempo, as bases são alimentadas com dados em forma de texto ou números organizados em matriz (como por exemplo planilhas em Excel).

### 2.6.5 Algoritmos baseados em Árvore de Decisão

Árvores de decisão são métodos não-paramétricos de aprendizado de máquina supervisionados, utilizados para classificação e previsão de dados. Essa metodologia é fundada no princípio da decomposição de um problema complexo em sub problemas mais simples (OSHIRO, 2013). Ou seja, uma árvore é construída por particionamentos recursivos no espaço das covariáveis.

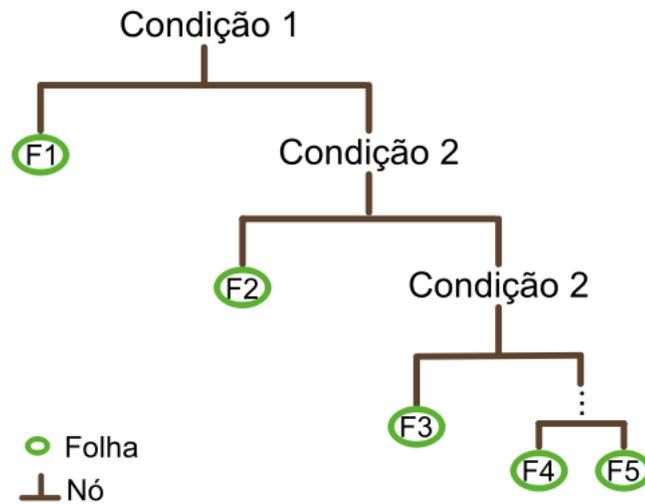
A construção da árvore baseia-se na escolha de atributos/variáveis que servirão para particionar/dividir os exemplos em subconjuntos. Cada particionamento recebe o nome de nó e cada resultado final o nome de folha (IZBICKI; SANTOS, 2019).

A utilização da árvore para prever uma nova observação é feita da seguinte maneira (OSHIRO, 2013):

- ❑ Primeiro é escolhido um atributo para particionar os exemplos em subconjuntos de acordo com os valores correspondentes desses atributos;
- ❑ Em seguida os exemplos são passados para os subconjuntos de acordo com o valor do atributo escolhido.
  - Quando todos os exemplos pertencerem a uma mesma classe, associa-se essa classe a um nó folha, contendo todos os exemplos desse subconjunto;
  - Caso os exemplos não pertençam todos a uma mesma classe, é escolhido um novo atributo para particionar os exemplos;
- ❑ Esse processo é repetido até que as classes dos exemplos de um subconjunto sejam homogêneas.

De modo a exemplificar o funcionamento de uma árvore de decisão temos:

Figura 29 – Funcionamento de uma árvore de decisão



Fonte: Autor.

A verificação começa pelo topo (condição 1), se a condição descrita for verdadeira, os exemplos são classificados para a esquerda. Caso contrário, eles são classificados para a direita, onde haverá outra condição (nó). E assim sucessivamente até atingir uma folha (IZBICKI; SANTOS, 2019).

A utilização de árvores de decisão como método de classificação e previsão possui algumas vantagens, e são elas (JAMES et al., 2013):

- O método de árvore de decisão é fácil de ser explicado;
- Acredita-se que as árvores de decisão refletem melhor a tomada de decisão humana;
- As árvores podem ser exibidas graficamente;
- Podem ser facilmente interpretadas, mesmo por um não especialista;
- As árvores podem lidar facilmente com preditores qualitativos sem a necessidade de criar variáveis fictícias.

Entretanto, o método de árvore de decisão geralmente não apresenta o mesmo nível de precisão preditiva que algumas outras abordagens de regressão e classificação (JAMES et al., 2013) (IZBICKI; SANTOS, 2019).

Notou-se que ao agregar muitas árvores de decisão o desempenho preditivo das árvores pode ser melhorado. Foi aí que surgiram métodos como *Bagging*, *Random Forest* e *Boosting* (JAMES et al., 2013).

### 2.6.5.1 *Bagging*

O método *Bagging* é um algoritmo de aprendizado *ensemble* mais antigo (BREIMAN, 1996). Ele utiliza amostras do tipo *bootstrap* do conjunto de treinamento (EFRON, 1979). A amostragem *bootstrap* é uma técnica de amostragem com reposição, que tem como fundamento criar um novo subconjunto de dados de treinamento com reposição a partir do conjunto de treinamento original (EFRON, 1979). Uma amostra *bootstrap* contém, em média, somente certa de 63,2% dos exemplos do conjunto de treinamento original, e com muitos exemplos de treinamento aparecendo múltiplas vezes (DIETTERICH, 2000).

Nesse método, diferentes subconjuntos de treinamento são formados aleatoriamente com reposição a partir do conjunto de treinamento completo. Cada subconjunto de treinamento é usado como entrada para os classificadores de base. Todos os classificadores extraídos são combinados usando um voto majoritário (BREIMAN, 1996).

A utilização do método *bagging* é muito eficaz quando os classificadores utilizados possuem um comportamento instável (como é o caso de árvores de decisão) (BREIMAN, 1996). Classificador instável é quando pequenas mudanças no conjunto de treinamento podem causar grandes mudanças no classificador gerado.

### 2.6.5.2 *Random Forest*

A definição formal de *Random Forest* dada por Breiman 2001 é: "Uma *random forest* é um classificador composto por uma coleção de árvores  $h(x, \theta_k)$ ,  $k = 1, 2, \dots, L$  sendo que  $\theta_k$  são vetores aleatórios independentes e identicamente distribuídos e cada árvore emite um voto único na classe mais frequente para o valor observado de  $X$ " (BREIMAN, 2001).

O método *Random Forest* (Floresta Aleatória) utiliza a combinação de diversas Árvores de decisão distintas com o intuito de melhorar o poder preditivo. Essa melhoria se dá por meio de um pequeno ajuste que descorrelaciona as árvores (JAMES et al., 2013). Segundo Santos 2018, a ideia central da *Random Forest* é modificar o método de criação das árvores para que estas se tornem diferentes umas das outras e possam reduzir a correlação existente entre as possíveis árvores.

A *Random Forest* é uma técnica de *machine learning ensemble*, isto é, um algoritmo de aprendizagem que constrói um conjunto de classificadores e, em seguida, classifica novas observações ao ponderar por meio de um voto, suas previsões (DIETTERICH, 2000). Essa técnica consiste em ajustar uma árvore de decisão em diferentes subconjuntos aleatórios criados a partir do conjunto de treinamento original via amostras *bootstrap* afim de reduzir a variância de um modelo de classificação (BREIMAN, 1996).

É utilizada amostragem *Bootstrap* para produzir amostras aleatórias do conjunto de treinamento. A árvore é construída usando esse novo subconjunto e uma seleção aleatória de covariáveis. A cada nó da árvore, um subconjunto de  $m$  covariáveis é selecionado

aleatoriamente e validado. Dessa forma, a covariável com mais poder preditivo é escolhida para dividir o nó (OSHIRO, 2013).

Assim como a Árvore de Decisão, a técnica de Floresta Aleatória também apresenta vantagens, e são elas:

- Altamente robusto contra ruídos e mais adequado para lidar com observações repetidas envolvendo dados de sensoriamento remoto que geralmente são afetados pela atmosfera, nuvens, tempos de observação e ruídos do sensor (VILJANEN et al., 2018);
- Menor sensibilidade a sobreajustes (VILJANEN et al., 2018);
- Excelentes resultados de classificação e velocidade de processamento (BELGIU; DRĂGUȚ, 2016);

### 2.6.5.3 *Boosting*

O método *Boosting* cria diferentes classificadores base reponderando sequencialmente os exemplos no conjunto de treinamento. No início, todos os exemplos são inicializados com pesos iguais. Cada exemplo classificado erroneamente pelo classificador anterior recebe um peso maior na próxima iteração do treinamento, a fim de tentar classificá-lo corretamente. O erro é computado, o peso dos exemplos classificados corretamente é reduzido e o peso dos exemplos classificados incorretamente é aumentado. O voto de cada classificador individual é ponderado proporcionalmente ao seu desempenho (WANG et al., 2011).

Os primeiros algoritmos eficazes de *boosting* foram apresentados por Schapire em 1990 (SCHAPIRE, 1990) e por Freund em 1996 (FREUND; SCHAPIRE, 1996). Depois dessas duas propostas iniciais, os estudos envolvendo a técnica *boosting* continuaram avançando e novas propostas surgiram, como por exemplo o *AdaBoost* (FREUND; SCHAPIRE, 1996). Entretanto, nesse projeto será abordado a técnica *XGBoost*, derivada do método *boosting*.

#### □ *XGBoost*

O nome *XGBoost* vem de *eXtreme Gradient Boosting* e dentro dos algoritmos baseados em Árvores de decisão, ele faz uso da técnica chamada de *Gradient Boosting*.

*Gradient Boosting*, como o próprio nome já diz, é uma técnica *Boosting*, incluída no grupo de classificadores *Ensemble*. *Ensemble* são classificadores que utilizam uma combinação de resultados de preditores fracos, com o objetivo de produzir um melhor modelo preditivo. É classificado como modelo preditivo fraco, aqueles modelos que quando utilizados individualmente possuem uma acurácia abaixo do esperado.

O objetivo do *Gradient Boosting* é criar uma corrente de modelos fracos, em que cada um tem como objetivo minimizar o erro do modelo anterior. Ele constrói o modelo em etapas e os generaliza, permitindo a otimização de uma função de perda. Aos ajustes de cada modelo fraco é multiplicado um valor chamado de taxa de aprendizagem. Esse valor tem como objetivo determinar o impacto de cada árvore no modelo final (FRIEDMAN, 2001).

O algoritmo de *gradient boosting* funciona da seguinte maneira (FRIEDMAN, 2001):

- Um primeiro modelo é criado com uma aproximação simples. Essa aproximação resulta em um resíduo, que é a distância entre o valor previsto e o valor real;
- Um segundo modelo é criado e ajustado em cima do resíduo gerado pelo modelo anterior. Em seguida, um novo resíduo é calculado com esse segundo modelo;
- Essas iterações são repetidas por um determinado número de vezes, buscando sempre minimizar o resíduo gerado pelos modelos fracos, ou seja, até que a distância entre o valor previsto e o valor real seja o menor possível;
- O modelo final é a soma dos ajustes de todos os modelos fracos.

O sucesso por trás do *XGBoost* é a sua escalabilidade em todos os cenários. Essa alta escalabilidade se deve a várias otimizações algorítmicas, e são elas (CHEN; GUESTRIN, 2016):

- Novo algoritmo de aprendizado de árvore desenvolvido para lidar com dados esparsos;
- Um procedimento de esboço de quantil ponderado teoricamente que permite manipular os pesos das instâncias no aprendizado;
- Exploração mais rápida do modelo, devido ao uso de computação paralela e distribuída que torna o modelo mais rápido;
- E a mais importante, segundo Chen, o *XGBoost* explora a computação fora do núcleo, isso permite o processamento de centenas de milhões de exemplos em um *desktop*.

## 2.6.6 Métricas de avaliação de performance

Ao desenvolver modelos de *Machine Learning*, é crucial mensurar a performance do modelo. Para isso existem métricas que avaliam matematicamente a qualidade do modelo proposto .

Um modelo de classificação, como o proposto nesse projeto, tem como objetivo decidir em qual classe uma nova observação pertence dentre as classes possíveis. Em geral a classificação pode ocorrer de duas maneiras: positiva (P) ou negativa (N), e indicam a ocorrência ou não de um determinado evento. Um exemplo prático seria classificar se um determinado *superpixel* possui formigueiro (positivo) ou não (negativo).

A avaliação de um modelo de classificação é feita a partir da comparação entre as classes preditas pelo modelo e as classes verdadeiras de cada exemplo. Todas as métricas de classificação têm como objetivo comum medir quão distante o modelo está da classificação perfeita, porém fazem isto de formas diferentes (KRÜGER, 2016). Será abordado abaixo as métricas para classificação mais utilizadas.

### 2.6.6.1 Matriz de Confusão

Uma das formas mais simples de visualizar a performance de um modelo de classificação é através de uma matriz de confusão. Esta matriz indica quantos exemplos existem em cada grupo: falso positivo (FP), falso negativo (FN), verdadeiro positivo (TP) e verdadeiro negativo (TN) (HOSSIN; SULAIMAN, 2015)(KRÜGER, 2016).

Figura 30 – Matriz de confusão teórica

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

Fonte: Autor.

É interessante visualizar a contagem destes grupos tanto em números absolutos quanto em porcentagens da classe real, já que o número de exemplos em cada classe pode variar (KRÜGER, 2016).

Figura 31 – Matriz de confusão exemplo

		Valor Predito	
		Sim	Não
Valor Real	Sim	240 (80%) Verdadeiro Positivo	60 (20%) Falso Negativo
	Não	135 (9%) Falso Positivo	1365 (91%) Verdadeiro Negativo

Fonte: Autor.

A matriz de confusão permite visualizar facilmente quantos exemplos foram classificados corretamente e erroneamente em cada classe. Esse tipo de análise ajuda a entender se o modelo está favorecendo uma classe em detrimento da outra.

### 2.6.6.2 Acurácia

A acurácia, do inglês *accuracy*, é a métrica que apresenta a quantidade de amostras que foram classificadas corretamente, independente da classe (KRÜGER, 2016). A acurácia é definida pela fórmula abaixo:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Em que,  $TP$  é o verdadeiro positivo,  $TN$  é o verdadeiro negativo,  $FP$  é o falso positivo e  $FN$  é o falso negativo.

Na fórmula, pode-se observar que esta métrica é definida pela razão entre o que o modelo acertou e todos os exemplos. Apesar da acurácia ser uma métrica simples, de fácil uso e interpretável, ela muitas vezes não é adequada na prática (KRÜGER, 2016).

Uma das maiores desvantagens é que em algumas aplicações a acurácia pode ser elevada mas, ainda assim, o modelo pode ter uma performance inadequada (HOSSIN; SULAIMAN, 2015).

Por exemplo, considere o modelo que classifica exames de câncer entre positivo ou negativo para a doença, e no conjunto de dados tem 1000 exemplos, sendo 990 de pacientes sem câncer e 10 de pacientes com câncer. Caso o modelo sempre classifique todos os exemplos com negativo (sem câncer), ele ainda obteria uma acurácia de 99%. O que parece uma excelente métrica, mas na verdade o modelo não está sendo avaliado de forma adequada. Para melhor avaliar modelos que possuem um conjunto de dados desbalanceados, como o do exemplo anterior, outras métricas devem ser utilizadas (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

### 2.6.6.3 Precisão

A precisão, do inglês *precision*, também é uma das métricas mais comuns para avaliar modelos de classificação. Esta métrica é definida pela razão entre a quantidade de exem-

plos classificados corretamente como positivos e o total de exemplos classificados como positivos (KRÜGER, 2016), conforme a fórmula abaixo:

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (20)$$

Em que,  $TP$  é o verdadeiro positivo e  $FP$  é o falso positivo.

Ao observar a fórmula da precisão, nota-se que a precisão dá um ênfase maior para os erros por falso positivo. Pode-se entender a precisão como sendo a expressão matemática para a pergunta: dos exemplos classificados como positivos, quantos realmente são positivos? Considerando o modelo de câncer como exemplo, se o valor para a precisão fosse de 90%, isto indicaria que a cada 100 pacientes classificados como positivo, é esperado que apenas 90 tenham de fato a doença (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

#### 2.6.6.4 *Recall*

O *recall*, também conhecido como sensibilidade ou taxa de verdadeiro positivo (TPR), ao contrário da precisão, dá maior ênfase para os erros por falso negativo. Esta métrica é definida pela razão entre a quantidade de exemplos classificados corretamente como positivos e a quantidade de exemplos que são de fato positivos (KRÜGER, 2016), conforme a fórmula abaixo:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (21)$$

Em que,  $TP$  é o verdadeiro positivo e  $FN$  é o falso negativo.

O *recall* busca responder a seguinte pergunta: de todos os exemplos que são positivos, quantos foram classificados corretamente como positivos? Considerando o exemplo do modelo de câncer novamente, se o valor para do *recall* fosse de 95%, isto indicaria que a cada 100 pacientes que são de fato positivos, é esperado que apenas 95 sejam corretamente identificados como doentes (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

#### 2.6.6.5 *F1-Score*

A métrica *F1-Score*, também conhecida como *F-measure*, leva em consideração tanto a precisão quanto o *recall* (KRÜGER, 2016). Ela é definida pela média harmônica entre as duas, como pode ser visto abaixo:

$$F1_{Score} = 2 \frac{\text{precisão} * \text{recall}}{\text{precisão} + \text{recall}} \quad (22)$$

$$F1_{Score} = \frac{2TP}{2TP + FP + FN} \quad (23)$$

Em que,  $TP$  é o verdadeiro positivo,  $FP$  é o falso positivo e  $FN$  é o falso negativo.

Uma das características da média harmônica é que se a precisão ou o *recall* for zero ou muito próximos disso, o *F1-score* também será baixo. Desta forma, para que o *F1-score* seja alto, tanto a precisão como a revocação também devem ser altas. Ou seja, um modelo que apresenta um bom *F1-score* é um modelo capaz tanto de acertar suas predições (precisão alta) quanto de recuperar os exemplos da classe de interesse (*recall* alto). Portanto, esta métrica tende a ser um resumo melhor da qualidade do modelo (KRÜGER, 2016).

Uma desvantagem é que o *F1-score* acaba sendo menos interpretável que a acurácia.

No exemplo do modelo de câncer, se o valor para a precisão fosse de 90% e o do *recall* fosse de 95%, o valor para do *F1-score* seria de 92.43% (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

---

## Formulação do Problema

### 3.1 Possibilidade de entrar no mercado

Além da contribuição científica, visa-se também a transformação do presente projeto em um produto/serviço a ser comercializado.

Algumas escolhas feitas ao longo do desenvolvimento do projeto, contribuem para que essa comercialização seja possível. A escolha do uso de um drone comercial e de câmera RGB convencional são exemplos dessas escolhas.

O drone Fimi X8 SE 2020 é um drone comercial, semi profissional, que pode ser facilmente encontrado em *e-commerces*. Seu preço varia entre 3 e 5 mil reais. Ele possui câmera RGB integrada, mais um ponto que facilita transformação do projeto final em produto.

A utilização de câmera multi/hiperespectral foi considerada, entretanto alguns pontos negativos fizeram com que a sua escolha fosse desconsiderada. Esses pontos negativos são:

- Drone comercial já com câmera multiespectral integrada possui um alto custo (cerca de 89 mil reais);
- A compra da câmera avulsa e adaptação da mesma em um drone comercial possuem dificuldades de implementação, além do valor elevado da câmera (entre 20 e 100 mil reais) (ELETRÔNICOS..., 2021). Uma dessas dificuldades é o acoplamento adequado da câmera no drone. Outra dificuldade é com relação ao aumento no peso final do drone, e sabe-se que quanto mais pesado o drone fica, maior será seu consumo de energia, acarretando em menor autonomia;

O uso da câmera RGB convencional abaixa o valor gasto em material e em mão de obra, o que conseqüentemente abaixaria do valor final do produto.

Segundo Amante, a presença de 10 colônias de *A. capiguara* por hectare, consome a quantia aproximada de 52,5 kg de capim por dia, o que equivale à ração diária de 3 bois em regime de pasto. Com base nesses dados, um boi come cerca de 17,5kg de pasto por dia, e um formigueiro adulto degrada 5,25kg de pasto por dia, um formigueiro adulto

"come"o equivalente a 0,3 boi por hectare. Esse valor é o número de cabeças de gado perdida por hectare para cada uma colônia presente no hectare (AMANTE, 1967b).

De acordo com o Censo Agropecuário de 2017, o número de cabeças de gado por hectare é igual a 1,15. Então um formigueiro adulto comer o equivalente a 0,3 cabeça de gado por hectare implica em uma queda de produção de quase um quarto.

Um cálculo simples foi desenvolvido afim de prever o prejuízo financeiro que as formigas podem causar ao dono de cabeças de gado. O cálculo pode ser observado na equação abaixo:

$$Prejuízo = \left( \frac{CPG}{Form} \right) * nForm * \left( Valor@ * \frac{Pmin}{15} \right) \quad (24)$$

Em que:

- CGP = número de cabeças de gado perdida por hectare;
- nForm = Número de formigueiro por hectare;
- Form = 1 formigueiro;
- Valor@ = cotação do arroba do boi;
- Pmin = Peso mínimo do boi.

Considerando 0,3 o número de cabeças de gado perdida por hectare, 1 o número de formigueiro por hectare, R\$260,45 a cotação do arroba (CAMBIO, 2023) e 200kg o peso mínimo do boi (SAÚDE, 2021). De acordo com a Equação 24, tem-se um prejuízo de R\$1041,80 por hectare.

$$Prejuízo = 0,3 * 1 * \left( 260,45 * \frac{200}{15} \right)$$

$$Prejuízo = 1041,80$$

Esse prejuízo de R\$1041,80 é considerando sempre os menores valores possíveis, o que significa que se tiver mais de um formigueiro por hectare aumenta esse prejuízo. Assim como, o peso do boi para abate foi considerado um peso de novillo super precoce, se o boi for abatido no peso recomendado esse prejuízo é ainda maior. E tudo isso ainda depende da cotação do arroba, que para esse exemplo foi considerado o menor valor em um período de três meses (março, abril, maio).

Tendo esses valores em vista e o alto prejuízo financeiro que as formigas cortadeiras podem causar para os empresários do ramo, há uma forte possibilidade de negócio como projeto futuro.

Como possibilidade de negócio foram pensados alguns modelos que serão estudados afim de ver o que melhor se aplica. São eles:

- Consultoria com equipamento: O empresário contrata a consultoria e nós vamos até a propriedade com nosso equipamento (drone), realizamos a detecção e entregamos um relatório de infestação;

- ❑ Consultoria sem equipamento: O empresário realiza a coleta de dados com equipamento próprio e nós realizamos a análise/detecção e entregamos o relatório de infestação;
- ❑ Para ambos os casos ainda pode-se sugerir um manejo adequado de acordo com a situação da infestação e talvez até prestar o serviço de manejo também;
- ❑ Pacote de serviço por anuidade: O empresário paga um valor anual para realização de análises periódicas de infestação, para fins de acompanhamento, seja ele "pré infestação" ou "pós manejo";
- ❑ Desenvolvimento de um *app*: Dentro do *app* estariam todos as análises, relatórios de infestação, informações sobre manejos sugeridos (ou manejo realizado). O *app* seria para fins de acompanhamento para aqueles que adquirirem o pacote de serviço por anuidade.

Todas essas possibilidades de negócio citadas ainda não foram estudadas, são apenas ideias que podem nortear uma escolha futura.

## 3.2 Tamanho a ser buscado

De acordo com o levantamento realizado na Revisão da Literatura, não foi encontrado um tamanho/idade ideal a ser buscado. Sendo assim, o tamanho e/ou idade do formigueiro não foi determinado previamente para a captura das imagens. Dessa maneira, nas imagens obtidas terão formigueiros maiores e formigueiro menores.

## 3.3 Dificuldades enfrentadas

Durante a etapa de revisão da literatura e de aquisição das imagens, foi possível identificar problemas que precisaram ser contornados. A saber:

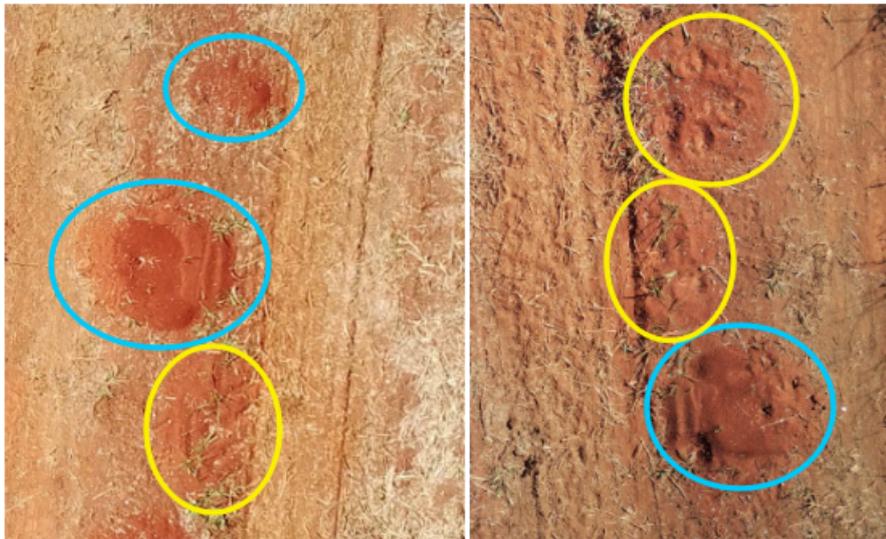
- ❑ Altura de voo (Subsecção 3.3.1);
- ❑ Horário de voo (Subsecção 3.3.2);
- ❑ Condições climáticas (Subsecção 3.3.3);
- ❑ Morfologia do formigueiro e preferência de localização (Subsecção 3.3.4);

### 3.3.1 Altura de voo

A altura de voo impacta diretamente na área que será abrangida por cada imagem e na resolução dos *pixels* dessa imagem. Assim, uma altura de voo que balanceie uma

boa área sem prejudicar a resolução deve ser encontrada. A saber, voos altos permitem uma varredura de área maior por foto, o que por sua vez, o habilita a voar mais rápido e cobrir a área desejada em menos tempo, parâmetro que pode ser importante para o futuro cliente. Entretanto, a queda na resolução do pixel pode impactar negativamente a detecção de formigueiros, uma vez que detalhes sutis que se observa quando se está perto de um, já não são completamente capturados a grandes alturas, podendo aumentar a confusão com classes como montes de terra (Fig. 32) e cupinzeiros (Fig. 33).

Figura 32 – Casos em que os formigueiros se assemelham ao solo. Marcados por um especialista, os formigueiros estão identificados em azul e, em amarelo estão os que não se sabem se é somente solo ou formigueiro



Fonte: Autor.

Baseado nessas considerações, optou-se por realizar a captura das imagens a uma altura de 10 a 20 metros do solo. A altura foi determinada empiricamente observando as primeiras imagens de teste que foram registradas em alturas variadas.

### 3.3.2 Horários de voo

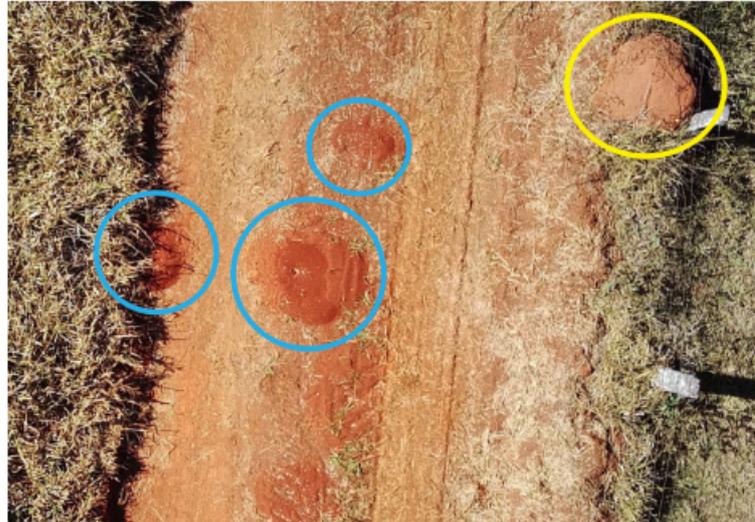
Os horários de voo são considerados dificuldades por dois motivos: oclusão do formigueiro por sombra e cromaticidade.

A oclusão ocorre causada pela sombra da vegetação sobre o formigueiro. Isso varia de acordo com a posição do sol em determinado horário do dia (Fig. 34).

De acordo com o Dicionário Online de Português, Cromaticidade é definida por "Qualidade de cor caracterizada por seu comprimento de onda dominante ou complementar e sua pureza, considerados em conjunto". Para fins de definição, pode-se considerar o comprimento de onda dominante como a Matiz e a pureza como a Saturação.

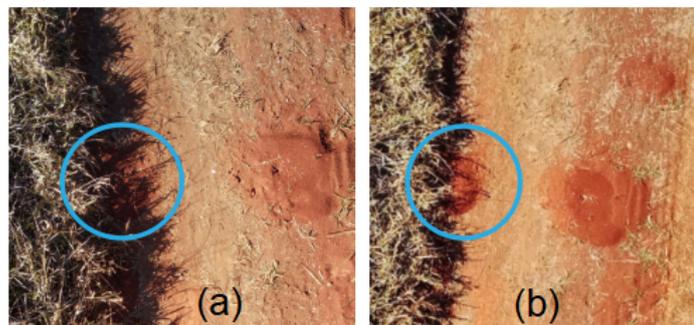
A palavra "cromaticidade" pode ser considerada uma junção da palavra "cromático" e do sufixo "dade" (DICIONÁRIO, 2021). "Cromático" por sua vez, deriva da palavra "croma",

Figura 33 – Casos ainda em que os formigueiros se assemelham a montes de cupim. Marcados por um especialista, os formigueiros estão identificados em azul e o cuínzeiro em amarelo.



Fonte: Autor.

Figura 34 – Formigueiros Oclusos. Em (a) Oclusão por sombra. Em (b) Oclusão parcial pela vegetação.



Fonte: Autor.

e "Croma" refere-se à pureza e à intensidade de uma cor (CHROMAFLO, 2021). O sufixo "dade" designa "qualidade, modo de ser, estado, propriedade" (PEZATTI, 1990).

Depois de definir o que é cromaticidade, parte-se para o motivo de tal propriedade ser um problema a ser enfrentado no desenvolvimento do presente projeto.

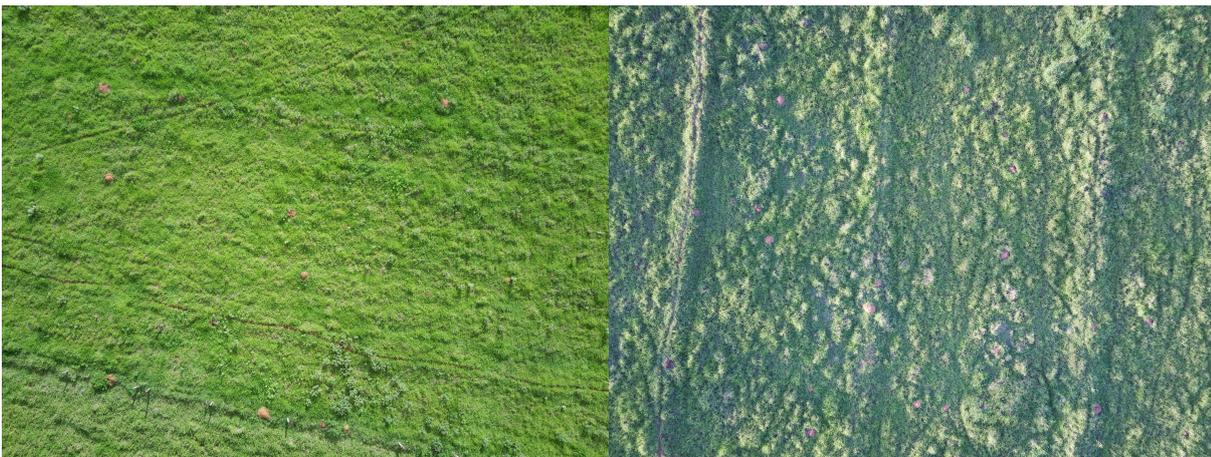
Foi visto que o que afeta a cromaticidade são alterações que podem ocorrer na saturação e/ou na matiz. O período do dia, as condições climáticas e Dispersão de Rayleigh podem afetar esses parâmetros que causam a cromaticidade.

Considerando que a cor real do objeto (saturação máxima) é a cor de um objeto quando iluminado por uma fonte de cor branca perfeita, qualquer alteração nessa fonte de luz impacta na cor real, resultando em possíveis divergências nos espectros de cores esperados para detecção do objeto.

Essa alteração na fonte de luz pode resultar em alteração na saturação e na matiz, que resulta em alteração de cromaticidade. Segundo Rayleigh, dispersão de Rayleigh é processo de espalhamento da luz por partículas muito menores que o comprimento de onda da radiação eletromagnética incidente, e essa é a causa da mudança na cor do céu, e conseqüentemente da luz que chega ao solo terrestre (RAYLEIGH, 1899). Esse fenômeno é o responsável pela cor azul do céu durante a maior parte do dia e também explica o que acontece durante o nascer ou pôr do sol. No caso do céu alaranjado, o fenômeno ocorre quando as partículas maiores em suspensão na atmosfera (geralmente poeira, pólen, sal marinho e minúsculas gotículas de água) espalham a luz solar na faixa dos comprimentos de onda que compreendem as cores que vão do vermelho ao laranja. Isso ocorre principalmente durante o nascer e o pôr do sol pois, a luz solar percorre uma camada maior na atmosfera, interagindo com mais partículas que estejam em suspensão.

Na Figura 35 pode-se observar o efeito do horário do dia e da Dispersão de Rayleigh na cromaticidade das imagens.

Figura 35 – Efeitos da cromaticidade.



(a) Imagem registrada por volta das 15:00 horas

(b) Imagem registrada por volta das 17:00 horas

Fonte: Autor.

Pensando nas principais variáveis que afetam a cromaticidade de uma imagem, optou-se por concentrar a captura das imagens em horários de maior incidência direta de luz solar (por volta de meio dia).

### 3.3.3 Condições climáticas

As condições climáticas também afetam a cromaticidade da imagem. Dado que em um dia ensolarado os elementos da imagem apresentam uma determinada cromaticidade e em dias nublados essa cromaticidade sofre alterações. Uma vez que, tendo a luz solar como fonte de luz, dias nublados impedem parcialmente a passagem dessa luz, o que resulta

em alterações na fonte de luz ideal, que afetam a cromaticidade, portanto, optou-se por concentrar a captura das imagens em dias ensolarados.

### 3.3.4 Morfologia do formigueiro e preferência de localização

Com base na revisão da literatura realizada, concluiu-se que a localização e a morfologia externa dos formigueiros variam de acordo com a espécie da formiga. Analisando as diferenças de características dos ninhos, decidiu-se então focar na detecção de ninhos de formiga *Atta Capiguara*.

Tal escolha foi baseada no fato de que as formigas *A. capiguara* tem predileção por áreas de pastagem, canaviais e campos abertos com grande insolação. Tal preferência facilita a detecção, dado que voar com o drone no interior de plantações, como de eucalipto por exemplo, enfrentam problemas de obstáculos (galhos) e perda de sinal GPS.

Outro ponto que motivou a escolha por essa espécie foi a morfologia externa do ninho, proveniente da deposição de terra retirada para construção das câmaras e canais internos. As formigas de *A. capiguara* realizam a deposição formando um grande monte de terra solta e montes menores ao redor. Acredita-se que essa caracterização de tamanho e disposição ajudará na identificação e detecção do ninho.

Além desses dois motivos, ainda há uma colaboração direta relacionada a dificuldade de controle dessa espécie (AMANTE, 1967a). Tal dificuldade é devido a seus hábitos de construir ninhos a grandes profundidade, hábito esse que dá as colônias dessa espécie maior proteção contra seus inimigos naturais.

Apesar da *A. capiguara* ter preferência por áreas de pastagem e campos com grande insolação, algumas dificuldades foram observadas durante a execução das campanhas experimentais. Dentre elas temos: Oclusão do formigueiro, Dificuldade de diferenciação entre formigueiro, solo e cupinzeiro.

## 3.4 Sumário das decisões

Sumarizando as informações apresentadas nesse capítulo temos:

- ❑ Espécie a ser detectada: *Atta capiguara*;
- ❑ Sem controle do tamanho do formigueiro a ser capturado;
- ❑ Altura de voo na faixa de 10 a 20 metros;
- ❑ Fotos tiradas em dias ensolarados e das 10:00 às 16 horas.

Apesar das dificuldades, ainda há uma grande oportunidade de negócio, tendo em vista que a escolha dos materiais e métodos foram feitos a fim reduzir o custo final do projeto.



---

## Materiais e Métodos

Considerando o disposto no capítulo 3, foi escolhido um drone comercial com câmera RGB, para a aquisição das imagens, suas especificações podem ser encontradas no Apêndice B. A linguagem escolhida foi o *python* por sua facilidade de prototipação e integração com IDEs como o *PyCharm* ou *Jupyter Notebook*. Além disso, boa parte das bibliotecas que este trabalho se baseou foram encontradas em *python*, além das bibliotecas desenvolvidas dentro do laboratório a que se pode ter acesso.

Foca-se, agora, em descrever os materiais e métodos pertinentes ao algoritmo de classificação.

### 4.1 Aquisição de dados

As campanhas experimentais foram divididas em duas etapas. Na primeira etapa, foram realizados testes de forma exploratória, que tinham como intuito testar diversas alturas de voo e horários do dia. Já na segunda etapa foram realizadas as campanhas experimentais controladas, com altura de voo e horário do dia definidos no capítulo 3, essa etapa foi realizada focada na obtenção das imagens que seriam utilizadas no modelo de classificação.

As campanhas experimentais foram realizadas em dois municípios em estados diferentes. Foram adquiridas imagens no município de Piracanjuba - GO e no município de São Carlos - SP.

Piracanjuba é um município do estado de Goiás e fica localizado nas coordenadas geográficas: 17° 18' 10"S, 49° 01' 04"O. Seu bioma é de cerrado e seu solo é classificado como Latossolos Vermelho Ácrico (SOLOS, 2018) (PRONASOLOS, 2023).

São Carlos é um município do estado de São Paulo e fica localizado nas coordenadas geográficas: 22° S, 47° 53' 27"O. Seu bioma é de transição entre Mata Atlântica e Cerrado e seu solo é classificado como Latossolos Vermelho Amarelo (SOLOS, 2018) (PRONASOLOS, 2023).

Apesar das imagens serem obtidas em meses distintos, sempre foi observado o horário de voo e os dias ensolarados. Na Tabela 6 pode ser observados a localização, os períodos, a quantidade de imagens obtidas em cada campanha experimental controlada e a quantidade de imagens utilizadas no modelo.

Tabela 6 – Aquisição de imagens

Campanha	Localização	Período	Total imagens obtidas	Quantidade utilizada no modelo
1	Piracanjuba	jun/21	66	11
2	São Carlos	jul/22	44	24

Fonte: Autor.

As campanhas experimentais preliminares foram realizadas em estações do ano distintas e com isso foi possível observar que em época de chuva alguns formigueiros, que antes haviam sido encontrados, sumiam ou estavam deformados. Isso ocorre porque as chuvas alteram as características externas do formigueiro, seja destruindo-o por completo, compactando-o ou alterando seu tamanho. Portanto, as campanhas experimentais controladas foram realizadas no inverno, estação de pouca chuva.

## 4.2 Determinação dos parâmetros do *Super Pixel*

Para que o *Super Pixel* seja gerado em uma imagem, são necessários dois parâmetros principais para o algoritmo do SLIC, a quantidade de *Super Pixel* contidos em uma imagem, ( $n\_seg$ ) e o grau de "compactação" ( $comp$ ) do *Super Pixel* (SP), ou seja, se o SP vai seguir uma forma fixa ou se ele vai se adequar aos contrastes da imagem. Há a possibilidade de se usar um filtro passa-baixa ou filtro de suavização antes de se aplicar o SLIC, nesse caso, o desvio-padrão  $\sigma$  foi fixado em 0,5.

O processo de determinação desses parâmetros foram divididos em 4 etapas:

### □ Etapa 1: Testes visuais

Nessa primeira etapa foram testados valores aleatórios de  $n\_seg$  e  $comp$ , afim de se determinar um intervalo menor de valores para ser utilizados nas etapas seguintes;

Dos testes visuais, ficou estabelecido que as etapas seguintes seriam realizadas com os seguintes valores:  $n\_seg$  entre 700 e 1700, com um incremento de 500 a partir do valor inicial estabelecido. E  $comp$  entre 20 a 60, com incremento de 20 a partir do valor inicial.

### □ Etapa 2: Criação de características para qualificação da variação dos parâmetros estabelecidos na etapa 1

Nesse etapa foram criadas quatro características que tem como objetivo qualificar, em porcentagem, a adequação do SP a cada parâmetro. As características são:

- *feat\_1*: Porcentagem que representa, de maneira geral, o quanto o formigueiro foi bem delimitado pelo(s) *Super Pixel(s)*;
- *feat\_2*: Porcentagem referente ao quanto as bordas do formigueiro coincidem com as bordas do *Super Pixel*;
- *feat\_3*: Cada *Super Pixel* que tem alguma parte de formigueiro recebe uma porcentagem referente a quantidade de formigueiro presente no mesmo, depois é realizado o calculo de Média aritmética dessas porcentagens.

A diferença entre '*feat\_1*' e '*feat\_3*' é que a '*feat\_1*' olha para o formigueiro inteiro e trás a porcentagem do quanto o formigueiro foi atendido pelos SP (independente de quantidade de *Super Pixels* necessários para cobrir o formigueiro). Já '*feat\_3*' olha para cada um desses *Super Pixels* necessários para cobrir o formigueiro e atribui uma porcentagem a ele, e posteriormente calcula uma média aritmética para compilar essas porcentagens.

#### □ Etapa 3: Atribuição de valores as características criadas

Os valores foram atribuídos de maneira manual de acordo com a observação da mestranda e priorizando ao máximo não enviesar os resultados;

Tanto para '*feat\_1*' quanto para '*feat\_2*' foram atribuídas porcentagens de 0, 25, 50, 75 ou 100%

#### □ Etapa 4: Análise fatorial. Ver subseção 4.2.1

### 4.2.1 Análise fatorial

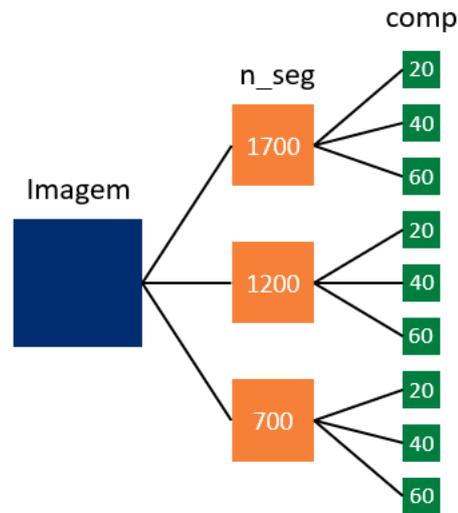
Para realização da etapa de análise fatorial, foram escolhidas 7 imagens, com os cenários mais variados possíveis. Essas 7 imagens passaram por segmentações por meio de *Super pixels*, tendo seus valores de *n\_seg* e *comp* variados de acordo com o estabelecido na etapa 1.

O processo esquematizado na Figura 36 foi aplicado para cada uma das 7 imagens. Isso resultou em 72 linhas de dados.

Na sequencia, foram atribuídos valores as características, criadas na etapa 2, de acordo com o estabelecido na etapa 3.

Como resultado, obteve-se uma matriz de 72 linhas por 7 colunas. As colunas sendo elas: "*img\_name*", "*n\_seg*", "*comp*", "*%feat\_1*", "*%feat\_2*", "*%feat\_3*" e "*%end*". Em que "*%end*" é a média aritmética dos valores atribuídos às três *features*.

Figura 36 – Esquema da análise fatorial



Fonte: Autor.

Após preencher toda a matriz com os valores correspondentes, utilizou-se a função "MEDIASES" do *software Microsoft Excel* para o cálculo das médias aritméticas condicionais. Como resultado obteve-se as Tabela 7 e Tabela 8:

Tabela 7 – Média aritmética condicional de "n\_seg"

n_seg/comp	Análise de n_seg		
	700	1200	1700
20	51,20	59,39	62,79
40	58,91	68,04	76,73
60	59,17	65,09	70,86

Fonte: Autor.

Tabela 8 – Média aritmética condicional de "comp"

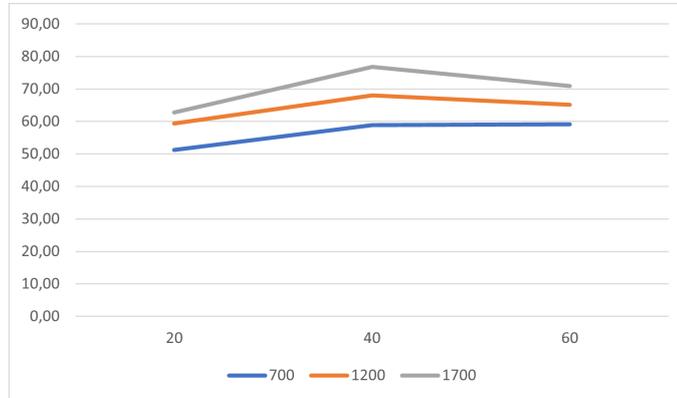
comp/n_seg	Análise de comp		
	20	40	60
700	51,20	58,91	59,17
1200	59,39	68,04	65,09
1700	62,79	76,73	70,86

Fonte: Autor.

Afim de obter-se os melhores valores para *n\_seg* e *comp*, foram construídos dois gráficos. Seu objetivo é encontrar o melhor resultado baseado nas interações entre as

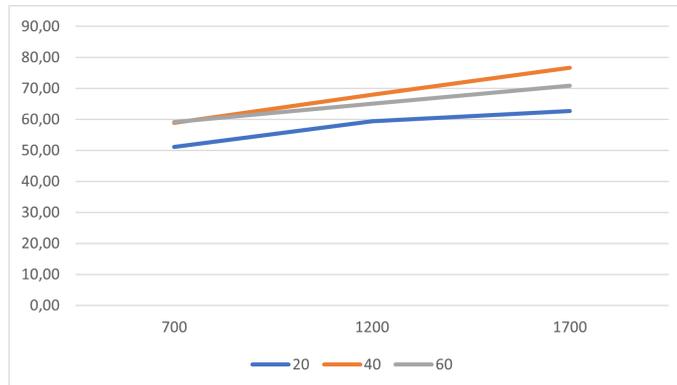
duas variáveis. Com as informações das Tabela 7 e Tabela 8, construiu-se Figura 37 e Figura 38:

Figura 37 – Gráfico de interações  $n\_seg$  x  $comp$



Fonte: Autor.

Figura 38 – Gráfico de interações  $comp$  x  $n\_seg$



Fonte: Autor.

Observando o gráfico da Figura 37 é possível notar que o valor de  $comp$  que apresentou melhor desempenho foi 40 dado que o gráfico apresenta um pico nesse valor, e depois o desempenho começa a cair.

Considerando que o valor de  $comp$  já foi estabelecido, a análise para a escolha de  $n\_seg$  ficou restrita. É possível observar que, na Figura 38, a linha laranja referente ao  $comp = 40$ , e ela tende a crescer quanto maior for o valor de  $n\_seg$ , entretanto, quanto maior for a quantidade de *Super pixels* que compor a imagem, será necessário maior processamento de dados para as etapas seguintes, então optou-se por utilizar o valor de 1700 para o parâmetro  $n\_seg$  afim de não gerar um esforço computacional muito elevado.

Ficando assim determinado que os valores para os parâmetros de *Super pixel* seriam:  $N\_seg = 1700$  e  $comp = 40$ . Esses valores são colocados manualmente dentro dos códigos que foram desenvolvidos e utilizados.

## 4.3 Classificação Manual

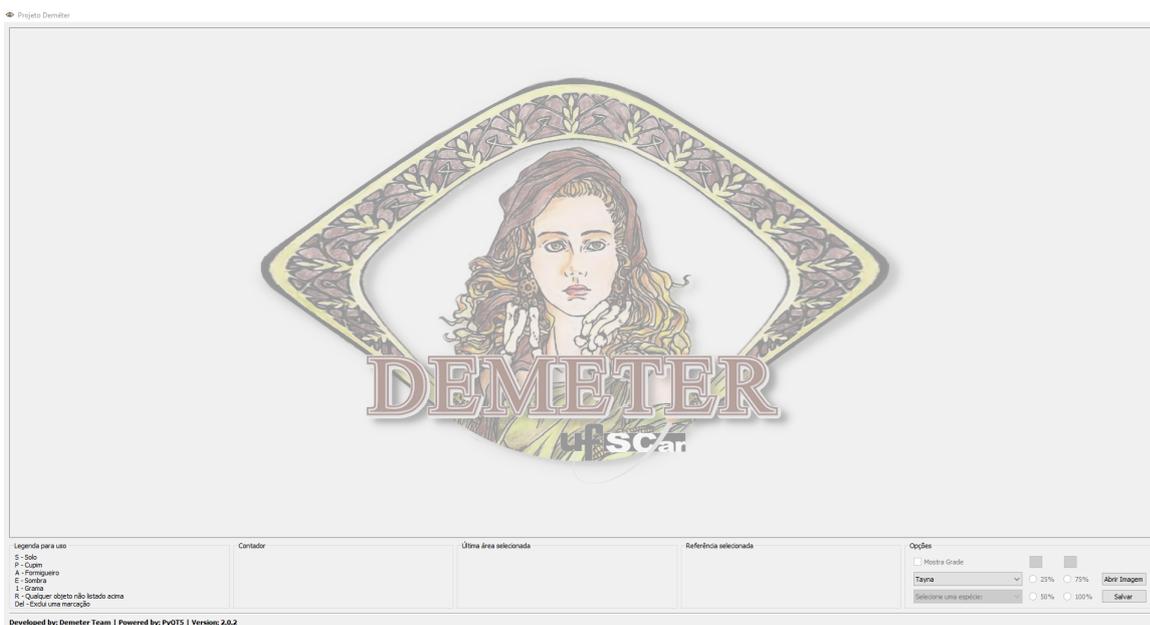
Baseado no referencial teórico estudado até o momento, optou-se por utilizar o método de aprendizagem supervisionada. Em decorrência dessa escolha, se fez necessária uma etapa de classificação/rotulação manual dos dados para que, futuramente, o algoritmo de *machine learning* escolhido tivesse como medir a precisão de sua classificação.

Dessa maneira, utilizou-se o *software* "Demeter" para atribuição de *labels* aos *Super Pixels*, a fim de obter uma base de dados classificada para treinamento do algoritmo de *machine learning*.

### 4.3.1 Demeter

Demeter é um *software* de rotulação manual de imagens, desenvolvido no ano de 2020 pelos pesquisadores André Carmona Hernandez, Ivan Carlos Perissini e André Luiz Carvalho Silva Santos, como parte da patente depositada (PERISSINI et al., 2022).

Figura 39 – IDE Demeter



Fonte: Autor.

Com acesso ao código fonte do software de rotulação, foi possível fazer modificações mínimas para atender as necessidades e peculiaridades deste trabalho. A saber, as alterações realizadas foram relacionadas as classes, reduzindo para 6 (seis). Sendo elas:

- ❑ Solo: classe que representa regiões que contenham em sua maioria solo/terra na cena;
- ❑ Formigueiro: classe que representa regiões que contenham em sua maioria formigueiro na cena;

- ❑ Sombra: classe que representa regiões que contenham em sua maioria sombra na cena;
- ❑ Grama: classe que representa regiões que contenham em sua maioria vegetação na cena;
- ❑ Cupim: classe que representa regiões que contenham em sua maioria cupinzeiro na cena;
- ❑ Outro: classe que representa qualquer objeto na cena que não possa ser representado por uma classe específica.

Na ?? abaixo pode-se obter qual foi o comando utilizado para atribuição de cada classe e sua respectiva cor.

Tabela 9 – Comando de classificação

Classe	Comando	Cor
Solo	S	Red
Formigueiro	A	Cyan
Sombra	E	White
Grama	1	Yellow
Cupim	P	Blue
Outros	R	Black

Fonte: Autor

Para a etapa de classificação manual, foram selecionadas 35 imagens das quais tentou-se variar ao máximo os cenários para que a base de dados rotulada apresentasse uma boa variação de características.

Foram classificados, no mínimo, 25% dos *Super Pixels* de cada imagem e cada uma das 35 imagens foi segmentada em 1700 *Super Pixels*, sendo assim, foram classificados, no mínimo, 14875 *Super Pixels* no total das 35 imagens.

Como resultado, o Demeter gera um arquivo *.csv* para cada imagem classificada. Cada linha desse arquivo corresponde a um *Super Pixel* e as colunas correspondem, respectivamente a: Classificação, Bloco, Posição\_X, Posição\_Y, Proporção\_Img, Data, Hora, Espécie e Usuário.

### 4.3.2 Tratamento dos dados

O tratamento dos dados foi realizado por meio do desenvolvimento de um código no *Jupyter Notebook*. O tratamento aplicado consistia em manipulações do *data frame* original, sendo elas:

- ❑ Concatenação dos arquivos *.csv* gerados para cada imagem

Cada imagem classificada teve como resultado um arquivo *.csv* correspondente, entretanto para a obtenção dos *insights* gerais, foi necessária a junção de todos esses arquivos *.csv* em um único arquivo;

- ❑ Criação de códigos de identificação (IDs)

Após a concatenação de todos os arquivos *.csv* em um único arquivo foram adicionadas duas novas colunas. Uma coluna chamada "Arquivo" que retornava o nome do arquivo *.csv* correspondente a imagem classificada e uma coluna chamada "ID" que retornava uma combinação da coluna "Arquivo" com a coluna "Bloco". A coluna "Bloco" corresponde ao *Super Pixel* classificado.

A criação de um código ID foi uma medida tomada para que no futuro fosse mais fácil realizar o rastreamento de qual imagem saiu aquela informação e também a possível necessidade de se agrupar mais informações correspondentes a cada *super pixel* de cada imagem proveniente de outros arquivos *.csv*.

- ❑ Exclusão de colunas

Para fins de *insights* gerais, algumas colunas não seriam utilizadas nas análises, sendo assim, as colunas que não eram interessantes foram excluídas. Sendo elas: "Posição\_X", "Posição\_Y", "Proporção\_Img", "Data", "Hora", "Espécie" e "Usuário".

Após esses tratamentos foi inserida uma linha de código que salva o *data frame* manipulado em um novo arquivo *.csv* com as alterações realizadas até o momento, entretanto para a análise gráfica relacionada ao balanceamento dos dados, mais algumas manipulações foram realizadas.

### 4.3.3 Análise gráfica

Para a geração do gráfico relacionado ao balanceamento dos dados rotulados, uma etapa adicional de tratamento dos dados foi realizada. Nela foi criado um novo *data frame* que continha apenas a transformação do comando correspondente a cada classe no nome da classe e um agrupamento das classificações, que resultou em uma nova coluna chamada "QTD" que retornava a quantidade de *Super Pixels* classificados de cada classe. Esse *data frame* foi salvo em um novo arquivo *.csv* e utilizado na geração do gráfico.

Após o tratamento final dos dados, foi realizada, por meio do *software Microsoft Power BI*, uma etapa de verificação/visualização do balanceamento dos dados classificados manualmente .

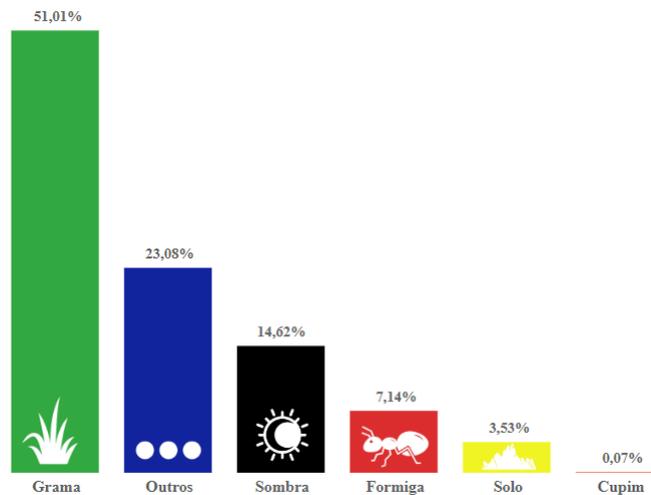
O *Power BI* é um *software* de *Business Intelligence* utilizado para análise de dados. O *software Microsoft Power BI* originou-se de vários recursos avançados do *Microsoft Excel*, ou seja, ele é um compilado de ferramentas. As ferramentas que compõem o *Power BI*

são: *Power Query* (para importar e modelar dados), *Power Pivot* (para modelar dados) e *Power View* (para criar visualizações interativas). Essas ferramentas foram organizadas na plataforma *Power BI* e disponibilizadas publicamente em 2015.

Dentre os principais recursos que o *Power BI* oferece, os que foram mais utilizados/aplicados nessa pesquisa foram as Visualizações e análises avançadas que o *Microsoft Power BI* permite realizar de forma mais simplificada, se comparado ao *Microsoft Excel*.

Abaixo pode-se observar o gráfico gerado utilizando-se o *Microsoft Power BI*:

Figura 40 – Balanceamento da base de dados classificada manualmente



Fonte: Autor.

Baseado na interpretação gráfica, é possível concluir que a base de dados classificadas manualmente é altamente desbalanceada. Saber sobre o desbalanceamento da base de dados implica diretamente na escolha do Método de Aprendizagem de Máquinas que será empregado.

O desbalanceamento dos dados corrobora a argumentação do uso de *Random Forest* como modelo de classificação, dado que, das opções estudadas/analísadas, ele apresenta bom desempenho para trabalhar com bases desbalanceadas.

## 4.4 Implementação dos Descritores

Como mencionado no referencial teórico, os descritores são variáveis ou atributos utilizados na descrição de um padrão. Quando se cria um modelo de aprendizado de máquinas, que tem como finalidade realizar classificações, busca-se encontrar padrões que servirão de base para a classificação. Sendo assim, necessita-se implementar descritores/*features* que são relevantes na busca de um padrão para a classificação dos formigueiros.

## 4.4.1 Seleção dos descritores

### 4.4.1.1 Descritores estatísticos

Dos descritores estatísticos, foi implementado o pacote de descritores denominado "*full\_statistics*". Esse pacote engloba os métodos estatísticos abaixo:

- Média Aritmética (Mean);
- Mediana (Med);
- Desvio Padrão (Std);
- Entropia (Entropy);
- Assimetria (Skew);
- Curtose (Kurtosis);

O funcionamento e cálculo de todos esses descritores pode ser encontrado na Seção subseção 2.6.3.

### 4.4.1.2 Espaços de cor

Na revisão bibliográfica foram estudados diversos espaços de cores. Esse estudo serviu de base para escolha de quais espaços de cores seriam utilizados como *features*. Os espaços de cores escolhidos foram:

- LAB;
- HSV;
- RGB;

As conversões desses espaços de cores podem ser consultadas na Seção 2.5.3.1.

### 4.4.1.3 Índices de cores/vegetativos

São funções que descrevem combinações entre os canais de cores afim de acentuar ou atenuar uma região/característica desejada. Baseado na revisão da literatura realizada, optou por trabalhar com os índices de cores abaixo: escolhidos foram:

- ExG;
- ExR;

- CIVE;
- NDI;
- VEG;
- mA;
- BmA;
- VARI;

As equações matemáticas por trás desses índices de cores podem ser consultadas na Seção subseção 2.5.3.2

#### 4.4.1.4 Descritores de textura

- canny;
- dft;

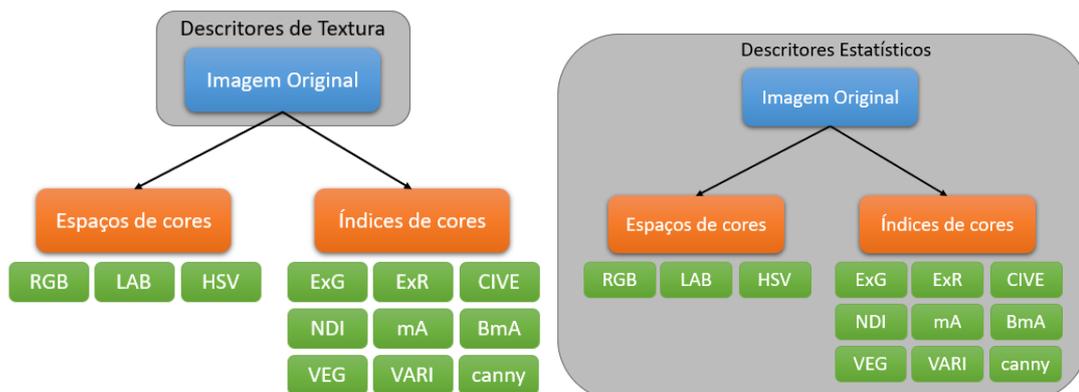
O funcionamento por trás desses descritores de textura podem ser consultados na Seção subseção 2.5.3.3.

Após a seleção dos descritores, foi desenvolvido um código em *Python* que faz a aplicação de cada um desses descritores para cada *Super Pixel* das imagens selecionadas.

#### 4.4.2 Aplicação dos descritores

A Figura 41 esquematiza a aplicação dos descritores para uma imagem.

Figura 41 – Aplicação dos descritores



(a) Aplicação dos descritores de textura

(b) Aplicação dos descritores estatísticos

Fonte: Autor.

Os descritores de textura foram aplicados somente a imagem original, já os descritores estatísticos são aplicados a imagem original, em cada um dos espaços de cores e em cada índice de cor. Lembrando que, quando se refere a aplicação dos descritores, eles são aplicados em cada *Super Pixel* de cada uma das imagens selecionadas.

Uma observação importante a ser feita é que nessa etapa de treinamento, os descritores só são calculados para os *Super Pixels* que receberam uma classificação manual por meio do *software* Demeter.

Isso foi feito pois o algoritmo escolhido é supervisionado e, o cálculo para todos os *Super Pixels* é onerosa e não contribui para o treinamento.

Então antes de calcular os descritores, o código carrega cada arquivo *.cvs* correspondente a cada uma das imagens classificadas manualmente e executa os descritores somente para aqueles *Super Pixels* que receberam uma classificação manual.

O resultado dessa implementação de descritores é um arquivo *.cvs* que contém, além do cálculo dos descritores aplicados conforme descrito na Figura 41, as seguintes informações:

- *ID: Index*;
- *image\_name*: Nome da imagem;
- *#\_SP*: Número do *Super Pixel*;
- *item*: Nome da imagem + número do *Super Pixel*;
- *label\_date* e *label\_time*: Data e hora da classificação manual;
- *pos\_x* e *pos\_y*: Posição *x* e *y* do *Super Pixel* na imagem;
- *GPS\_alt*, *GPS\_lat*, *GPS\_long*: Informações das coordenadas GPS retiradas das imagens;
- *label*: Classe atribuída (Classificação manual);
- *date* e *time*: Data e hora que o arquivo foi gerado;

Com a base de dados pronta, utilizou-se o *Jupyter Notebook* para o desenvolvimento dos modelos de classificação.

## 4.5 Modelo de classificação

Como visto anteriormente, a técnica de classificação escolhida foi *Random Forest*. As bibliotecas e funções utilizadas para a modelagem da *random forest* foram todas baseadas no *scikit-learn*.

Como parte importante da etapa de desenvolvimento do código, é necessária a criação de diversas versões com testes experimentais e melhorias realizadas nos códigos. Afim de registrar todas essas etapas, e não somente sobrescrever as melhorias, utilizou-se a técnica de versionamento. Esses versionamentos são provenientes de testes iterativos com o intuito de melhorar a visualização dos resultados, melhorar o desempenho do modelo e calcular mais métricas de análise de performance.

Apesar das versões serem iterativas, todas as versões contam com seis etapas bases, e são elas: Carregamento da base de dados, tratamento da base, divisão entre treino e teste, parametrização da *random forest*, geração de uma matriz de confusão e análise de performance. Sendo assim, essas seis etapas serão descritas para cada versão e ao final será gerada uma tabela de comparação entre as versões.

### 4.5.1 Versão zero (*v0*)

Essa primeira versão tem como intuito rodar uma *Random Forest* apenas para ter um direcionamento de como o modelo se comporta. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Carregamento da base de dados: A base de dados carregada é o arquivo *.csv* resultante da aplicação dos descritores, então ele contém toda a informação citada na subseção 4.4.2;
- ❑ Tratamento da base: Foram feitos alguns tratamentos na base de dados, e são eles, retirada das colunas que não são descritores e nem *label/classificação* e a conversão das variáveis da coluna *labels* para a classe correspondente. Não houve tratamento quanto a desbalanceamento da base;
- ❑ Divisão entre treino e teste: Os dados foram divididos entre 70% para treino e 30% para teste;
- ❑ Parametrização da *random forest*: Os parametros da *random forest* foram, 10 árvores com 4 de profundidade;
- ❑ Geração de uma matriz de confusão: Gerou-se uma matriz de confusão simples (apenas valores absolutos) com o intuito de comparar a classificação predita pela modelo da classificação real;
- ❑ Análise de performance: Utilizou-se apenas acurácia como métrica de análise de performance do modelo.

A etapa de carregamento da base de dados e tratamento da base só serão descritas quando houver alguma alteração nas mesmas, caso a configuração se mantenha igual a da versão zero ela não será descrita.

### 4.5.2 Versão um (v1)

Essa versão já foi desenvolvida pensando em incorporar alguns *insight* obtidos da v0. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Carregamento da base de dados;
- ❑ Tratamento da base: Foram utilizadas apenas duas classes. As seis classes foram transformadas em duas, "Formiga" continuou como formiga e todas as de mais classes foram agrupadas na classe denominada "Outros";
- ❑ Divisão entre treino e teste: Os dados foram divididos entre 70% para treino e 30% para teste;
- ❑ Parametrização da *random forest*: Os parametros da *random forest* foram, 10 árvores com 4 de profundidade;
- ❑ Geração de uma matriz de confusão: Gerou-se uma matriz de confusão simples (apenas valores absolutos) com o intuito de comparar a classificação predita pela modelo da classificação real;
- ❑ Análise de performance: Utilizou-se apenas acurácia como métrica de análise de performance do modelo.

Foi adicionado um novo passo a partir dessa versão e foi ele a "Geração gráfica das árvores utilizadas pelo modelo na classificação". Sua finalidade é tentar compreender quais são os descritores de maior ocorrência nas árvores. Acredita-se que a incorporação desse passo é importante para entender qual o caminho que o modelo está seguindo para fazer as classificações.

### 4.5.3 Versão dois (v2)

Essa versão já foi desenvolvida pensando em criar um balanceamento dos dados. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Carregamento da base de dados;
- ❑ Tratamento da base: Foi utilizada das seis classes. Entretanto, a fim de balancear as classes, foram escolhidos aleatoriamente 12 *Super Pixels* de cada classe (número referente a quantidade máxima da classe com menos SP classificado, que nesse caso é a classe de "Cupim");
- ❑ Divisão entre treino e teste: Os dados foram divididos entre 70% para treino e 30% para teste;

- ❑ Parametrização da *random forest*: Os parâmetros da *random forest* foram, 10 árvores com 4 de profundidade;
- ❑ Geração de uma matriz de confusão: Gerou-se uma matriz de confusão simples (apenas valores absolutos) com o intuito de comparar a classificação predita pelo modelo da classificação real;
- ❑ Análise de performance: Utilizou-se apenas acurácia como métrica de análise de performance do modelo.
- ❑ Geração gráfica das árvores;

#### 4.5.4 Versão três (v3.0)

Essa versão já foi desenvolvida pensando em incorporar alguns *insight* obtidos da v2. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Carregamento da base de dados;
- ❑ Tratamento da base: Foi utilizada apenas duas classes, "Formiga" e "Outros" (da mesma maneira que na v.1. E como forma de balanceamento, foram escolhidos aleatoriamente 1300 *Super Pixels* de cada classe (número referente a quantidade máxima da classe com menos SP classificado, que nesse caso é a classe de "Formiga");
- ❑ Divisão entre treino e teste: Os dados foram divididos entre 70% para treino e 30% para teste;
- ❑ Parametrização da *random forest*: Os parâmetros da *random forest* foram, 10 árvores com 4 de profundidade;
- ❑ Geração de uma matriz de confusão: Gerou-se uma matriz de confusão simples (apenas valores absolutos) com o intuito de comparar a classificação predita pelo modelo da classificação real;
- ❑ Análise de performance: Utilizou-se apenas acurácia como métrica de análise de performance do modelo.
- ❑ Geração gráfica das árvores;

#### 4.5.5 Versão três ponto um (v3.1)

Essa versão já foi desenvolvida com pequenos ajustes estéticos se comparado as versões anteriores. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Geração de duas matriz de confusão: Gerou-se duas matriz de confusão que além de serem mais agradáveis esteticamente, contém a informação em valores absolutos e em porcentagem;

Esses ajustes estéticos foram incorporados nas versões seguintes e além deles, também foi adicionado o cálculo do Desvio Padrão como métrica de avaliação de performance do modelo.

Observando-se as árvores utilizadas pelo modelo gerou a necessidade de realizar uma análise dos histogramas dos três canais de cores (RGB, HSV e LAB) para todas as seis classes.

Na geração dos histogramas, as intensidades de cada canal de cor foram divididas em 8 *bins* e os gráficos podem ser visualizados na seção de Resultados.

Com os resultados obtidos dessa análise, concluiu-se que não era necessária a utilização do espaço de cor LAB como descritor. As análises que levaram a tal decisão podem ser encontradas na seção de Resultados.

#### 4.5.6 Versão quatro (v4.0)

Essa versão já foi desenvolvida com alterações realizadas diretamente na base de dados, além do acréscimo de uma técnica de validação da divisão dos dados e mais métricas de análise de performance também foram calculadas. O passo a passo dessa versão pode ser encontrado abaixo:

- ❑ Carregamento da base de dados: A base de dados agora contém menos descritores. Foram retirados todos os descritores que envolvem o espaço de cor LAB, isso inclui os descritores estatísticos calculados em cima de cada canal do espaço e também os índices de cores que tinham os canais LAB como base. Foi retirado também os descritores de textura, esses descritores necessitam de um esforço computacional maior para serem calculados, então optou-se por retirar-los afim de analisar o custo benefício da sua implementação;
- ❑ Tratamento da base: O tratamento da base foi o mesmo realizado na v3;
- ❑ Divisão entre treino e teste: Os dados foram divididos entre treino e teste de acordo com a validação cruzada com 5 *fold slip*;
- ❑ Parametrização da *random forest*: Os parametros da *random forest* foram, 10 árvores com 4 de profundidade;
- ❑ Geração de duas matriz de confusão: Uma em valores absolutos e uma em porcentagem;
- ❑ Análise de performance: Utilizou-se quatro métricas de análise de performance, e foram elas, Acurácia, Precisão, *Recall* e *F1-Score*, além do Desvio Padrão.

- Geração gráfica das árvores;

#### 4.5.6.1 *Cross-Validation*

*Cross-validation* é uma técnica estatística utilizada para avaliar o desempenho e a capacidade de generalização de um modelo de *machine learning*. Ela é especialmente útil quando há uma quantidade limitada de dados disponíveis para treinar e testar o modelo (BERRAR, 2018) (JAMES et al., 2021).

A ideia básica por trás do *cross-validation* é dividir o conjunto de dados em duas ou mais partes: uma parte é usada para treinar o modelo e a outra parte é usada para avaliar o desempenho do modelo. O processo é repetido várias vezes, com diferentes combinações de treinamento e teste, e os resultados são combinados para obter uma estimativa mais robusta do desempenho do modelo (BERRAR, 2018) (JAMES et al., 2021).

A forma mais comum de *cross-validation* é a chamada validação cruzada *k-fold*. Nesse método, o conjunto de dados é dividido em  $k$  partes. O modelo é treinado  $k$  vezes, cada vez usando  $k - 1$  partes como dados de treinamento e a parte restante como dados de teste. O desempenho do modelo é então avaliado pela média dos resultados obtidos em cada uma das  $k$  iterações (BERRAR, 2018) (JAMES et al., 2021).

O *cross-validation* é uma técnica importante para avaliar o desempenho do modelo de aprendizado de máquina de forma mais confiável, ajudando a detectar problemas de *overfitting* ou *underfitting* do modelo aos dados de treinamento.

#### 4.5.7 Versão quatro ponto um (v4.1)

Essa versão já foi desenvolvida pensando em incorporar alguns *insight* obtidos da v4.0. O passo a passo dessa versão pode ser encontrado abaixo:

- Carregamento da base de dados: Foram retirados todos os descritores que envolvem o espaço de cor LAB, isso inclui os descritores estatísticos calculados em cima de cada canal do espaço e também os índices de cores que tinham os canais LAB como base. Foi retirado também os descritores de textura. Entretanto, acrescentou-se a informação relacionada aos histogramas de cada canal para cada classe;
- Tratamento da base: O tratamento da base foi o mesmo realizado na v3;
- Divisão entre treino e teste: Os dados foram divididos entre treino e teste de acordo com a validação cruzada com 5 *fold slip*;
- Parametrização da *random forest*: Os parametros da *random forest* foram, 10 árvores com 4 de profundidade;
- Geração de duas matriz de confusão: Uma em valores absolutos e uma em porcentagem;

- ❑ Análise de performance: Utilizou-se quatro métricas de análise de performance, e foram elas, Acurácia, Precisão, *Recall* e *F1-Score*.
- ❑ Geração gráfica das árvores;

Tabela 10 – Resumo das versões dos códigos de *Random Forest*

Versão	Base de dados	Quantidade de classes	Balanceamento dos dados	Divisão teste/treino	Parametros <i>Random Forest</i>	Matriz de Confusão
v0	Completa	6 (todas)	Sem balanceamento	70/30	10 árvores, 4 profundidade	Valor absoluto
v1	Completa	2 (formiga/outros)	Sem balanceamento	70/30	10 árvores, 4 profundidade	Valor absoluto
v2	Completa	6 (todas)	12 SP de cada classe	70/30	10 árvores, 4 profundidade	Valor absoluto
v3.0	Completa	2 (formiga/outros)	1300 SP de cada classe	70/30	10 árvores, 4 profundidade	Valor absoluto
v3.1	Completa	2 (formiga/outros)	1300 SP de cada classe	70/30	10 árvores, 4 profundidade	Valor absoluto + Porcentagem
v4.0	Sem LAB, sem textura e sem histograma	2 (formiga/outros)	1300 SP de cada classe	70/30 + <i>Cross Validation (5 fold split)</i>	10 árvores, 4 profundidade	Valor absoluto + Porcentagem
v4.1	Sem LAB, sem textura e com histograma	2 (formiga/outros)	1300 SP de cada classe	70/30 + <i>Cross Validation (5 fold split)</i>	10 árvores, 4 profundidade	Valor absoluto + Porcentagem

Fonte: Autor

---

## Resultados

Os resultados serão divididos de acordo com as versões dos códigos apresentadas na metodologia e depois uma seção para as considerações gerais.

### 5.1 Versão zero (*v0*)

Sendo esta uma versão de teste e de exploração, optou-se por manter todas as 6 classes, toda a base de dados da maneira que ela foi classificada e com todos os descritores estabelecidos na seção 4.4.

A divisão dos dados entre porcentagem utilizada para treino e para teste foi variada entre 60/40, 70/30 e 80/20. A divisão que apresentou melhor resultado foi a 70/30 então ficou estabelecido que essa seria a divisão a ser utilizada nas versões seguintes.

Os parâmetros da *random forest* também foram variados, tanto a quantidade de árvores quanto a profundidade das árvores. A combinação de parâmetros que melhor performou foi de 10 árvores com 4 de profundidade.

A matriz de confusão resultante da classificação realizada pelo modelo pode ser observada na Tabela 11.

O valor de acurácia obtido por meio dessa versão foi de aproximadamente 0.57 ou 57%.

### 5.2 Versão um (*v1*)

Nesta versão foi realizada uma das alterações significativas com relação a quantidade de classes utilizadas. O que na *v0* foi utilizada as seis classes, na *v1* foi utilizada apenas duas classes. A classe "Formiga" foi mantida como formiga e todas as outras cinco classes receberam o *label* de "Outros". Nenhum balanceamento foi proposto.

A matriz de confusão resultante da classificação realizada pelo modelo pode ser observada na tabela 12:

Tabela 11 – Matriz de confusão  $v0$ 

Predito	Gramma	Outros	Solo	Sombra	All
<b>Real</b>					
Cupim	2	3	0	0	5
Formiga	366	31	6	0	403
Gramma	2668	65	9	58	2800
Outros	928	360	1	8	1297
Solo	175	7	13	0	195
Sombra	643	29	8	88	768
All	4782	495	37	154	5468

Fonte: Autor

Tabela 12 – Matriz de confusão  $v1$ 

Predito	Formiga	Outros	All
<b>Real</b>			
Formiga	2	401	403
Outros	1	5064	5065
All	3	5465	5468

Fonte: Autor

O valor de acurácia obtido por meio dessa versão foi de aproximadamente 0.93 ou 93%.

Acredita-se que alto desempenho apresentado pelo modelo é devido ao desbalanceamento da base de dados, considerando que a base tem cerca de apenas 1300 *Super Pixels* classificados como formiga contra quase 17000 *Super Pixels* que foram atribuídos a "Outros". Sendo assim se o modelo classificar tudo como "Outros", já resultaria em uma acurácia de 92.6%.

### 5.3 Versão dois ( $v2$ )

Para esta versão, voltou-se a utilizar todas as seis classes, entretanto foi proposto uma forma de balanceamento. Considerando que a classe que tem menos *Super Pixels* classificados é a de "Cupim" com 12 SP, foi propôs um balanceamento aleatório de 12 SP de cada classe.

A matriz de confusão resultante da classificação realizada pelo modelo pode ser observada na Tabela 13.

Tabela 13 – Matriz de confusão v2

Predito	Cupim	Formiga	Gramma	Outros	Solo	Sombra	All
Real							
Cupim	1	1	0	2	0	1	5
Formiga	0	1	1	1	0	1	4
Gramma	0	0	1	1	1	1	4
Outros	3	0	0	0	0	0	3
Solo	2	1	0	0	0	2	5
Sombra	0	0	0	0	0	1	1
All	6	3	2	4	1	6	22

Fonte: Autor

O valor de acurácia obtido por meio dessa versão foi de aproximadamente 0.18 ou 18%.

Acredita-se que a baixa acurácia dessa versão é devido a pouquíssima quantidade de dados disponível para treinar o modelo.

## 5.4 Versão três ponto zero (v3.0)

Para a v3 optou-se por testar o que seria uma mistura da v1 com a v2. Realizou-se um agrupamento das classes, resultando em apenas duas classes ("Formiga" e "Outros"), assim como na v1, combinado com um balanceamento da quantidade de *Super Pixels* de cada classe. Sendo assim, foi selecionado 1300 *Super Pixels* aleatórios de cada classe, sendo esse valor a quantidade máxima de *Super Pixels* classificados como "Formiga".

A matriz de confusão resultante da classificação realizada pelo modelo pode ser observada na Tabela 14.

Tabela 14 – Matriz de confusão v3.0

Predito	Formiga	Outros	All
Real			
Formiga	312	67	379
Outros	145	256	401
All	457	323	780

Fonte: Autor

O valor de acurácia obtido por meio dessa versão foi de aproximadamente 0.73 ou 73%.

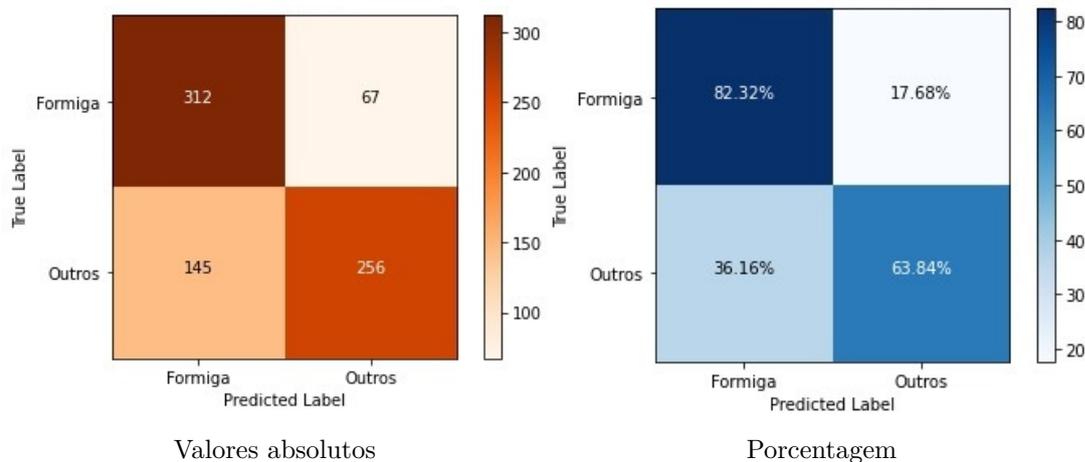
Comparando com as versões anteriores, esse foi o melhor resultado obtido. Então ficou estabelecido que para as próximas versões seriam utilizadas apenas duas classes ("Formiga" e "Outros") e que haveria um balanceamento aleatório de 1300 SP de cada classe.

## 5.5 Versão três ponto um (*v3.1*)

Nesta versão nenhuma alteração foi realizada na base de dados, nem na divisão das classes e nem no balanceamento dos dados. Portanto, o resultado apresentado é o mesmo da *v3.0*. Entretanto, ela foi uma versão criada para realização de ajustes estéticos, criação de uma segunda matriz de confusão com valores em porcentagem e criação de uma representação visual das árvores criadas pelo modelo.

As matrizes de confusão resultante da classificação realizada pelo modelo podem ser observadas na Tabela 15:

Tabela 15 – Matriz de confusão *v3.1*



Fonte: Autor.

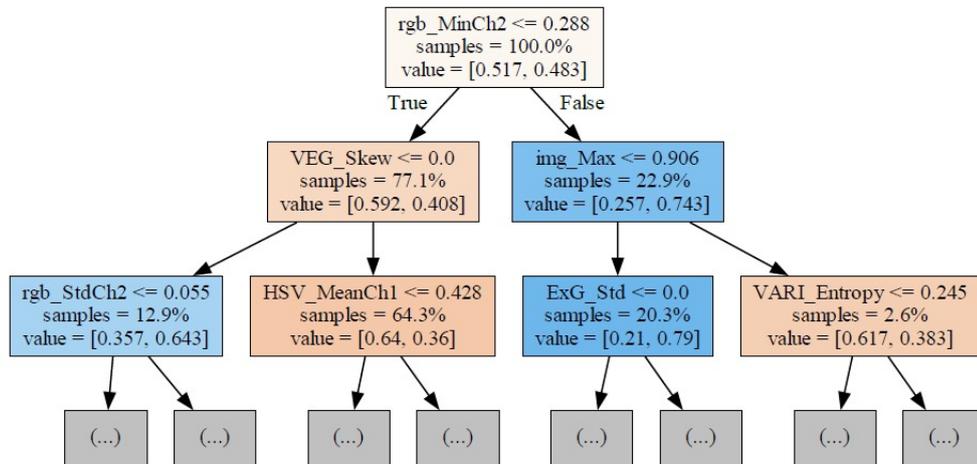
A Figura 42 é um exemplo de representação gráfica de uma das árvores geradas pelo modelo.

Optou-se por visualizar apenas duas camadas da árvore pois com as quatro camadas a imagem fica muito grande e os textos dentro dos *box* ficam muito pequenos e ilegíveis, inviabilizando a análise pretendida.

## 5.6 Histogramas dos canais de cores

Observando as imagens das árvores geradas a partir da versão *v3.1* foi possível notar que os descritores relacionados ao espaço de cor LAB quase não apareciam nos nós das árvores. Com isso foi desenvolvido um código que gerasse informação suficiente para

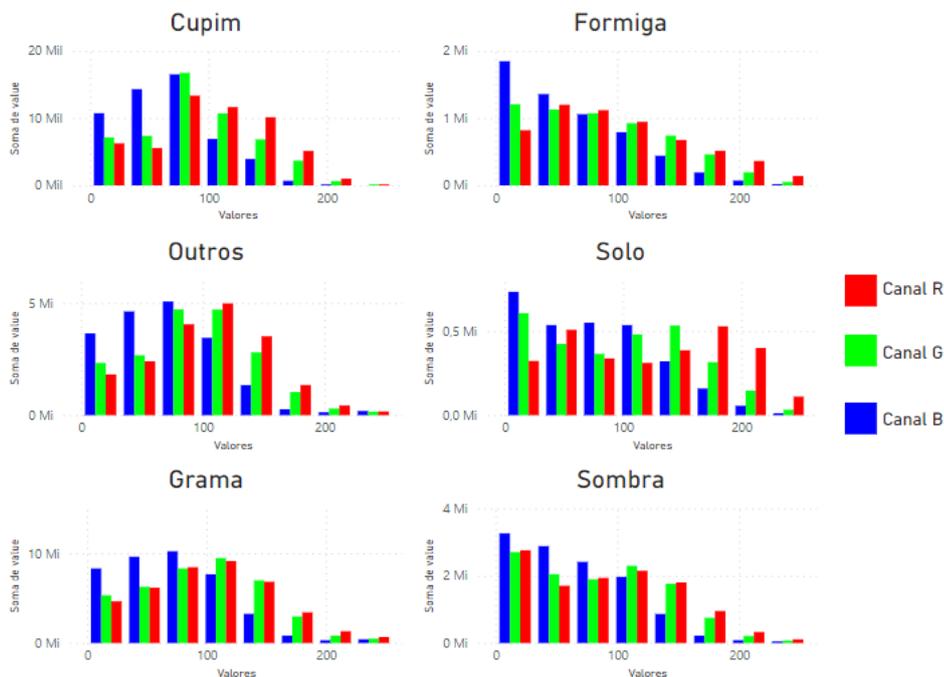
Figura 42 – Exemplo de árvore gerada pelo modelo



Fonte: Autor.

criação de histogramas dos três espaços de cores estudados (RGB, HSV e LAB) afim de analisar se há ou não um padrão de distribuição de cores entre os canais. A visualização dos gráficos foi feita por meio do *Microsoft Power BI* e pode ser observada nas figuras 43, 44 e 45.

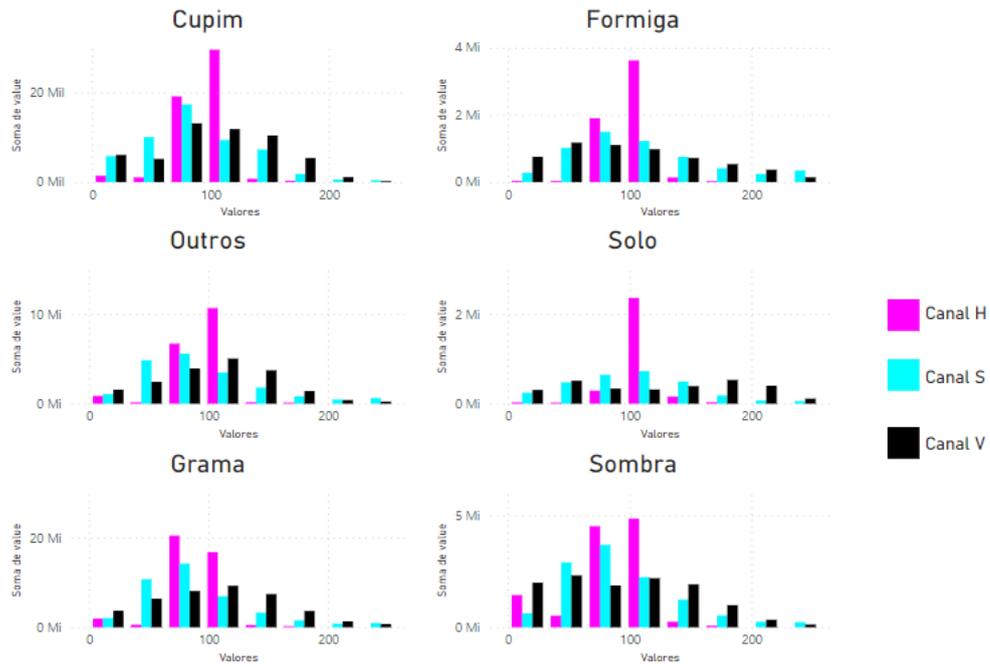
Figura 43 – Histograma espaço de cor RGB



Fonte: Autor.

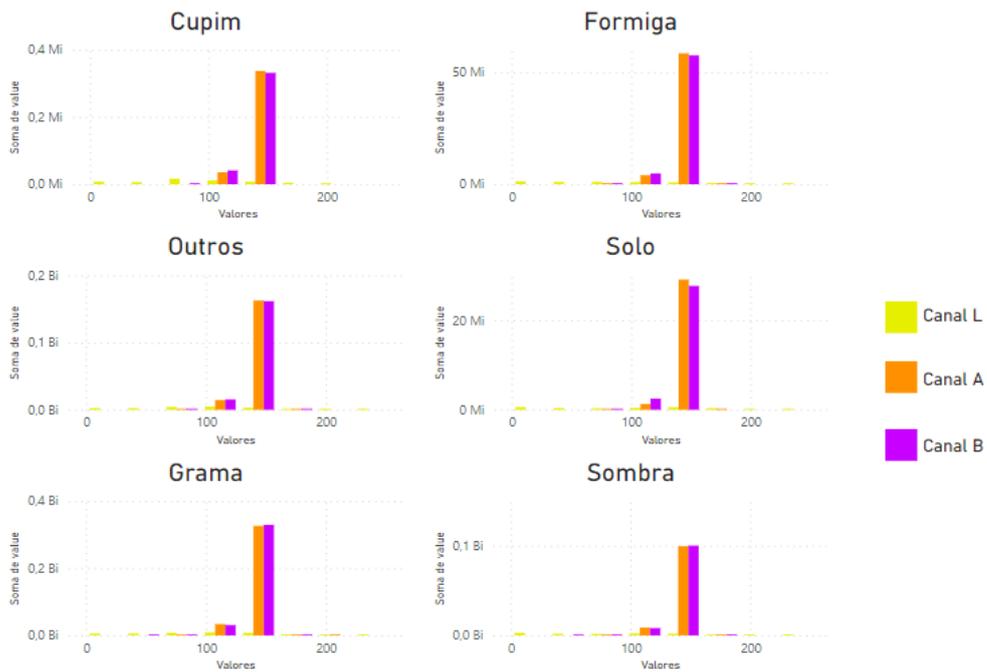
Analisando o histograma do espaço de LAB é possível notar que as classes não apresentam diferença significativa na distribuição de intensidade dos canais. Acredita-se que isso ocorre devido a própria técnica de segmentação de imagem, *SLIC*, já utilizar o espaço

Figura 44 – Histograma espaço de cor HSV



Fonte: Autor.

Figura 45 – Histograma espaço de cor LAB



Fonte: Autor.

de cor LAB para realizar a segmentação.

Sendo assim, ficou acordado que as versão seguintes não teriam mais o espaço de cor LAB, e nem os índices de cores que utilizam esse espaço, como descritores.

## 5.7 Versão quatro ponto zero (v4.0)

A versão quatro ponto zero tem três principais diferenças das versões anteriores.

Para essa versão uma nova base de dados foi gerada, agora a nova base não possui mais o espaço de cor LAB como descritor e nem os Índices de cores que utilizavam esse canal em sua composição/cálculo. Outra diferença na base de dados é optou-se por tirar também os descritores de textura. Os descritores de textura exigem um esforço computacional maior para serem calculados, então essa versão testa o quanto a presença desses descritores contribui para uma boa performance e se vale a pena mantê-los ou não.

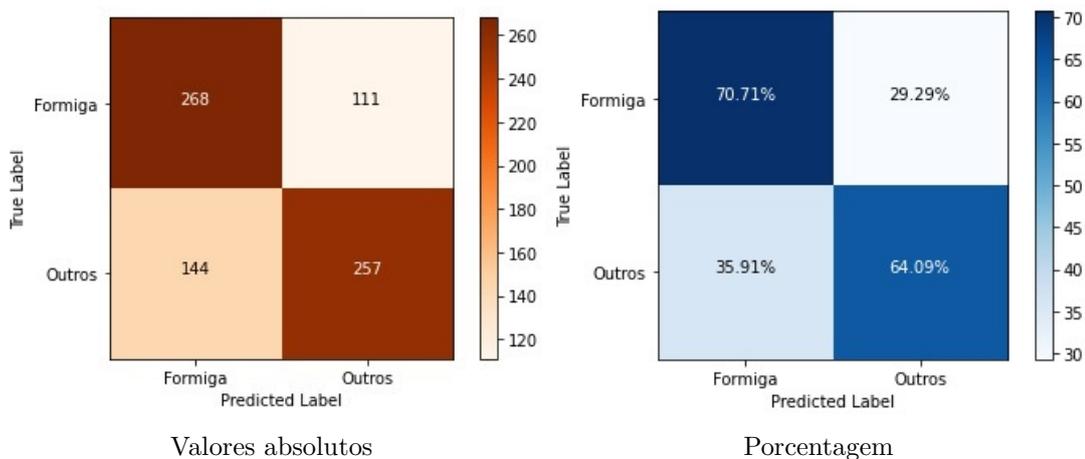
Outra grande diferença é que além da divisão 70/30 para treino e teste, também foi implementada uma validação cruzada dessa divisão *Cross Validation*. Essa validação cruzada contou com 5 *fold slip*.

A terceira e última grande diferença é que a partir dessa versão serão calculadas mais de uma métrica de avaliação de performance do modelo. As métricas escolhidas foram: Acurácia, Precisão, *Recall* e *F1-Score*

A visualização gráfica das árvores geradas pelo modelo foi mantida nesta versão e na versão seguinte.

As matrizes de confusão resultante da classificação realizada pelo modelo podem ser observadas na Tabela 16.

Tabela 16 – Matriz de confusão v4.0



Fonte: Autor.

Os valores das métricas obtidas por meio dessa versão foram de aproximadamente:

– Acurácia: 0.6731 ou 67.31%

- Precisão: 0.6758 ou 67.58%
- *Recall*: 0.6731 ou 67.31%
- *F1-Socre*: 0.6926 ou 69.26%

Todas as métricas desta versão apresentaram um desvio padrão de aproximadamente 0.04 ou 4%.

Acredita-se que a razão pela qual todas as métricas apresentaram valores próximos é que a base de dados utilizada nessa versão foi balanceada.

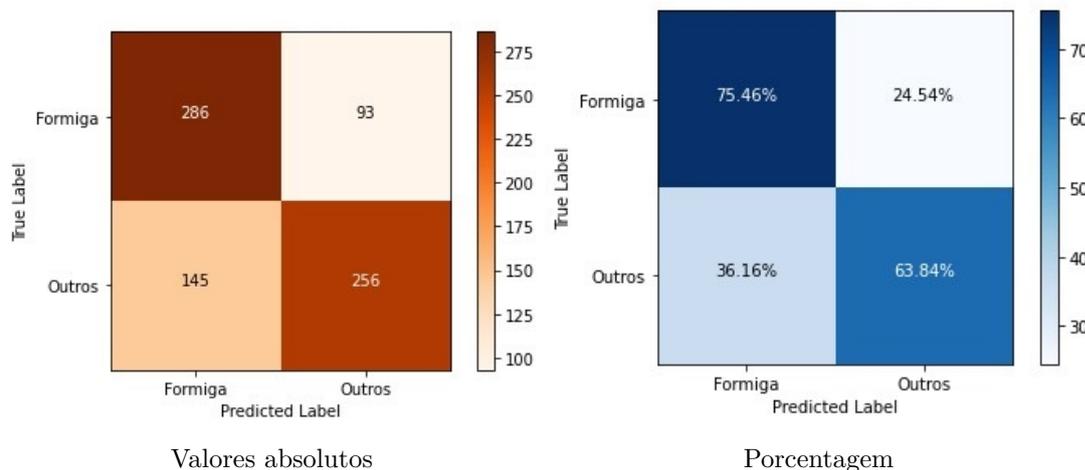
A retirada dos descritores de textura não apresentou piora significativa no desempenho do modelo. Para obter uma análise mais profunda sobre o quanto compensa utilizar esses descritores de textura, fica a sugestão de um trabalho futuro juntamente com um Engenheiro de dados.

## 5.8 Versão quatro ponto um (*v4.1*)

A diferença entre esta versão e a *v4.0* é que nesta versão optou-se por acrescentar as informações dos histogramas como descritores. Dessa forma, somente a base de dados sofreu alteração, as de mais etapas seguem iguais a *v4.0*

As matrizes de confusão resultante da classificação realizada pelo modelo podem ser observadas abaixo:

Tabela 17 – Matriz de confusão *v4.1*



Fonte: Autor.

Os valores das métricas obtidas por meio dessa versão foram de aproximadamente:

- Acurácia: 0.6949 ou 69.49%
- Previsão: 0.7014 ou 70.14%

- *Recall*: 0.6949 ou 69.49%
- *F1-Socre*: 0.6936 ou 69.36%

Todas as métricas desta versão apresentaram um desvio padrão de aproximadamente 0.02 ou 2%.

## 5.9 Considerações gerais

A Tabela 18 abaixo, podem ser observados os valores das métricas de performance obtidos em cada versão.

Tabela 18 – Resumo dos resultados

Versão	<i>F1-Score</i>	<i>Recall</i>	Precisão	Acurácia
<b>v0</b>	-	-	-	0.5722 (57.22%)
<b>v1</b>	-	-	-	0.9265 (92.65%)
<b>v2</b>	-	-	-	0.1818 (18.18%)
<b>v3.1</b>	-	-	-	0.7278 (72.78%)
<b>v4.0</b>	0.6726 (67.26%)	0.6731 (67.31%)	0.6757 (67.57%)	0.6731 (67.31%)
<b>v4.1</b>	0.6936 (69.36%)	0.6949 (69.49%)	0.7014 (70.14%)	0.6949 (69.49%)

Fonte: Autor

Ao observar os valores da acurácia, métrica principal e a única que utilizada em todas as versões, é possível notar que o modelo proposto apresentou, de maneira geral, melhora no desempenho classificatório a cada nova versão desenvolvida.

Um das exceções foi a versão um (*v1*), que pesar de ter apresentado a melhor acurácia, acredita-se que isso se deve ao desbalanceamento da base, fazendo com que, mesmo o modelo classificando tudo como "Outros" ainda resultaria em uma alta acurácia.

Outro ponto que chamou a atenção é que as versões três, (*v3.0*) e (*v3.1*), apresentaram um valor de acurácia maior do que as versões quatro, (*v4.0*) e (*v4.1*). Supõe-se que, a adição da técnica de *Cross-Validation*, nas versões quatro, minimizou um possível *overfitting* que pode ter ocorrido nas versões três. A minimização desse possível *overfitting* resultou em uma "piora" no desempenho do modelo, entretanto tornou o mesmo mais confiável.

Foi possível observar também que, realizar um balanceamento dos dados apresentou uma melhora significativa na acurácia entre os modelos. Lembrando que esse balanceamento precisa possuir uma quantidade razoável de *Super Pixel* de cada classe, dado que uma base de dados com poucas amostras não é suficientes para treinar bem o modelo.

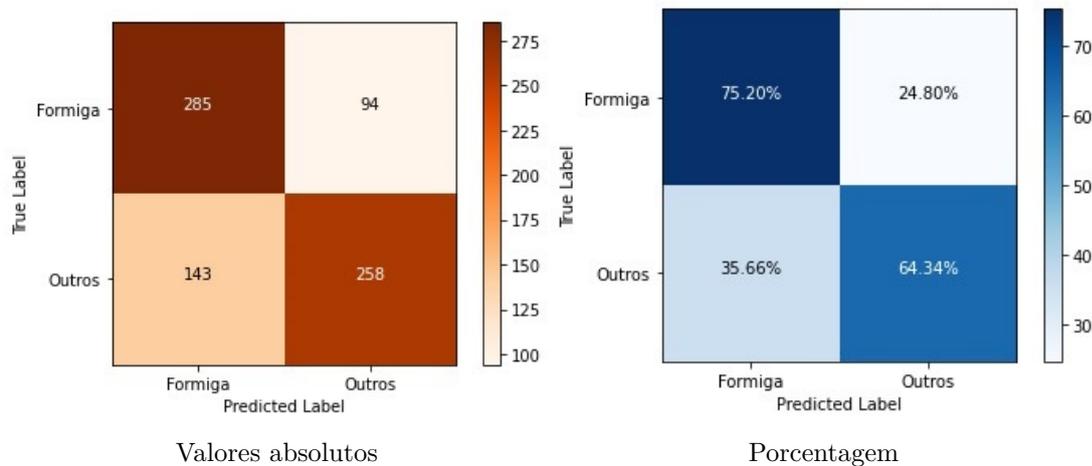
A princípio a retirada dos descritores que envolviam o espaço de cor LAB não apresentou piora significativa na acurácia. Entretanto, considerando que essa retirada aconteceu na mesma versão que a adição do *Cross-Validation*, e também a retirada dos descritores de textura, realizou-se um teste adicional para analisar o comportamento do modelo quando o espaço de cor LAB era mantido. Esse teste, que pode ser chamado de Versão Extra.

### 5.9.1 Versão extra (*v.extra*)

Afim de comprovar a hipótese de que retirada do espaço de cor LAB não afeta significativamente o modelo e que o que resultou em uma "piora" no valor da acurácia da *v3.1* um para a *v4.0* foi a combinação do acréscimo do *Cross-Validation* com a retirada dos descritores de textura, um versão extra foi desenvolvida.

As matrizes de confusão resultante da classificação realizada pelo modelo podem ser observadas abaixo:

Tabela 19 – Matriz de confusão versão extra



Fonte: Autor.

O valor de acurácia obtido por meio dessa versão foi de aproximadamente 0.70 ou 70%.

Com esse resultado é possível confirmar que a retirada do espaço de cor LAB não foi o responsável pela "piora" no valor da acurácia. Essa retirada afetou sim o desempenho do modelo, mas não foi significativo, considerando que o desvio padrão apresentado foi de 0.03 ou 3%.

Comparando a acurácia entre a *v3.1*, *v4.0* e *v.extra*, e o desvio padrão entre a *v4.0* e *v.extra* pode considerar que a retirada do espaço de cor LAB, a retirada dos descritores de textura e o acréscimo do *Cross-Validation* mantiveram a o valor da acurácia dentro do desvio padrão. Da mesma maneira pode-se dizer que o acréscimo do histograma como descritor também não foi significativo para o modelo.

Tabela 20 – Análise desvio padrão

<b>Versão</b>	<b>Base de dados</b>	<b>Divisão teste/treino</b>	<b>Acurácia</b>	<b>Desvio Padrão</b>
<b>v3.1</b>	Completa	70/30	0.7278 (72.78%)	-
<b>v.extra</b>	Completa	70/30 + <i>Cross Validation</i>	0.6962 (69.62%)	0.03 (3%)
<b>v4.0</b>	Sem LAB, sem textura e sem histograma	70/30 + <i>Cross Validation</i>	0.6731 (67.31%)	0.02 (2%)
<b>v4.1</b>	Sem LAB, sem textura e com histograma	70/30 + <i>Cross Validation</i>	0.6949 (69.49%)	0.04 (4%)

Fonte: Autor

E por fim, observando as matrizes de confusão é possível dizer que os modelos tem uma boa porcentagem de acerto quando classifica o *super pixel* como formigueiro. Na ultima versão, (v4.1), o modelo acertou 75.46% das vezes que classificou como formigueiro.



---

## Conclusão

Em suma, esta dissertação explorou a aplicação de visão computacional e quadricópteros na detecção de ninhos de formigas do gênero *Atta*.

É notável que o Brasil é um país com uma agricultura e agropecuária muito forte, sendo um importante exportador desses insumos. Com o passar dos anos aumentou-se a preocupação em otimizar a utilização das terras e as plantações, considerando que a produção aumentava a cada ano mas não se aumentava os espaços para cultivo. Assim surgiu a agricultura de precisão, que visa auxiliar os produtores na redução dos custos, melhora da qualidade e quantidade dos produtos e aumento da renda agrícola. Uma das ferramentas da agricultura de precisão é o sensoriamento remoto, que muitas vezes é aplicado utilizando-se de aeronaves não tripuladas.

Da análise bibliográfica realizada pode-se observar que apesar do crescente interesse na área de *Remote Sensing*, agricultura de precisão e uso de drone, pouco se pesquisa sobre suas aplicações à detecção de ninhos de formiga. Isso apontava que muitas dificuldades seriam enfrentadas, considerando que o estado da arte está no início de seu desenvolvimento.

Do levantamento do estado da arte, a pesquisa que mais se aproximava deste projeto ainda estava muito distante pois fazia uso de aeronave tripuladas (avião) e câmeras infravermelhas. Em posse dessa informação, o presente projeto teve como objetivo a utilização de drone comercial e câmeras RGB, o que viabiliza a possibilidade da transformação deste projeto em um produto a ser comercializado.

Algumas dificuldades foram enfrentadas, o que fez com o que a possibilidade de detecção ficasse cada mais restrita. Uma altura de voo de 10 metros foi estabelecida, a fim de que os formigueiros não aparecessem tão pequenos nas imagens, mas que a varredura da área também pudesse ser realizada de maneira eficiente. Um horário de voo e condição climática foram estabelecidos para evitar os efeitos da cromaticidade, e foram eles: um horário por volta de meio dia e em dias ensolarados. Uma espécie de formiga do gênero *Atta* também precisou ser estabelecida. Considerando a sua preferência de localização e morfologia externa do ninho a espécie escolhida foi a *Atta Capiguara*.

Com base nas campanhas experimentais realizadas ficou-se estabelecido também uma época do ano para a captura das imagens. As fortes chuvas que ocorrem no verão acabam alterando as características externas dos formigueiros, em alguns casos até extinguindo-o. Sendo assim a melhor época do ano para a detecção é o inverno, ou pelo após um período sem chuva.

Dos resultados obtidos, pode-se concluir que é possível a detecção de ninhos de formigas do gênero *Atta* utilizando visão computacional e quadrirrotores, considerando as limitações impostas. O modelo de *Random Forest* desenvolvido apresentou uma acurácia de aproximadamente 70%, o que se assemelha a acurácia obtida pela pesquisa mais semelhante a esta, que foi de aproximadamente 75% limitada ao mês de maio.

Considerando que a pesquisa desenvolvida por Vogt, (VOGT, 2004a), que além de fazer uso de aeronave tripulada, ainda utilizava câmera infravermelha, o resultado apresentado pelo modelo desenvolvido nesta pesquisa foi extremamente satisfatório.

Como trabalhos futuros sugere-se que sejam realizados testes utilizando outros tipos de modelos de classificação, a expansão da base de dados afim de contemplar cenários mais diversos, a realização de um *tuning* de parâmetros, uma *feature selection*, o possível desenvolvimento de um índice de cor.

Acredita-se que esta pesquisa contribui significativamente para o campo de sensoria-mento remoto e agricultura de precisão ao apresentar materiais e métodos mais acessíveis que conseguem performar de maneira semelhante ao uso de equipamentos mais sofisticados e técnicas mais robustas.

Em última análise, a aplicação de visão computacional e quadrirrotoers na detecção de ninho de formiga representa uma oportunidade empolgante para impulsionar a agricultura de precisão e controle de pragas no campo. No entanto, é fundamental abordar que ainda existem limitações quanto ao cenário e condições de aquisição das imagens.

---

## Referências

- ABERTURA, velocidade e ISO: Os 3 pilares da fotografia. 2018. Disponível em <<https://www.eduardo-monica.com/new-blog/iso-velocidade-abertura-exposicao-fotografia>>. Acesso em 13 setembro 2021.
- ACHANTA, R. et al. SLIC superpixels compared to state-of-the-art superpixel methods. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 34, n. 11, p. 2274–2282, nov. 2012.
- ALGORITMOS para detecção de bordas. 1998. Disponível em <<https://www.inf.ufsc.br/~aldo.vw/visao/1998/seara/index.html>>. Acesso em 03 agosto 2021.
- AMANTE, E. A formiga saúva atta capiguara, praga das pastagens. **O Biológico**, v. 33, n. 6, p. 113–120, 1967a.
- \_\_\_\_\_. Saúva tira boi das pastagem. **Coopercotia**, v. 23, n. 207, p. 38–40, 1967b.
- ARENAS, A.; ROCES, F. Avoidance of plants unsuitable for the symbiotic fungus in leaf-cutting ants: Learning can take place entirely at the colony dump. **PLoS One**, v. 12, n. 3, p. 1–16, mar. 2017.
- BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à Visão Computacional usando Matlab**. [S.l.: s.n.], 2019.
- BALAFOUTIS, A. et al. Precision agriculture technologies positively contributing to GHG emissions mitigation, farm productivity and economics. **Sustain. Sci. Pract. Policy**, Multidisciplinary Digital Publishing Institute, v. 9, n. 8, p. 1339, jul. 2017.
- BARBEDO, J. G. A. A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses. **Drones**, Multidisciplinary Digital Publishing Institute, v. 3, n. 2, p. 40, abr. 2019.
- BARSANTE, C. **Tipos de drones no mercado e suas aplicações**. 2020. Disponível em <<https://www.linkedin.com/pulse/tipos-de-drones-mercado-e-suas-aplica%C3%A7%C3%B5es-christiano-barsante/?originalSubdomain=pt>>. Acesso em 06 setembro 2021.
- BATISTA, G. de; FILHO, H. F.; LOUREIRO, e. P. E. de A. V. Controle da saúva-parda com isca de dodecacloro e seus resíduos em solo e pastagens. **Pesqui. Agropecu. Bras.**, v. 20, n. 3, p. 273–276, 1985.

- BELGIU, M.; DRĂGUȚ, L. Random forest in remote sensing: A review of applications and future directions. **ISPRS J. Photogramm. Remote Sens.**, v. 114, p. 24–31, abr. 2016.
- BERRAR, D. **Reference Module in Life Sciences**. [S.l.: s.n.], 2018.
- BIOTEC, O. **Drones agrícolas: função, tipos e vantagens**. 2022. Disponível em <<https://oligosbiotec.com.br/blog/drones-agricolas-funcao-tipos-e-vantagens/>>. Acesso em 20 maio 2023.
- BREIMAN, L. Bagging predictors. **Mach. Learn.**, v. 24, p. 123–140, 1996.
- \_\_\_\_\_. Random forests. **Mach. Learn.**, v. 45, p. 5–32, 2001.
- BURGOS-ARTIZZU, X. P. et al. Real-time image processing for crop/weed discrimination in maize fields. **Comput. Electron. Agric.**, v. 75, n. 2, p. 337–346, fev. 2011.
- CALDATO, N. et al. Foraging activity pattern and herbivory rates of the grass-cutting ant *atta capiguara*. **Insectes Soc.**, v. 63, n. 3, p. 421–428, ago. 2016.
- CAMBIO, M. **Cotação do Arroba do Boi**. 2023. Disponível em <<https://www.melhorcambio.com/boi-hoje>>. Acesso em 23 maio 2023.
- CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2016.
- CHROMAFLO. **Chromaflo Tecnologias**. 2021. Disponível em <<https://chromaflo.com/pt-br/teoria-das-cores/caracteristicas-da-cor/>>. Acesso em 03 outubro 2021.
- CIPRA, M.; JAN, B. **Identification of Agricultural Crops by Computer Processing of ERTS MSS Data**. [S.l.]: LARS Technical Reports, 1973. 20 p.
- CNA. **Panorama do Agro**. 2020. Disponível em <[https://www.cnabrazil.org.br/cna/panorama-do-agro#\\_ftn1](https://www.cnabrazil.org.br/cna/panorama-do-agro#_ftn1)>. Acesso em 03 outubro 2021.
- COOLEY, J. W.; PETER, A. W. L.; WELCH, P. D. Historical notes on the fast fourier transform. **IEEE Transactions on Audio and Electroacoustics**, v. 15, p. 76–79, jun. 1967.
- COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. **Math. Comput.**, American Mathematical Society (AMS), v. 19, n. 90, p. 297–297, maio 1965.
- COSTA, N. A. et al. Do herbivores exert top-down effects in neotropical savannas? estimates of biomass consumption by leaf-cutter ants. **J. Veg. Sci.**, v. 19, p. 849–854, 2008.
- CUNHA, L. S. da. **Assimetria e Curtose**. 2017. Disponível em <<http://www.uel.br/pessoal/lscunha/pages/arquivos/uel/Economia%20Noturno/Aula%206%20-%20Assimetria%20e%20Curtose.pdf>>. Acesso em 18 setembro 2021.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, p. 886–893, jul. 2005.

DIAS, L. V. A. **Estudo e análise de diferentes métodos de calibração de câmeras**. Dissertação (Mestrado), 2015.

DICIONÁRIO. **Dicionário Online de Português**. 2021. Disponível em <<https://www.dicio.com.br/cromaticidade/>>. Acesso em 03 outubro 2021.

DIETTERICH, T. G. EnsembleMethods in Machine Learning. **Plan. Perspect.**, v. 1, p. 15, 2000.

DILEEP, M. R. et al. A study and analysis on various types of agricultural drones and its applications. In: **2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)**. [S.l.: s.n.], 2020. p. 181–185.

EFRON, B. Bootstrap methods: Another look at the jackknife. **Ann. Stat.**, v. 7, p. 1–26, 1979.

ELETRÔNICOS Mercado Livre. 2021. Disponível em <[https://eletronicos.mercadolivre.com.br/drones-acessorios/camera-multiespectral\\_NoIndex\\_True](https://eletronicos.mercadolivre.com.br/drones-acessorios/camera-multiespectral_NoIndex_True)>. Acesso em 30 setembro 2021.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of database systems**. 7. ed. Upper Saddle River, NJ: Pearson, 2015.

EMBRAPA. **Visão 2030: O futuro da agricultura brasileira**. [S.l.: s.n.], 2018.

FARMFOR. **Drone pulverizador da chinesa Chufangagri leva até 18 kg de carga**. 2021. Disponível em <<https://www.farmfor.com.br/posts/drone-pulverizador-da-chinesa-chufangagri-leva-ate-18-kg-de-carga/>>. Acesso em 20 maio 2023.

FELTRAN-BARBIERI, R.; FÉRES, J. G. Degraded pastures in brazil: improving livestock production and forest restoration. **R Soc Open Sci**, v. 8, n. 7, p. 201854, jul. 2021.

FERREIRA, A. dos S. et al. Weed detection in soybean crops using ConvNets. **Comput. Electron. Agric.**, v. 143, p. 314–324, dez. 2017.

FILHO, O. M.; NETO, H. V. **Processamento Digital de Imagens**. [S.l.]: Brasport, 1999.

FIMI. **FIMI X8 SE**. 2021. Disponível em <<https://www.fimi.com/fimi-x8-se.html>>. Acesso em 01 setembro 2021.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: . [S.l.]: Machine Learning: Proceedings of the Thirteenth International Conference, 1996.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **Ann. Stat.**, v. 29, 2001.

GANESAN, P.; RAJINI, V.; RAJKUMAR, R. I. Segmentation and edge detection of color images using CIELAB color space and edge detectors. In: . [S.l.]: International Conference on “Emerging Trends in Robotics and Communication Technologies, 2010.

- GANESAN, P. et al. A comprehensive review of the impact of color space on image segmentation. In: . [S.l.: s.n.], 2019.
- GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing Using Matlab**. [S.l.]: Tom Robbins, 2001.
- \_\_\_\_\_. **Digital Image Processing**. 3a. ed. [S.l.]: Pearson International, 2008.
- GOUVEIA, R. **Desvio Padrão**. 2021. Disponível em <<https://www.todamateria.com.br/desvio-padrao/>>. Acesso em 18 setembro 2021.
- GUERRERO, J. M. et al. Support vector machines for crop/weeds identification in maize fields. **Expert Syst. Appl.**, v. 39, n. 12, p. 11149–11155, set. 2012.
- HAGUE, T.; TILLET, N. D.; WHEELER, H. Automated crop and weed monitoring in widely spaced cereals. **Precision Agriculture**, 2006.
- HAMAGUCHI, A. **Explore o mundo da fotografia manual com sua câmera DSLR**. 2013. Disponível em <<https://www.techtudo.com.br/noticias/2013/01/explore-o-mundo-da-fotografia-manual-com-sua-camera-dslr.ghtml>>. Acesso em 03 outubro 2021.
- HAMUDA, E.; GLAVIN, M.; JONES, E. A survey of image processing techniques for plant extraction and segmentation in the field. **Comput. Electron. Agric.**, v. 125, p. 184–199, jul. 2016.
- HARALICK, R. M. Statistical and structural approaches to texture. **Proc. IEEE**, v. 67, n. 5, p. 786–804, 1979.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. [S.l.]: Springer, 2009.
- HISTOGRAM of Oriented Gradients explained using OpenCV. 2016. Disponível em <<https://learnopencv.com/histogram-of-oriented-gradients/>>. Acesso em 01 agosto 2021.
- HOSSIN, M.; SULAIMAN, M. A review on evaluation metrics for data classification evaluations. **International Journal of Data Mining Knowledge Management Process**, v. 5, n. 2, 2015.
- HOW does shutter speed affect motion blur? 2019. Disponível em <<https://www.sony.com/electronics/support/articles/00031181>>. Acesso em 13 setembro 2021.
- HUNG, C.; XU, Z.; SUKKARIEH, S. Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a UAV. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 6, n. 12, p. 12037–12054, dez. 2014.
- ICA100-40. **Aeronaves não tripuladas e o acesso ao espaço aéreo brasileiro**. 2020. <<https://publicacoes.decea.mil.br/api//storage/uploads/files/75a09bfd-5e5d-4f9a-b4485ccd3fd4627a.pdf>>.
- IDCGLOBAL. 2021. Disponível em <<https://www.idcglobal.com/>>. Acesso em 08 outubro 2021.

- IZBICKI, R.; SANTOS, T. M. dos. **Machine Learning sob a ótica estatística**. Tese (Doutorado), 2019.
- JAFFE, K. **El mundo de las hormigas**. [S.l.]: NY: W. W. Norton & Company: New York, 2010.
- JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. [S.l.]: Springer, New York, NY, 2013.
- \_\_\_\_\_. **An Introduction to Statistical Learning with Applications in R**. [S.l.]: Springer, 2021.
- JULESZ, B. Visual pattern discrimination. **IRE Transactions on Information Theory**, v. 8, n. 2, p. 84–92, 1962.
- KATAOKA, T. et al. Crop growth estimation system using machine vision. In: . [S.l.: s.n.], 2003.
- KIM, A. Fast SLIC: Optimized SLIC superpixel.
- KITAHARA, J. R. **Quora**. 2021. Disponível em <<https://www.quora.com/Is-it-a-normal-probability-distribution-that-has-a-mean-%CE%BC-0-and-a-standard-deviation-%CF%83-1>>. Acesso em 03 outubro 2021.
- KOZACEK, B.; GRAUZEL, J.; FRIVALDSKY, M. The main capabilities and solutions for different types of the image sensors. **2018 ELEKTRO**, 2018.
- KRÜGER, F. **Activity, Context, and Plan Recognition with Computational Causal Behaviour Models**. Tese (Doutorado), 2016.
- KULBACKI, M. et al. Survey of drones for agriculture automation from planting to harvest. In: . [S.l.]: IEEE, 2018.
- LEE, W. S. et al. Sensing technologies for precision specialty crop production. **Comput. Electron. Agric.**, v. 74, n. 1, p. 2–33, out. 2010.
- LI, Y. et al. The recognition of rice images by UAV based on capsule network. **Cluster Comput.**, v. 22, n. 4, p. 9515–9524, jul. 2019.
- LOURES, L. et al. Assessing the effectiveness of precision agriculture management systems in mediterranean small farms. **Sustain. Sci. Pract. Policy**, Multidisciplinary Digital Publishing Institute, v. 12, n. 9, p. 3765, maio 2020.
- LUCIA, T. M. C. D. **Formigas Cortadeiras: da bioecologia ao manejo**. [S.l.]: Editora UFV, 2011.
- MAHESWARI, S.; KORAH, R. Review on image segmentation based on color space and its hybrid. In: . [S.l.: s.n.], 2016.
- MEHTA, S.; PATEL, A.; MEHTA, J. CCD or CMOS image sensor for photography. In: . [S.l.: s.n.], 2015.
- MEYER, G. E.; HINDMAN, T.; LAKSMI, K. Machine vision detection parameters for plant species identification. **Precision Agriculture and Biological Quality**, v. 3543, nov. 1998.

MEYER, G. E.; NETO, J. C. Verification of color vegetation indices for automated crop imaging applications. **Comput. Electron. Agric.**, v. 63, n. 2, p. 282–293, out. 2008.

MOGHADAM, P. A.; SHARMA, N.; HEFEEDA, M. Enabling hyperspectral imaging in diverse illumination conditions for indoor applications. In: **Proceedings of the 12th ACM Multimedia Systems Conference**. New York, NY, USA: Association for Computing Machinery, 2021. (MMSys '21), p. 23–35.

MOGILI, U. M. R.; DEEPAK, B. B. V. L. Review on application of drone systems in precision agriculture. **Procedia Comput. Sci.**, v. 133, p. 502–509, jan. 2018.

MORA, A. et al. Land cover classification from multispectral data using computational intelligence tools: A comparative study. **Information**, Multidisciplinary Digital Publishing Institute, v. 8, n. 4, p. 147, nov. 2017.

MORETTIN, P. A.; BUSSAB, W. de O. **Estatística Básica**. [S.l.]: Saraiva, 2013.

MOURHIR, A. et al. Exploring precision farming scenarios using fuzzy cognitive maps. **Sustain. Sci. Pract. Policy**, Multidisciplinary Digital Publishing Institute, v. 9, n. 7, p. 1241, jul. 2017.

MULLA, D. J. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. **Biosystems Eng.**, v. 114, n. 4, p. 358–371, abr. 2013.

MURARI, A. **Triângulo de exposição na fotográfica: Saiba como funciona**. 2020. Disponível em <<https://www.epics.com.br/blog/triangulo-de-exposicao-na-fotografia-saiba-como-funciona>>. Acesso em 03 outubro 2021.

MUTANGA, O.; DUBE, T.; GALAL, O. Remote sensing of crop health for food security in africa: Potentials and constraints. **Remote Sensing Applications: Society and Environment**, v. 8, p. 231–239, nov. 2017.

NARENDRA, V. G.; HAREESH, K. S. Prospects of computer vision automated grading and sorting systems in agricultural and food products for quality evaluation. **Int. J. Comput. Appl. Technol.**, Foundation of Computer Science, v. 1, n. 4, fev. 2010.

NIXON, M. S.; AGUADO, A. S. 8 - region-based analysis. In: NIXON, M. S.; AGUADO, A. S. (Ed.). **Feature Extraction and Image Processing for Computer Vision (Fourth Edition)**. [S.l.]: Academic Press, 2020. p. 399–432.

NOTÍCIAS Terra. 2019. Disponível em <<https://www.terra.com.br/noticias/dino/como-a-machine-learning-esta-revolucionando-o-cenario-dos-negocios,3397de1473cb228e7d5579bc46190c95kqowb3my.html>>. Acesso em 08 outubro 2021.

OJALA, T.; PIETIKAINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on feature distributions. v. 29, p. 51–59, 1996.

OLIVEIRA, A. **Assimetria e Curtose**. 2021. Disponível em <<http://alexandreprofessor.blogspot.com/p/assimetria-e-curtose.html>>. Acesso em 18 setembro 2021.

- ONYANGO, C. M. et al. Precision agriculture for resource use efficiency in smallholder farming systems in Sub-Saharan africa: A systematic review. **Sustainability**, v. 13, 2021.
- OPENCV. **Color conversions**. 2021. Disponível em <[https://docs.opencv.org/3.4/de/d25/imgproc\\_color\\_conversions.html](https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)>. Acesso em 15 setembro 2021.
- O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. nov. 2015.
- OSHIRO, T. M. **Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica**. Dissertação (Mestrado), 2013.
- PAPAGEORGIU, E. I.; MARKINOS, A. T.; GEMTOS, T. A. Fuzzy cognitive map based approach for predicting yield in cotton crop production as a basis for decision support system in precision agriculture application. **Appl. Soft Comput.**, v. 11, n. 4, p. 3643–3657, jun. 2011.
- PAUSTIAN, M.; THEUVSEN, L. Adoption of precision agriculture technologies by german crop farmers. **Precis. Agric.**, v. 18, n. 5, p. 701–716, out. 2017.
- PÉREZ-ORTIZ, M. et al. A semi-supervised system for weed mapping in sunflower crops using unmanned aerial vehicles and a crop row detection method. **Appl. Soft Comput.**, v. 37, p. 533–544, dez. 2015.
- PERISSINI, I. C. et al. **MÉTODO E SISTEMA DE ANÁLISE E PROCESSAMENTO DE IMAGENS PARA ESTIMATIVA DE COBERTURA VEGETAL POR MEIO DE SOFTWARE EM APRENDIZAGEM DE MÁQUINA**. 2022. Depositante: Fundação Universidade Federal de São Carlos, Universidade Federal de Uberlândia - Ufu, Pasture View Serviços de Agropecuária Ltda. BR n. 10 2022 004512 7. Depósito: 11 mar. 2022.
- PEZATTI, E. G. A gramática da derivação sufixal: Os sufixos formadores de substantivos abstratos. 1990.
- PRONASOLOS. **Mapa do solo brasileiro**. 2023. Disponível em <<https://geoportal.cprm.gov.br/pronasolos/>>. Acesso em 07 junho 2023.
- PURI, V.; NAYYAR, A.; RAJA, L. Agriculture drones: A modern breakthrough in precision agriculture. **J. Stat. Manage. Syst.**, Taylor & Francis, v. 20, n. 4, p. 507–518, jul. 2017.
- RADHAKRISHNA, M. V. V.; GOVINDH, M. V.; VENI, P. K. A review on image processing sensor. **J. Phys. Conf. Ser.**, IOP Publishing, v. 1714, n. 1, p. 012055, jan. 2021.
- RATH, S. R. **Image Recognition using Histogram of Oriented Gradients (HOG) Descriptor**. 2020. Disponível em <<https://debuggercafe.com/image-recognition-using-histogram-of-oriented-gradients-hog-descriptor/>>. Acesso em 20 maio 2023.

RAYLEIGH, L. On the transmission of light through an atmosphere containing small particles in suspension, and on the origin of the blue of the sky. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 47, p. 375–384, 1899.

RIBEIRO, P. P. E.; PAIVA, M. S. V. de; JORGE, L. A. de C. Application of texture analysis for differentiation of the greening from others pests. In: . [S.l.]: X Workshop de Visão Computacional, 2014.

RURAL, R. G. **O que é agricultura de precisão?** 2015. Disponível em <<https://revistagloborural.globo.com/Tecnologia-no-Campo/noticia/2015/12/o-que-e-agricultura-de-precisao.html>>. Acesso em 03 outubro 2021.

SAID, Y.; ATRI, M.; TOURKI, R. Human detection based on integral histograms of oriented gradients and SVM. **International Conference on Communications, Computing and Control Applications (CCCA)**, 2011.

SANDINO, J.; WOOLER, A.; GONZALEZ, F. Towards the automatic detection of Pre-Existing termite mounds through UAS and hyperspectral imagery. **Sensors**, v. 17, n. 10, set. 2017.

SANTOS, A. dos et al. Mapping defoliation by leaf-cutting ants *atta* species in eucalyptus plantations using the sentinel-2 sensor. **Int. J. Remote Sens.**, Taylor & Francis, v. 41, n. 4, p. 1542–1554, fev. 2020.

SANTOS, T. S. et al. Histograma de gradientes orientados na detecção de motocicletas. In: . [S.l.]: IV Jornada de Informática do Maranhão, 2012.

SAÚDE, B. **Dicas Boi Saúde**. 2021. Disponível em <<https://dicas.boisaude.com.br/melhor-peso-e-idade-para-abate-de-bovinos/>>. Acesso em 01 outubro 2021.

SCHAPIRE, R. E. The strength of weak learnability. **30th Annual Symposium on Foundations of Computer Science**, 1990.

SHEN, H.; JIANG, Z.; PAN, W. D. Efficient lossless compression of multitemporal hyperspectral image data. **Journal of Imaging**, Multidisciplinary Digital Publishing Institute, v. 4, n. 12, p. 142, dez. 2018.

SIGNORONI, A. et al. Deep learning meets hyperspectral image analysis: A multidisciplinary review. **Journal of Imaging**, Multidisciplinary Digital Publishing Institute, v. 5, n. 5, p. 52, maio 2019.

SOLOS, E. B. de P. A. E. **Sistema Brasileiro de classificação de solos**. [S.l.]: Embrapa, 2018.

STOCK, A. **Drone quadcopter with camera. New tool for farmers use drones to inspect of cultivated fields. Modern technology in agriculture**. 2022. Disponível em <<https://stock.adobe.com/br/images/drone-quadcopter-with-camera-new-tool-for-farmers-use-drones-to-inspect-of-cultivated-fields-modern-103287582>>. Acesso em 20 maio 2023.

STURM, P. Panoramic camera. **Computer Vision**, 2014.

THEODORIDIS, S. et al. **Introduction to Pattern Recognition: A Matlab Approach**. [S.l.]: Academic Press, 2010.

TIAN, L. F.; SLAUGHTER, D. C. Environmentally adaptive segmentation algorithm for outdoor image segmentation. **Comput. Electron. Agric.**, Elsevier BV, v. 21, n. 3, p. 153–168, dez. 1998.

TONINI, A. et al. Remote estimation of target height from unmanned aerial vehicle (UAV) images. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 12, n. 21, p. 3602, nov. 2020.

TRAVAINI, J. N. **Descritores de textura local para reconhecimento biométrico da íris humana**. Tese (Doutorado) — Escola de Engenharia de São Carlos, 2015.

TRUCCO, E.; VERRI, A. **Introductory techniques for 3-D computer vision**. [S.l.]: Prentice Hall, 1998.

TSOUROS, D. C.; BIBI, S.; SARIGIANNIDIS, P. G. A review on UAV-Based applications for precision agriculture. **Information**, Multidisciplinary Digital Publishing Institute, v. 10, n. 11, p. 349, nov. 2019.

ULLAH, A. et al. Agricultural innovation and sustainable development: A case study of Rice–Wheat cropping systems in south asia. **Sustainability**, v. 12, 2020.

VILJANEN, N. et al. A novel machine learning method for estimating biomass of grass swards using a photogrammetric canopy height model, images and vegetation indices captured by a drone. **Collect. FAO Agric.**, Multidisciplinary Digital Publishing Institute, v. 8, n. 5, p. 70, maio 2018.

VOGT, J. T. Detection of imported fire ant (hymenoptera: Formicidae) mounds with satellite imagery. **Environmental entomology**, 2004.

\_\_\_\_\_. Quantifying imported fire ant (hymenoptera: Formicidae) mounds with airborne digital imagery. **Environmental Entomology**, 2004.

WANG, G. et al. A comparative assessment of ensemble learning for credit scoring. **Expert Syst. Appl.**, v. 38, n. 1, p. 223–230, jan. 2011.

WHAT is depth of field (DOF)? 2019. Disponível em <<https://www.sony.com/electronics/support/articles/00031291>>. Acesso em 13 setembro 2021.

WHAT is the relationship between aperture, shutter speed, and ISO? 2019. Disponível em <<https://www.sony.com/electronics/support/articles/00025973>>. Acesso em 13 setembro 2021.

WITTENSTEIN, W. **Survey of Hyperspectral and Multispectral Imaging Technologies**. 2007.

WOEBBECKE, D. M. et al. Plant species identification, size, and enumeration using machine vision techniques on near-binary images. **Optics in Agriculture and Forestry**, 1993.

\_\_\_\_\_. Color indices for weed identification under various soil, residue, and lighting conditions. **Trans. ASAE**, v. 38, n. 1, p. 259–269, 1995.

- WU, W. B. et al. Detection of *solenopsis invicta* (hymenoptera: Formicidae) nests using spectral data. **Florida Entomologist**, Florida Entomological Society, v. 97, n. 3, p. 967–971, set. 2014.
- YANG, G. et al. Unmanned aerial vehicle remote sensing for Field-Based crop phenotyping: Current status and perspectives. **Front. Plant Sci.**, v. 8, p. 1111, jun. 2017.
- YUAN, Y.; HU, X. Random forest and objected-based classification for forest pest extraction from uav aerial imagery. **ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.**, Copernicus GmbH, XLI-B1, p. 1093–1098, jun. 2016.
- YUE, Y. et al. Assessing wheat frost risk with the support of GIS: An approach coupling a growing season meteorological index and a hybrid fuzzy neural network model. **Sustain. Sci. Pract. Policy**, Multidisciplinary Digital Publishing Institute, v. 8, n. 12, p. 1308, dez. 2016.
- ZAREIFOROUGH, H. et al. Potential applications of computer vision in quality inspection of rice: A review. **Food Engineering Reviews**, v. 7, n. 3, p. 321–345, set. 2015.
- ZHANG, C.; KOVACS, J. M. The application of small unmanned aerial systems for precision agriculture: a review. **Precis. Agric.**, v. 13, n. 6, p. 693–712, dez. 2012.
- ZHENG, Y.-J.; SONG, Q.; CHEN, S.-Y. Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. **Appl. Soft Comput.**, v. 13, n. 11, p. 4253–4263, nov. 2013.

---

## Materiais em detalhes

### A.1 Drone

O tipo de Veículo Aéreo Não Tripulado escolhido para ser utilizado no desenvolvimento desse projeto foi um quadricóptero de modelo Fimi X8 SE.

Figura 46 – Drone modelo FIMI X8 SE



Fonte: (FIMI, 2021)

Ele grava vídeos em HDR 4k, possui mecanismo de *gimbal* de 3 eixos, *design* portátil, rastreamento e modo de voo inteligente, range de sinal de 8KM, 35 minutos de voo de autonomia (com uma bateria), sistema de posicionamento visual preciso, é a prova de chuva (chuvisco), câmera integrada, controle remoto, GPS em tempo real.

#### A.1.1 Especificações técnicas

As informações técnicas abaixo estão de acordo com o site oficial da marca Fimi (FIMI, 2021).

- Dimensões: 204 X 106 X 72,6mm;
- Velocidade máxima: 18m/s;

- Limite de altitude de voo: 500m;
- Peso: 765g;
- Resistência ao vento:  $\leq 50kph$ ;
- Acurácia:
  - $\pm 0,1m$  (sem a faixa de detecção ultrassônica);
  - $\pm 0,5m$  (com o posicionamento GPS ativado).

### A.1.2 Autonomia da bateria

Tempo de voo (sem vento) : aproximadamente 35 minutos (a velocidade constante de 8m/s)

□ Especificações da bateria:

- Tensão limite de carga: 13.05V;
- Capacidade: 4500mAh / 51.3Wh;
- Tensão: 11.4V;
- Peso: aproximadamente de 270g.

### A.1.3 *Range de voo*

O *range* de voo de uma aeronave não tripulada é estabelecido de acordo com a legislação vigente no país em que a operação será realizada, mesmo que o equipamento tenha condições de realizar um range maior de voo.

De acordo com a Legislação brasileira (ICA 100-40)(ICA100-40, 2020), as especificações do VANT utilizado e com a solicitação realizada para voo, as operações realizadas se enquadram no que do que é chamado de "Operações em alturas muito baixas".

Assim sendo, o *range* de voo estabelecido pela mesma legislação enquadra-se em operação VLOS, em que limita o voo a uma distância que permita a visualização e manutenção da aeronave. VLOS é a sigla para "Operação em Linha de Visada Visual", isso significa que na operação o piloto remoto (responsável, exclusivamente, pelo manuseio dos controles de pilotagem) mantém contato visual direto com a aeronave, sem o auxílio de lentes ou outros equipamentos.

### A.1.4 Estabilidade do *Gimbal*

- Intervalo de rotação controlado:  $0^\circ \sim -90^\circ$  Ângulo de inclinação;
- Precisão de controle do ângulo:  $\pm 0.004^\circ$ ;
- Estabilização: *gimbal* de 3 eixos

### A.1.5 Câmera integrada

Como mencionado acima, o VANT possui câmera integrada. As especificações da câmera são apresentadas abaixo:

- Modelo: O tipo da câmera integrada é RGB;
- Resolução máxima da foto:  $4000 * 3000$  pixels;
- Resolução de vídeo:
  - \*  $3840 \times 2160$  30/25/24fps;
  - \*  $2560 \times 1440$  60/50/30/25/24fps;
  - \* 1080P 100/60/50/30/25/24fps;
  - \* 720P 200fps.
- Formato da imagem: JPG; JPG+DNG;
- Formato do vídeo: MP4;
- Ângulo de abertura da lente:
  - \* Lente: FOV  $80^\circ$ ;
  - \* Abertura: f2,0;
  - \* Distância focal: 3,54mm;
  - \* Distância focal equivalente: 26mm;
  - \* Sensor: 1/2,6"SONY CMOS;
  - \* Pixels efetivos: 12M;
  - \* Faixa ISO: 100 - 3200;
  - \* Velocidade de Shutter:  $32 \sim 1/8000s$ .

A principal vantagem de fazer uso de um drone com câmera integrada é o menor custo se comparado com que seria da adaptação de uma outra câmera.

Entretanto, a desvantagem é que fica-se limitado quanto ao tipo de câmeras, não podendo fazer uso, por exemplo, de uma câmera multi/hiperespectrais.

## A.2 Linguagem de Programação

Linguagem de programação é uma linguagem formal que, por meio de uma série de instruções, permite que um programador escreva um conjunto de ordens, ações consecutivas, dados e algoritmos para criar programas que controlam o comportamento físico e lógico de uma máquina. As linguagens de programação são classificadas em dois tipos: linguagem de baixo nível e de alto nível.

As linguagens de baixo nível são totalmente orientadas à máquina. Ela serve como uma interface e cria um link entre o *hardware* e o *software*. Essa categoria ainda pode ser subdividido em dois tipos:

### □ Linguagem de máquina

É uma coleção de dígitos/*bits* binários que o computador lê e interpreta;

### □ Linguagem *Assembly*

Nessa linguagem o programa escrito é armazenado como texto e consiste em uma série de instruções que correspondem ao fluxo de pedidos executáveis por um microprocessador. Entretanto, máquinas não entendem essa linguagem, portanto o texto inicial deve ser convertido em linguagem de máquina por meio de um programa chamado *Assembler*. Ele gera códigos compactos, rápidos e eficientes criados pelo programador que tem o controle da máquina.

As linguagens de programação de alto nível usam instruções que são mais fáceis de serem entendidas. Nela o programador escreve os códigos usando os idiomas conhecidos (mais comum ser o inglês) e eles são traduzidos em seguida para o idioma da máquina por meio de tradutores ou compiladores.

Os tradutores convertem programas escritos em uma linguagem de programação para a linguagem de máquina e são executados à medidas que são traduzidos. Já nos compiladores os programas traduzem o programa inteiro de uma vez, tornando-o mais rápido e podendo ser armazenado para uso posterior sem a necessidade de uma nova tradução.

A linguagem de alto nível facilita a aquisição das instruções. Isso resulta em redução do tempo de programação, a tarefa a ser executada é mais facilmente entendida e permite que o programador se desconecte da operação interna da máquina.

### A.2.1 *Open Source*

*Software* de código aberto (*open source*) é aquele cujo código-fonte é disponibilizado livremente para que os usuários possam acessá-lo e modificá-lo. A criação de *softwares* de código aberto é baseada em princípios como transparência e colaboração entre os usuários e membros da comunidade de desenvolvimento.

Há diversos benefícios na utilização desse tipo de *software*, não só para o criador/desenvolvedor inicial, mas para todos os usuários de maneira geral.

O acesso gratuito é a principal vantagem, visto que muitas vezes estudantes precisam de determinados *softwares* para fins de aprendizado e desenvolvimento de pesquisas. É um erro pensar que por ser gratuito esses *softwares* têm menor qualidade do que as opções pagas.

Outro benefício do código aberto é a constante melhora na qualidade do *software*. Devido a grande quantidade de pessoas utilizando e trabalhando nesses *softwares*, a resolução de erros e criação de novas funcionalidades são rápidas. Essa resolução de erros implica diretamente na segurança do *softwares*, dado que os erros de segurança podem ser corrigidos rapidamente.

Outro ponto positivo da utilização do *Open Source* é sua disponibilidade. Em casos de códigos fechados, a empresa que criou o produto pode descontinua-lo a qualquer momento, isso é bem problemático, visto que caso o desenvolvedor continue utilizando o *software* ele não receberá mais suporte. Já em *softwares* de código aberto, há um grande número de pessoas trabalhando para manter o projeto vivo.

Por meio da utilização de códigos abertos em larga escala, foram criados repositórios para esses códigos. Nesses repositórios, a comunidade de usuários e desenvolvedores se unem para promover melhorias no *softwares*, relatar erros encontrados no sistema e auxiliar em possíveis soluções.

### A.2.2 *Python*

Fundada em 1991 pelo desenvolvedor holandês Guido van Rossum, *python* é uma linguagem de programação dinâmica e de alto nível. Seus recursos versáteis e sua sintaxe ajuda os programadores a desenvolverem os códigos em menos etapas e em menos linhas, se comparados a linguagem Java e C++.

A linguagem *python* tem diversas possibilidades de aplicações, tanto no meio acadêmico quanto no meio empresarial.

Suas vantagens sobre as demais linguagens de programação são:

- Código aberto;
- Bibliotecas: O *python* conta com grandes bibliotecas padrões já implementadas que incluem áreas como operações de *string*, *internet*, ferramentas de serviço *web* e interfaces de sistemas operacionais. Isso faz com que a maioria das tarefas de programação que são altamente utilizadas, já estejam programadas, o que resulta no menor comprimento dos códigos;
- Facilidade de aprendizagem e codificação;

Já suas desvantagens incluem:

- Baixo uso na computação móvel;
- Sua execução é realizada por meio de um interpretador, enquanto outras linguagens fazem uso de compiladores. Isso faz com que ele possa ficar lento, dependendo da aplicação;

As vantagens da utilização de *python* ainda são maiores que as desvantagens, isso faz com que ela seja a linguagem de programação em maior ascendência de desenvolvimento, aplicações, usabilidade e na busca do mercado por desenvolvedores com proficiência na linguagem. Acredita-se também que parte da popularidade de *Python* esteja ligada ao avanço do *Machine Learning*.

#### A.2.2.1 Interface utilizada (IDE)

O Ambiente de Desenvolvimento Integrado (IDE) é um *software* que combina ferramentas comuns de desenvolvimento em uma única interface gráfica, facilitando o desenvolvimento de aplicações. Esses ambientes ajudam os desenvolvedores a programar de forma rápida, visto que o objetivo da maior parte das funcionalidades presentes em uma IDE é economizar tempo.

Algumas funcionalidades de uma IDE são:

##### ❑ Editor de código-fonte

Um editor de texto que auxilia na criação do código por meio de destaques da sintaxe com indicadores visuais, recursos de preenchimento automático e verificação de *bugs* durante a criação;

##### ❑ *Debugger*

É um recurso que mostra graficamente a localização do *bug* no código.

Além dessas funcionalidades seu uso também possui outras vantagens:

- Ajudam o desenvolvedor a organizar o fluxo de trabalho e solucionar problemas;
- Os código são analisados no momento em que está sendo escrito, assim *bugs* causados por erro humano são identificados em tempo real;
- Os desenvolvedores podem executar várias tarefas sem precisar trocar de aplicação.

Visto as diversas vantagens do uso de uma IDE a questão não é "usar ou não usar uma IDE" e sim "qual usar".

Para esse estudo, as IDEs escolhidas e utilizadas foram: *PyCharm* e *Jupyter*. O *PyCharm* fornece complementação de código inteligente, inspeção de código, realce dinâmico de erros e correções rápidas, recursos de navegação rápida e *plugins* de sincronização com o repositório utilizado para salvamento dos códigos. Já o *Jupyter Notebook*, apesar de não apresentar todas as funcionalidades de uma IDE tradicional, ele ainda assim funciona como uma ferramenta de suporte ao desenvolvimento e possui algumas vantagens, tais como: Execução e visualização da saída do código em cada célula (em blocos), uma maior facilidade na criação de protótipos de forma rápida, e a possibilidade de atualização do notebook (ou partes dele) com novos dados executando novamente as células.



---

## Base de dados e Códigos

### B.1 GitHub Projeto Tamanduá

Link: <<https://github.com/TaynaBertacine/TamanduaProject>>