

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MÉTODO PARA CLASSIFICAÇÃO DE PADRÕES DA
LAGARTA DO CARTUCHO (*Spodoptera frugiperda*) NA
CULTURA DO MILHO BASEADO EM PROCESSAMENTO
DE IMAGENS DIGITAIS E APRENDIZADO DE MÁQUINA**

ALEX BISETTO BERTOLLA

ORIENTADOR: PROF. DR. PAULO ESTEVÃO CRUVINEL

São Carlos – SP

Dezembro, 2021

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**MÉTODO PARA CLASSIFICAÇÃO DE PADRÕES DA
LAGARTA DO CARTUCHO (*Spodoptera frugiperda*) NA
CULTURA DO MILHO BASEADO EM PROCESSAMENTO
DE IMAGENS DIGITAIS E APRENDIZADO DE MÁQUINA**

ALEX BISETTO BERTOLLA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais.

Orientador: Prof. Dr. Paulo Estevão Cruvinel

São Carlos – SP

Dezembro, 2021



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Alex Bisetto Bertolla, realizada em 29/12/2021.

Comissão Julgadora:

Prof. Dr. Paulo Estevão Cruvinel (EMBRAPA)

Prof. Dr. Alexandre Luis Magalhães Levada (UFSCar)

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco (USP)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

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

Dedico este trabalho à Sabrina Volante

AGRADECIMENTOS

À minha Esposa, Sabrina Volante, pelo companheirismo e por estar sempre ao meu lado me apoiando em todos os momentos. Pela compreensão durante os períodos dedicados ao desenvolvimento deste trabalho.

Aos meus Pais, Hermínio e Neusa, pela dedicação e esforços na continuidade dos meus estudos que me permitiram chegar ao curso de Mestrado em Ciência da Computação. Aos meus irmãos, Gustavo e Ivo, pelo companheirismo e apoio.

Ao Prof. Dr. Paulo Estevão Cruvinel, pela dedicação na orientação deste trabalho. Por apoiar e acreditar na realização deste projeto de pesquisa, sempre com palavras de apoio e compreensão nos momentos mais difíceis. Pelos ensinamentos compartilhados..

Aos colegas da Embrapa Instrumentação, pela colaboração e apoio. Sobretudo, os colegas Boanerges, Flávio e Álvaro. Aos colegas da UFSCar, com quem tive a oportunidade de conviver durante o período do curso, sobre tudo, os colegas André, Wilbur e Bruno.

Ao Prof. Dr. Gabriel Alves, Prof. Dr. Ricardo Neves e Prof. Dr. Maurício Pereira, pelo convívio, compartilhamento de experiências e sugestões durante todo o período do curso.

À Universidade Federal de São Carlos (UFSCar) e ao Programa de Pós-Graduação em Ciência da Computação (PPGCC), pela oportunidade de desenvolvimento profissional e pessoal. Aos professores do programa, sempre dedicados ao ensino. Aos colaboradores do setor administrativo, sobretudo, ao Ivan sempre solícito à frente da secretaria do programa.

À CAPES, pelo apoio na concessão de bolsas de estudos aos alunos de pós-graduação.

À todos que, de alguma forma, contribuíram para a realização deste trabalho.

Pior que não terminar uma viagem é nunca partir.
(Amyr Klink)

RESUMO

A detecção, identificação e combate à Lagarta do cartucho (*Spodoptera frugiperda*) na cultura do milho (*Zea mays*) depende muito do fator humano. Atualmente, o controle ocorre principalmente por meio de armadilhas de captura. Isso faz com que o diagnóstico de infestações dessa praga seja pouco eficiente, podendo ocasionar significativos prejuízos à produção, assim como o uso excessivo de agrotóxicos. Este trabalho apresenta um método para seu controle com base no uso de técnicas do processamento de imagens e sinais para reconhecimento de padrões da Lagarta do Cartucho na cultura do milho, permitindo assim, seu reconhecimento precoce, confiável e supervisionado. Seu emprego melhora o estado da arte dos procedimentos de controle para se chegar à possibilidade de automatização. Para o desenvolvimento do método foram consideradas técnicas de obtenção de imagens, pré-processamento, segmentação, extração de características, uso da técnica PCA e classificação supervisionada. Para a etapa de aquisição de imagens, foi utilizado banco de imagens *online*. Para a etapa de pré-processamento, foram avaliados filtros Gaussiano e *Non-local Means* para a suavização de ruídos. As métricas Erro Quadrado Médio (*Mean Square Error* - MSE) e Pico da Razão Sinal-Ruído (*Peak Signal-to-Noise Ratio* - PSNR) foram utilizadas para a validação do processo de filtragem de imagens. Para a etapa de segmentação de imagens, foram avaliadas técnicas de limiarização e *pixels* sementes por meio do padrão de cores HSV (*Hue*, *Saturation* e *Value*) e CIE L*a*b*. Foi também estabelecido a automatização do processo de segmentação das imagens utilizando a técnica de Otsu. Para a etapa de extração de características, foram aplicados e considerados os descritores de textura (*Histogram of Orientation Gradient* - HOG), de momentos invariantes de Hu, para a obtenção de descrição de formas e tamanhos, assim como a redução de dimensionalidade dos vetores de características por meio da Análise de Componentes Principais (*Principal Component Analysis* - PCA). Para a classificação dos padrões da Lagarta do Cartucho foi estabelecido um conjunto de classificadores baseados em Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM). O método desenvolvido se mostrou adequado para a classificação dos padrões da Lagarta do Cartucho (*Spodoptera frugiperda*), o que contribui para o auxílio à tomada de decisão na identificação dessa praga na cultura do milho e seu controle. O método também contribui para a evolução das técnicas e ferramentas de análise, obtidas a partir de técnicas do processamento de imagens digitais.

Palavras-chave: Processamento de imagens digitais; Reconhecimento de padrões; Classificação de padrões de pragas; Aprendizado de máquina.

ABSTRACT

The detection, identification, and control of the Fall Armyworm (*Spodoptera frugiperda*) pest into the maize culture (*Zea mays*) are greatly dependent on the human factor. Currently, such control occurs mainly through the use of capture traps. This makes the diagnosis of infestations of this pest inefficient and can cause significant damage to production, as well as in general some additional use of pesticides. The objective of this research is to use image and signal processing techniques to establish a method for recognizing the Fall Armyworm (*Spodoptera frugiperda*) pattern in maize culture, allowing its early, reliable and supervised recognition, which improves the state of art of controller procedments in order to obtain an automatized process. Image aquisition, image enhancement, segmentation, features extraction, the use of Principal Components Analisis (PCA) and superviosioned classification techniques were considered for the method development. For image aquisition, it has been used an online image data base. For image enhancement, Gaussiam and Non-Local Means filters were experimented for noise reduction. Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) were measured for filters avaliation. For semgmentation process, thresholding with seed pixels were experimented on HSV and CIE L*a*b* color spaces. In order to automatize, Otsu technique was applied in the process of image segmentation. For feature extraction, Histogram of Orientation Gradient (HOG) and invariants moments of Hu were experimentd, in order to obtain texture and geometric information, respectively, as well as, for feature vector dimensionality reduction Princial Components Analisis (PCA) were experimented. For pattern classification of Fall Armyworm (*Spodoptera frugiperda*) a set of classifiers based on Support Verctor Machine (SVM) was established. The developed method has shown to be suitable for Fall Armyworm (*Spodoptera frugiperda*) pattern classification, wich has contributed to the porpouse of decision making for pest identification and control on maize culture. The method has also contributed to the evolution of digital image processing techniques and analisis tools.

Keywords: Digital image processing; Pattern recognition; Pests classification; Machine learning

LISTA DE SIGLAS

AM	Aprendizado de Máquina
BMP	Bit Map Picture
BNB	Banco do Nordeste do Brasil
CCD	Charged-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
FNLM	Fast Non-Local Means
dB	Decibéis
HOG	Histogram of Oriented Gradients
GIF	Graphics Interchanged Format
HSI	Hue Saturation Intensity
HSL	Hue Saturation Lightness
HSV	Hue Saturation Value
IA	Inteligência Artificial
JPEG/JPG	Joint Photographic (Experts) Group
KNN	K-Nearest Neighbors
MSE	Mean Squared Error
NLM	Non-Local Means
PCA	Principal Component Analysis
PID	Processamento de Imagens Digitais
PIS	Processamento de Imagens e Sinais
PNG	Portable Network Graphics

PSNR	Peak Signal-to-Noise Ratio
RGB	Red Green Blue
RNA	Rede Neural Artificial
ROI	Region Of Interest
SSIM	Structural Similarity Index Measure
SV	Support Vector
SVM	Support Vector Machine
TIF/TIFF	Tagged Image (File) Format
USDA	United States Department of Agriculture

LISTA DE FIGURAS

Figura 1 – Estágios de desenvolvimento da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>). (Fonte: Adaptado de <i>Grains Research Development Corporation</i> ¹)	25
Figura 2 – Armadilha para captura da Lagarta do Cartucho. (Fonte: Adaptado de BRUNA (2018))	26
Figura 3 – Processo padrão para um sistema de processamento de imagem digital. (Fonte: Adaptado de Dey (2018))	27
Figura 4 – Diagrama em blocos do método para classificação de padrões de da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) na cultura do milho.	40
Figura 5 – Processo de aumento do banco de imagens da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>).	45
Figura 6 – Processo de restauração de imagem. (Fonte: Adaptado de Gonzalez e Woods (2009))	47
Figura 7 – Representação gráfica do padrão RGB. (Fonte: Adaptado de Ibraheem et al. (2012))	49
Figura 8 – Representação gráfica do padrão de cor HSV. (Fonte: Adaptado de Ribeiro e Gomes (2010))	50
Figura 9 – Representação gráfica dos padrão de cor HSI. (Fonte: Adaptado de Matos et al. (2007))	50
Figura 10 – Representação gráfica do padrão CIE L*a*b*. (Fonte: Adaptado de Moghadam et al. (2013))	54
Figura 11 – Máquina de vetores de suporte com margens rígidas. (Fonte: Noyum et al. (2021))	67
Figura 12 – Máquina de vetores de suporte com margens suaves. (Fonte: Pecha e Horák (2020))	70
Figura 13 – Conjunto de dados não linearmente separáveis (espaço de entrada) convertido para em espaço de maior dimensão (espaço de características). (Fonte: Adaptado de Nalini e Palanivel (2016))	71
Figura 14 – Processo de treinamento do conjunto de classificadores do tipo SVM.	73
Figura 15 – Exemplo de curva ROC. (Fonte: <i>Scikit-learn webpage</i> ²)	76

Figura 16 – Exemplos da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>). (Fonte: (a) Clemson University - USDA Cooperative Extension Slide Series , Bugwood.org, (b) John C. French Sr., Retired, Universities:Auburn, GA, Clemson and U of MO, Bugwood.org)	77
Figura 17 – Exemplos da Lagarta do cartucho (<i>Spodoptera frugiperda</i>) geradas por meio da técnica <i>data augmentation</i> . (a) Imagem da lagarta gerada sobre uma folha, (b) imagem da lagarta gerada sobre uma espiga de milho, (c) imagem composta da lagarta gerada sobre uma folha e (d) imagem composta da lagarta gerada sobre uma espiga de milho.	78
Figura 18 – Resultado do filtro Gaussiano. (a) Imagem original, (b) imagem afetada por ruído do tipo Gaussiano e (c) imagem filtrada.	79
Figura 19 – Resultado do filtro <i>Non-local Means</i> . (a) Imagem original, (b) imagem afetada por ruído do tipo Gaussiano e (c) imagem filtrada.	80
Figura 20 – Resultado do cálculo da métrica MSE do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e <i>Non-local Means</i>	81
Figura 21 – Resultado do cálculo da métrica PSNR do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e <i>Non-local Means</i>	81
Figura 22 – Resultado da conversão do padrão de cor RGB para o padrão HSV. (a) Imagem original, (b) mapa de H.	82
Figura 23 – Resultado da conversão do padrão de cor RGB para o padrão HSV de uma imagem com fundo de espiga de milho. (a) Imagem original, (b) mapa de H.	83
Figura 24 – Resultado da conversão do padrão de cor RGB para o padrão CIE L*a*b* de uma imagem com fundo de folha. (a) Imagem original, (b) mapa de a* e (c) mapa de b*.	83
Figura 25 – Resultado da conversão do padrão de cor RGB para o padrão CIE L*a*b* de uma imagem com fundo de espiga. (a) Imagem original, (b) mapa de a* e (c) mapa de b*.	84
Figura 26 – Resultado da segmentação por limiarização e <i>pixels</i> sementes com proporção de $1,5\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H e (d) imagem segmentada.	85
Figura 27 – Resultado por segmentação por limiarização e <i>pixels</i> sementes com proporção de 2σ . (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.	86
Figura 28 – Resultado da segmentação por limiarização e <i>pixels</i> sementes com proporção de $0,6\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.	86
Figura 29 – Resultado da segmentação por limiarização e <i>pixels</i> sementes com proporção de $0,7\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.	87

Figura 30 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) mapa de a^* , (c) histograma do mapa de a^* e (d) resultado da segmentação pelo mapa de a^*	88
Figura 31 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) mapa de b^* , (c) histograma do mapa de b^* e (d) resultado da segmentação pelo mapa de b^*	89
Figura 32 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) resultado da segmentação pelo mapa de a^* , (c) resultado da segmentação pelo mapa de a^* e (d) resultado final da segmentação.	90
Figura 33 – Resultado da segmentação de uma imagem da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) pelo método de Otsu. (a) Imagem original, (b) mapa de b^* , (c) imagem segmentada.	91
Figura 34 – Resultado da extração da características de uma imagem da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) pelo descritor HOG. (a) Imagem da lagarta segmentada, (b) imagem HOG da lagarta segmentada.	93
Figura 35 – Resultado da extração da características de uma imagem da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) pelos momentos invariantes de Hu. (a) Imagem da lagarta no instar 1 segmentada na posição vertical, (b) Imagem da lagarta no instar 3 segmentada na posição horizontal.	93
Figura 36 – Exemplo do vetor de características $Vetor_{[HOG,Hu]}$	94
Figura 37 – Exemplo do vetor de características $Vetor_{[HOG,Hu]}$ obtido por meio da técnica de PCA.	95
Figura 38 – Matriz de confusão do Classificador 1 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	97
Figura 39 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	98
Figura 40 – Matriz de confusão do Classificador 2 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	98
Figura 41 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	99
Figura 42 – Matriz de confusão do Classificador 3 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	99
Figura 43 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	100
Figura 44 – Matriz de confusão do Classificador 4 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	100
Figura 45 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	101

Figura 46 – Matriz de confusão do Classificador 5 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	101
Figura 47 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	102
Figura 48 – Matriz de confusão do Classificador 1 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	103
Figura 49 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	103
Figura 50 – Matriz de confusão do Classificador 2 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	104
Figura 51 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	104
Figura 52 – Matriz de confusão do Classificador 3 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	105
Figura 53 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	105
Figura 54 – Matriz de confusão do Classificador 4 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	106
Figura 55 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	106
Figura 56 – Matriz de confusão do Classificador 5 com função <i>kernel</i> linear e proporção 50% teste 50%treinamento.	107
Figura 57 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 50% teste 50% treinamento.	107
Figura 58 – Matriz de confusão do Classificador 1 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	108
Figura 59 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	109
Figura 60 – Matriz de confusão do Classificador 2 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	109
Figura 61 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	110
Figura 62 – Matriz de confusão do Classificador 3 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	110
Figura 63 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	111
Figura 64 – Matriz de confusão do Classificador 4 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	111

Figura 65 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	112
Figura 66 – Matriz de confusão do Classificador 5 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	112
Figura 67 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	113
Figura 68 – Matriz de confusão do Classificador 1 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	114
Figura 69 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	114
Figura 70 – Matriz de confusão do Classificador 2 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	115
Figura 71 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	115
Figura 72 – Matriz de confusão do Classificador 3 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	116
Figura 73 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	116
Figura 74 – Matriz de confusão do Classificador 4com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	117
Figura 75 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	117
Figura 76 – Matriz de confusão do Classificador 5 com função <i>kernel</i> linear e proporção 70% teste 30%treinamento.	118
Figura 77 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	118
Figura 78 – Matriz de confusão do Classificador 1 com função <i>kernel</i> linear e proporção 80% teste 20%treinamento.	119
Figura 79 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 70% teste 30% treinamento.	120
Figura 80 – Matriz de confusão do Classificador 2 com função <i>kernel</i> linear e proporção 80% teste 20%treinamento.	120
Figura 81 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	121
Figura 82 – Matriz de confusão do Classificador 3 com função <i>kernel</i> linear e proporção 80% teste 20%treinamento.	121
Figura 83 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	122

Figura 84 – Matriz de confusão do Classificador 4 com função <i>kernel</i> linear e proporção 80% teste 20%treinamento.	122
Figura 85 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	123
Figura 86 – Matriz de confusão do Classificador 5 com função <i>kernel</i> linear e proporção 80% teste 20%treinamento.	123
Figura 87 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	124
Figura 88 – Matriz de confusão do Classificador 1 proporção 80% teste 20%treinamento.	125
Figura 89 – Curva ROC do Classificador 1 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	125
Figura 90 – Matriz de confusão do Classificador 2 proporção 80% teste 20%treinamento.	126
Figura 91 – Curva ROC do Classificador 2 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	126
Figura 92 – Matriz de confusão do Classificador 3 proporção 80% teste 20%treinamento.	127
Figura 93 – Curva ROC do Classificador 3 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	127
Figura 94 – Matriz de confusão do Classificador 4 proporção 80% teste 20%treinamento.	128
Figura 95 – Curva ROC do Classificador 4 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	128
Figura 96 – Matriz de confusão do Classificador 5 proporção 80% teste 20%treinamento.	129
Figura 97 – Curva ROC do Classificador 5 com função <i>kernel</i> linear e proporção 80% teste 20% treinamento.	129
Figura 98 – Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	130
Figura 99 – Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	131
Figura 100–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	131
Figura 101–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	132
Figura 102–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	132
Figura 103–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	133
Figura 104–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	133
Figura 105–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	134

Figura 106–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	134
Figura 107–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	135
Figura 108–Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	136
Figura 109–Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	136
Figura 110–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	137
Figura 111–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	137
Figura 112–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	138
Figura 113–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	138
Figura 114–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	139
Figura 115–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	139
Figura 116–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 50% teste 50%treinamento.	140
Figura 117–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 50% teste 50% treinamento.	140
Figura 118–Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	141
Figura 119–Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	142
Figura 120–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	142
Figura 121–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	143
Figura 122–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	143
Figura 123–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	144
Figura 124–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	144

Figura 125–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	145
Figura 126–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	145
Figura 127–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	146
Figura 128–Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	147
Figura 129–Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	147
Figura 130–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 70% teste 30%treinamento.	148
Figura 131–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	148
Figura 132–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 70% teste 30%treinamento.	149
Figura 133–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	149
Figura 134–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 70% teste 30%treinamento.	150
Figura 135–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	150
Figura 136–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 70% teste 30%treinamento.	151
Figura 137–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 70% teste 30% treinamento.	151
Figura 138–Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	152
Figura 139–Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	153
Figura 140–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	153
Figura 141–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	154
Figura 142–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	154
Figura 143–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	155

Figura 144–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	155
Figura 145–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	156
Figura 146–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	156
Figura 147–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	157
Figura 148–Matriz de confusão do Classificador 1 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	158
Figura 149–Curva ROC do Classificador 1 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	158
Figura 150–Matriz de confusão do Classificador 2 com função <i>kernel</i> sigmoide e proporção 80% teste 20%treinamento.	159
Figura 151–Curva ROC do Classificador 2 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	159
Figura 152–Matriz de confusão do Classificador 3 com função <i>kernel</i> sigmoide e proporção 80% teste 20%treinamento.	160
Figura 153–Curva ROC do Classificador 3 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	160
Figura 154–Matriz de confusão do Classificador 4 com função <i>kernel</i> sigmoide e proporção 80% teste 20%treinamento.	161
Figura 155–Curva ROC do Classificador 4 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	161
Figura 156–Matriz de confusão do Classificador 5 com função <i>kernel</i> sigmoide e proporção 80% teste 20%treinamento.	162
Figura 157–Curva ROC do Classificador 5 com função <i>kernel</i> sigmoide e proporção 80% teste 20% treinamento.	162
Figura 158–Diagrama do método para classificação de padrões de da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) na cultura do milho.	164

LISTA DE TABELAS

Tabela 2 – Maiores produtores mundiais de milho, safra 2018/2019 (Fonte: Adaptado de <i>World Production, Markets, and Trade Report</i> ³)	24
Tabela 3 – Produção nacional de milho, por regiões (Fonte: Adaptado de Caderno Setorial ETENE ⁴)	24
Tabela 1 – Produtos derivados do milho e suas principais aplicações (Fonte: Adaptado de Strazzi (2015))	24
Tabela 4 – Principais formatos de imagens digitais (Fonte: Adaptado de Solomon e Breckon (2000))	41
Tabela 5 – Características das imagens da Lagarta do Cartucho obtidas pelo banco de imagens <i>Insect Image</i>	42
Tabela 6 – Tipos de <i>Kenels</i> (Fonte Lorena e Carvalho (2007))	72
Tabela 7 – Matriz de confusão para solução de problemas binários.	74
Tabela 8 – Valores dos parâmetros para o filtro do tipo Gaussiano.	79
Tabela 9 – Valores dos parâmetros para o filtro do tipo <i>Non-local Means</i>	79
Tabela 10 – Valores dos parâmetros do descritor HOG.	92
Tabela 11 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 50% para treinamento e 50% para teste.	97
Tabela 12 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 50% para treinamento e 50% para teste.	102
Tabela 13 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 70% para treinamento e 30% para teste.	108
Tabela 14 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 70% para treinamento e 30% para teste.	113

Tabela 15 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 80% para treinamento e 20% para teste.	119
Tabela 16 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função <i>kernel</i> linear utilizando a proporção de 80% para treinamento e 20% para teste.	124
Tabela 17 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 50% para treinamento e 50% para teste.	130
Tabela 18 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 50% para treinamento e 50% para teste.	135
Tabela 19 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 70% para treinamento e 30% para teste.	141
Tabela 20 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 70% para treinamento e 30% para teste.	146
Tabela 21 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 80% para treinamento e 20% para teste.	152
Tabela 22 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro <i>Non-local Means</i> para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) por meio do classificador SVM com função sigmoide utilizando a proporção de 80% para treinamento e 20% para teste.	157
Tabela 23 – Conjunto de classificadores SVM para a classificação de padrões da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>).	163

SUMÁRIO

CAPÍTULO 1–INTRODUÇÃO	23
1.1 Contextualização e Motivação	23
1.2 Processamento de imagens digitais	26
1.3 Trabalhos relacionados	28
1.3.1 Abordagem baseada em aquisição de imagens aumento de dados	28
1.3.2 Abordagem baseada no pré-processamento de imagens	29
1.3.3 Abordagem baseada em segmentação de imagens	31
1.3.4 Abordagem baseada em extração de características	32
1.3.5 Abordagem baseada em aprendizado de máquina	33
1.4 Objetivo geral	37
1.5 Objetivos específicos	37
1.6 Organização do documento	38
CAPÍTULO 2–MATERIAIS E MÉTODOS	39
2.1 Abordagem metodológica e principais materiais	39
2.2 Pré-processamento das imagens da Lagarta do Cartucho em cultura do milho	46
2.3 Segmentação da imagem	55
2.4 Extração de características	60
2.4.1 Histograma de Gradientes Orientados	60
2.4.2 Momentos invariantes de Hu	60
2.4.3 Redução de dimensionalidade do vetor de características	63
2.5 Reconhecimento de padrões e aprendizado de máquina	65
2.5.1 Máquina de vetores de suporte	66
2.5.2 SVM linear com margens rígidas	66
2.5.3 SVM linear com margens suaves	69
2.5.4 SVM não linear	71
2.5.5 Conjunto de classificadores SMV	73
CAPÍTULO 3–RESULTADOS E DISCUSSÕES	77
3.1 Qualificação do banco de imagem da Lagarta do Cartucho (<i>Spodoptera frugiperda</i>) em cultura do milho	77
3.2 Resultados obtidos por meio da etapa de filtragem de ruídos	78
3.2.1 Avaliação da etapa de filtragem de ruídos	80
3.3 Resultados obtidos por meio da etapa de operação de conversão de padrões de cores	82
3.3.1 Avaliação da etapa de operação de conversão de padrões de cores	84

3.4	Resultados obtidos por meio da etapa de segmentação de imagens	84
3.4.1	Avaliação da etapa de segmentação de imagens	91
3.5	Resultados obtidos por meio da etapa de extração de características	92
3.5.1	Avaliação da etapa de extração de características	94
3.6	Resultados obtidos por meio da etapa de redução de dimensionalidade do vetor de características	94
3.6.1	Avaliação da etapa de redução de dimensionalidade do vetor de características	95
3.7	Resultados obtidos por meio da etapa de treinamento do conjunto de classificadores SVM	95
3.7.1	Avaliação da etapa de treinamento do conjunto de classificadores SVM	163
CAPÍTULO 4–CONCLUSÕES		166
4.1	Principais conclusões	166
4.2	Dificuldades e limitações encontradas	168
4.3	Sugestões para Trabalhos Futuros	168
REFERÊNCIAS		169

Capítulo 1

INTRODUÇÃO

Neste Capítulo são apresentadas a motivação para o desenvolvimento da pesquisa, assim como, a contextualização pela seleção das técnicas para o processamento das imagens digitais no controle de pragas que afeta a produção do milho (*Zea mays*). Também é apresentada a revisão da literatura em estado da arte relacionada ao tema para o desenvolvimento do método apresentado.

1.1 Contextualização e Motivação

A cultura do milho (*Zea mays*) é, certamente, uma das mais importantes no cenário agrícola mundial. O milho é um dos cereais mais cultivados no planeta. Existem indícios do cultivo do milho que datam de cerca de sete mil anos em regiões onde hoje se localiza o México (ABIMILHO, 2021).

A importância econômica que o milho tem hoje se caracteriza principalmente pela diversidade na forma de consumo. Principal fonte de alimento para os povos americanos, o milho é nutritivo, pois diferentemente de outros grãos como o arroz e o trigo, por exemplo, o grão do milho conserva sua casca, que é rica em fibras e fundamental para a eliminação de toxinas (SINDMILHO, 2005).

Apesar do milho ser a base da alimentação humana e animal, suas formas de consumo vão muito além, tendo grande aplicação na indústria e no setor energético. Ainda no que se refere à alimentação humana, o milho é, em sua maior parte, consumido como o fubá, farinhas, óleo de cozinha, entre outros. O consumo do milho para a alimentação animal, principalmente para animais domésticos, tem crescido nos últimos anos na forma de rações. Porém, sua utilização como fonte de alimento animal se destaca na criação de suínos e aves (ABIMILHO, 2021).

Na indústria fabril, o milho é utilizado na fabricação de papelão ondulado e fitas adesivas (REGITANO-D'ARCE et al., 2015). Na indústria alimentícia, é fonte para o xarope de glucose, muito utilizado na fabricação de doces e balas, também para a fabricação do corante caramelo, utilizado principalmente na fabricação de refrigerantes e cervejas (STRAZZI, 2015). Já no setor energético, o milho é utilizado para a fabricação do etanol, sendo os Estados Unidos os principais

Tabela 2 – Maiores produtores mundiais de milho, safra 2018/2019 (Fonte: Adaptado de *World Production, Markets, and Trade Report*¹)

País	Safra 2019/2020 (milhões de toneladas)	Previsão Safra 2020/2021 (milhões de toneladas)
Estados Unidos	345,96	358,45
China	260,78	260,67
Brasil	102,00	86,00
Produção Mundial	1.119,68	1.119,02

Tabela 3 – Produção nacional de milho, por regiões (Fonte: Adaptado de Caderno Setorial ETENE²)

Unidade geográfica	Safra 2019/2020 (mil toneladas)	Previsão Safra 2020/2021 (mil toneladas)
Centro-Oeste	56.836,0	51.612,4
Norte	3.518,7	3.466,5
Sul	21.663,1	20.828,2
Sudeste	11.764,0	11.655,1
Nordeste	8.804,6	8.829,9
Total	102.586,4	96.392,1

produtores desse tipo de biocombustível de derivado do milho (BORTOLETTO; ALCARDE, 2015).

A Tabela 1 detalha alguns produtos derivados do milho e suas principais aplicações.

Tabela 1 – Produtos derivados do milho e suas principais aplicações (Fonte: Adaptado de Strazzi (2015))

Produtos	Principais aplicações
Canjicas	alimentos matinais, bebidas alcoólicas
Canjiquinhas	Salgadinhos (<i>snacks</i>)
Farinha de milho pré-gelatinizada	Alimentos instantâneos
Dextrina	Adesivos, produtos farmacêuticos
Amido de milho	Panificação, fermento em pó

De acordo com o Departamento de Agricultura dos Estados Unidos (*United States Department of Agriculture - USDA*) o Brasil é o terceiro maior produtor de milho no mundo, seguido apenas dos Estados Unidos e China, conforme detalhado na Tabela 2. No Brasil, em relação à safra 2019/2020, a região Centro-Oeste foi a maior produtora de milho segundo o Banco do Nordeste do Brasil (BNB), conforme detalhado na Tabela 3.

¹ Disponível em: <<https://apps.fas.usda.gov/psdonline/circulars/production.pdf>>. Acessado em 01 de dez. de 2021.

² Disponível em: <https://www.bnb.gov.br/s482-dspace/bitstream/123456789/910/1/2021_CDS_182.pdf>. Acessado em 01 de dez. de 2021.

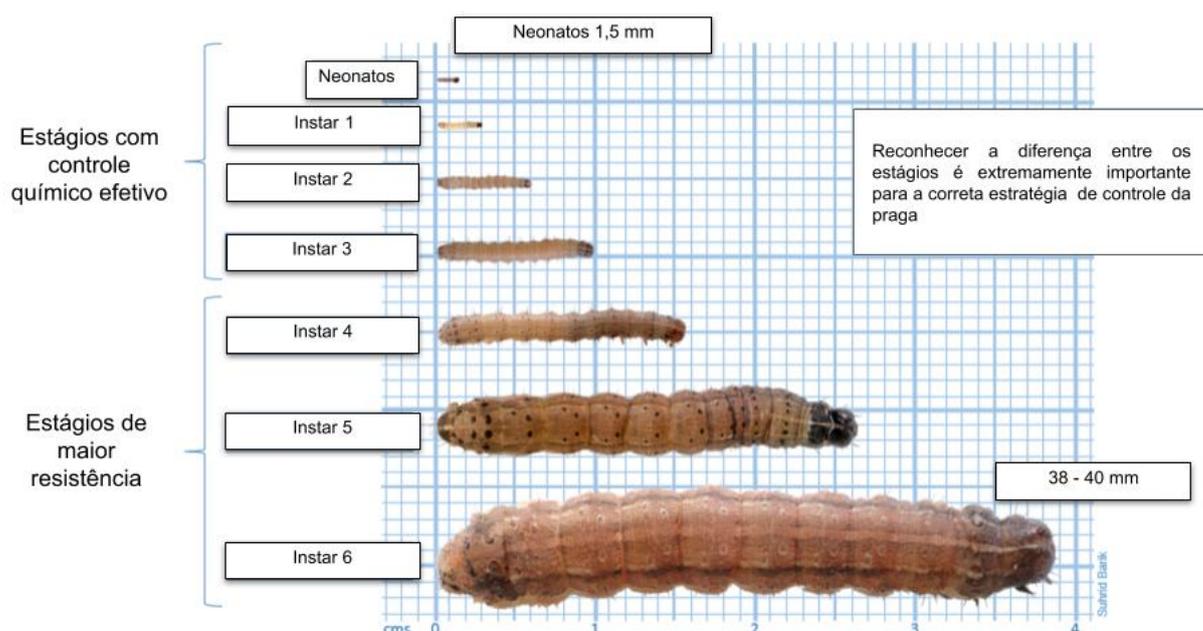
No entanto, cultura do milho é atacada por diferentes tipos de pragas e doenças, em diferentes estágios do desenvolvimento das plantas. Ataques de pragas e doenças afetam severamente o potencial produtivo do milho, pois diferentemente de outras culturas como a soja e o trigo, por exemplo, o cultivo do milho é baseado em um número relativamente pequeno de plantas por área, ou seja, mesmo que a perda da planta seja parcial, esse fato causa grande prejuízo à produção (CASELA et al., 2011).

Dentre essas principais pragas da cultura do milho, destaca-se a Lagarta do Cartucho (*Spodoptera frugiperda*). Ela é considerada uma das principais e mais devastadoras pragas dessa cultura, podendo causar prejuízos da ordem de 60% da produção, pois ataca a planta ainda em seu estágio de formação (VIANA et al., 2011; ROSA; BARCELOS, 2012).

A Figura 1 ilustra a Lagarta do Cartucho em seus diferentes estágios de desenvolvimento, também chamados de instar. É importante destacar que em relação ao estágio neonateo (recem-nascida), dado o tamanho da lagarta, é inviável a tentativa de sua detecção por imagem, sendo a detecção de colônias (ovos) uma alternativa.

Por outro lado, no que se refere às características, método de controle e danos à cultura do milho, os instares 5 e 6 podem ser tratados como um único instar, de acordo com especialistas. Dessa forma, o foco principal para a classificação dos padrões da Lagarta do Cartucho são os instares 1 à 5.

Figura 1 – Estágios de desenvolvimento da Lagarta do Cartucho (*Spodoptera frugiperda*). (Fonte: Adaptado de *Grains Research Development Corporation*³)



³ Disponível em: <<https://www.planthealthaustralia.com.au/fall-armyworm/>>. Acessado em 01 de dez. de 2021.

Atualmente um dos principais métodos de monitoramento da Lagarta do Cartucho na cultura do milho é a sua captura por meio da utilização de armadilhas (Figura 2), de forma à capturar machos pela exalação de odor de feromônios similares ao das fêmeas. Uma vez identificada a presença da praga em áreas de cultura, são utilizados métodos que envolvem a aplicação de agrotóxicos (inseticidas) para seu controle.

Figura 2 – Armadilha para captura da Lagarta do Cartucho. (Fonte: Adaptado de BRUNA (2018))



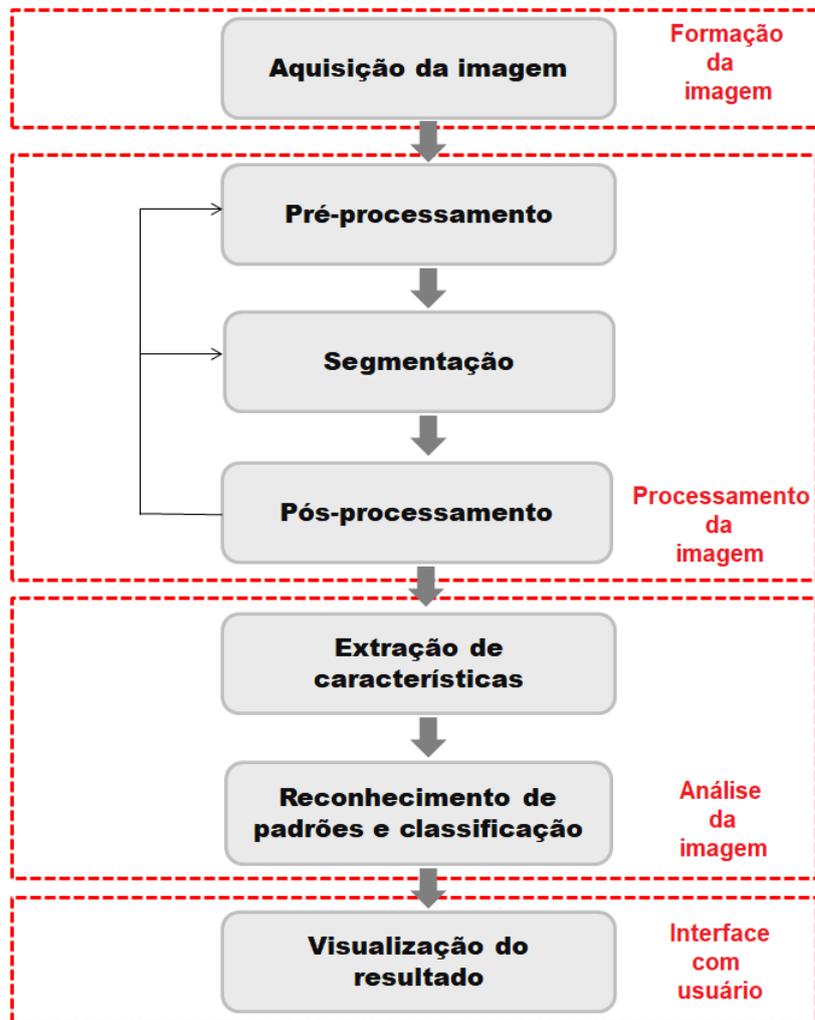
Porém, essa técnica possibilita capturar somente as espécimes que já estão prestes a se transformarem em mariposas ou, as que já estão nessa forma, momento em que já não causam significativo dano à produção (BRUNA, 2018). Tal aspecto motivou a pesquisa a encontrar outros caminhos para a detecção precoce dessa praga em área de cultura, o que enfoca a utilização do processamento de imagens e sinais (PIS) para a busca de soluções e inovação em métodos que possam auxiliar no reconhecimento e classificação dessa praga em diferentes estágios, a fim de favorecer a produtividade da cultura do milho.

1.2 Processamento de imagens digitais

O processamento de imagens digitais (PID) pode ser entendido como um conjunto de técnicas de transformação, análise e interpretação de uma imagem digital que são implementadas por meio de algoritmos computacionais (DEY, 2018). As técnicas de PID relacionam diferentes áreas do conhecimento (JAHNE, 2004).

A Figura 3 ilustra o conjunto de processos padrões para um sistema de processamento de imagens digitais.

Figura 3 – Processo padrão para um sistema de processamento de imagem digital. (Fonte: Adaptado de Dey (2018))



Aquisição da imagem pode ser compreendida como etapa na qual a imagem é obtida (câmeras digitais por exemplo) e como será armazenada (meio físico, tipo do arquivo). Pré-processamento é a etapa na qual ocorre a execução de algoritmos para a correção da imagem (eliminação de ruídos, melhoria do contraste, entre outras operações). A etapa de segmentação representa a etapa na qual é realizada a separação do objeto, ou regiões, de interesse do restante da imagem (fundo). Extração de características é compreendida como a etapa onde são caracterizadas as informações da imagem (textura, forma, etc.) e extraídas do objeto de interesse. Por fim, na etapa de reconhecimento de padrões e classificação é possível reconhecer os objetos de interesse e verificar se pertencem a uma determinada classe, ou não, a partir das informações extraídas na etapa anterior.

1.3 Trabalhos relacionados

A visão computacional busca, por meio de modelos computacionais, reproduzir a capacidade e funções da visão humana, ou seja, a capacidade de enxergar e interpretar uma cena. A capacidade de enxergar pode ser executada por meio do uso de dispositivos de aquisição de imagens e métodos adequados para o reconhecimento de padrões. Da mesma forma, a aplicação de inteligência artificial (IA) e do aprendizado de máquina (AM) podem auxiliar o processo de tomada de decisão para o estabelecimento de diagnóstico orientado ao controle dessa praga em área de cultura do milho.

Atualmente o uso de técnicas do PIS podem ser encontradas em diversas áreas. Dentre elas encontra destaque as áreas de medicina (FERNANDES et al., 2012; ARAÚJO et al., 2012; XAVIER et al., 2012), indústria, segurança (OLIVEIRA; GONZAGA, 2012; SILVA et al., 2012; MENKE et al., 2017), agricultura (DELGADO et al., 2012; OYAMA et al., 2012; WU; SUN, 2013; MENKE et al., 2017), entre outras.

Neste trabalho, a revisão bibliográfica em estado da arte foi realizada considerando consultas às bases bibliográficas: *Google Scholar*; *IEEE Xplore*; *Precision Agriculture*; *Scopus*; *Spring* e *Elsevier*, resultando na seleção de aproximadamente 100 artigos relevantes e de interesse para a pesquisa. Para o gerenciamento do material bibliográfico foi utilizada a ferramenta *Mendeley*.

1.3.1 Abordagem baseada em aquisição de imagens aumento de dados

Com base nas palavras-chave: *image processing*, *pest classification*, *data augmentation* e *pest control*, houve o levantamento em estado da arte no tema aquisição de imagens e aumento de dados.

Com o objetivo de auxiliar a identificação de plantas doentes com *greening* ou com alguma deficiência nutricional, Sankaran e colaboradores propuseram um método baseado em espectroscopia do infra-vermelho médio (SANKARAN et al., 2010). As análises foram divididas em duas etapas onde as amostras foram analisadas por um espectrômetro de infravermelho-médio portátil. Primeiramente foram calculadas a primeira e segunda derivada dos espectros das amostras foliares e em seguida aplicada a análise por PCA. Em seguida dois algoritmos de classificação foram aplicados, a análise discriminante quadrática e KNN. Foi observado também que o pico do espectro correspondente à faixa de valores entre $9\mu\text{m}$ à $10,5\mu\text{m}$ permite diferenciar as amostras foliares doentes com *greening*, das amostras foliares com deficiência nutricional e saudáveis, devido ao fato das folhas doentes possuírem um acúmulo de amido.

Miranda e colaboradores propuseram o estudo de diferentes técnicas de processamento digital de imagem para a detecção de pragas em arrozais (MIRANDA et al., 2014). Eles propuseram uma metodologia na qual, em uma sequência de imagens, duas imagens eram comparadas *pixel a pixel*. A primeira imagem era utilizada como imagem de referência e a segunda como

imagem de entrada. Caso o valor do *pixel* da imagem de entrada fosse igual ao valor do *pixel* da imagem de referência, o valor deste *pixel* era convertido para 255 (branco). Porém, se os valores dos *pixel* fossem diferentes, o valor do *pixel* da imagem de entrada era salvo como valor do *pixel* da imagem de saída. Após este processo, a imagem de entrada era considerada como imagem de referência e o processo seguia até o processamento de todas as imagens. Como resultado, esse processo gerava uma imagem de fundo branco destacando apenas os *pixel* nos quais as pragas foram detectadas. Para a eliminação de ruído, filtros de mediana foram aplicados. As imagens foram então verificadas *pixel a pixel*, tanto horizontal quanto verticalmente. Esse processo foi realizado para que fosse possível a detecção e o cálculo do tamanho (em *pixel*) da praga detectada.

Mikolajczyk e Grochowski compararam as principais técnicas de *data augmentation* para a classificação de imagens e propuseram um método a partir do uso de redes neurais convolucionais (MIKOŁAJCZYK; GROCHOWSKI, 2018). Para validação do método proposto os autores utilizaram imagens médicas para diagnosticar doenças de pele e câncer de mama. Os demonstraram que o uso de técnicas tradicionais de *data augmentation*, como rotação, translação, adição de ruídos, entre outros, são suficientes para a melhora do processo de treinamento de classificadores, porém, o uso de aprendizado profundo permite gerar modelos de imagens maior potencial de melhora da acurácia e diminuição de *overfitng* nas etapas de treinamento, teste e validação no processo de classificação de imagens.

1.3.2 Abordagem baseada no pré-processamento de imagens

Com base nas palavras-chave: *digital filters, image filtering process, Gaussian filters e Non-local Means filters*, houve o levantamento em estado da arte no tema pré-processamento de imagens.

Para analisar o impacto causado por gotas de chuva nos solos agrícolas, Cruvinel e colaboradores propuseram um método para a identificação do tamanho e distribuição das gotas baseados em técnicas de processamento de imagens com análise de correlação no domínio da frequência (CRUVINEL et al., 1999). A operação de correlação foi obtida por meio da transformada inversa de Fourier de uma imagem de entrada, a qual continha os padrões de gotas a serem analisados e reconhecidos e uma imagem com os padrões já reconhecidos. Como medidas de referências os autores utilizaram volume, variando de 0,0054 até 0,0810cm³, diâmetro, variando de 1,2 a 1,5mm e uma referência externa com diâmetro de 10mm. Utilizando tais parâmetros foi possível determinar três fatores importantes, o volume total de água em cada amostra, a fração de gotas em cada intervalo e a distribuição acumulativa de tamanho. Por fim, os autores concluíram que o método proposto se mostrou viável para a identificação automática de gotas de chuva, tanto em performance quanto em tempo de processamento e confiabilidade.

Buades e colaboradores apresentaram um método para filtragem de imagens digitais baseado em médias não locais (*Non-local Means*). Este método foi comparado com métodos

tradicionais de filtragem de imagens digitais, como por exemplo, filtro Gaussiano, filtro de difusão anisotrópica e filtro baseado em vizinhança (BUADES et al., 2005). Para validação do método os autores adicionaram ruído branco em imagens digitais e, por meio da métrica MSE mostraram a superioridade do método desenvolvido em relação aos filtros que utilizaram como comparação.

Mythili e Kavitha compararam a eficiência da aplicação de diferentes tipos de filtros para redução de ruídos em imagens digitais coloridas (MYTHILI; KAVITHA, 2011). Para a validação, as autoras utilizaram filtros lineares (filtro de média e filtro Gaussiano), filtros não lineares (mediana e filtro *Fuzzy*) para a redução de ruídos do tipo aditivo (ruído Gaussiano), impulsivo (sal-e-pimenta), multiplicativo (ruído *speckle*), entre outros. Para avaliar a performance de cada filtro, as autoras utilizaram a métrica MSE e PSNR. De acordo com as autoras, os resultados demonstraram melhor eficiência do filtro *Fuzzy* para a redução dos ruídos utilizados nos experimentos, pois esse tipo de filtro foi capaz de reduzir o ruído presente nas imagens e também preservar a qualidade das imagens. Porém, a utilização do filtro *Fuzzy* necessitou de maior poder computacional em relação aos filtros de média, mediana e Gaussiano.

Mirsha e colaboradores compararam três tipos de filtros para a redução de ruídos em imagens digitais. Foram utilizados os filtros de Wiener, Lucy-Richardson e de regularização (*Regularized Filter*) (MISHRA et al., 2019). Como exemplos de ruídos os autores utilizaram ruídos do tipo Gaussiano e borramento Gaussiano (*Gaussian Blur*). Para a validação da proposta, os autores utilizaram as métricas MSE, PSNR e SSIM. De acordo com os autores, os resultados demonstraram que para a restauração de imagens apenas com ruído do tipo borramento Gaussiano todos os filtros obtiveram resultados similares, porém, para tratamento de imagens com borramento e ruído Gaussiano, o filtro Lucy-Richardson obteve os melhores resultados. Para a suavização de imagens com ruído Gaussiano, o filtro de regularização obteve o pior resultado e os resultados obtidos pelo filtro de Wiener e pelo filtro Lucy-Richardson foram próximos, porém, a aplicação das métricas PSNR, MSE e SSIM mostraram que o filtro Lucy-Richardson obteve os melhores resultados em todos os cenários apresentados.

Kang e Kim apresentaram um estudo para avaliar e comparar a parametrização do filtro *Fast Non-local Means* (FNLM) em comparação ao filtro *Non-local Means* (NLM) convencional (KANG; KIM, 2021). Como parâmetros, os autores avaliaram o tamanho do *kernel* do filtro, o tamanho da janela de busca e também o fator de filtragem (*smoothing factor*). Como métricas para validação do estudo, os autores utilizaram o coeficiente de variação (COV), avaliador de qualidade espacial da imagem (BRISQE) e avaliador da qualidade da imagem natural (NIQE). De acordo com os autores, os resultados obtidos mostraram que a aplicação do filtro FNLM com alto fator de filtragem foi mais eficiente, porém, exceder esse fator pode degradar a imagem a ponto de perder informações, como por exemplo, informações de textura. Os resultados também mostraram que, em relação ao tamanho do *kernel* e a janela de busca, quanto maior o *kernel* e menor a janela de busca, melhor o resultado do processo de filtragem. Por fim, os autores

compararam a aplicação do filtro FNLM com o filtro NLM convencional em relação ao tempo de execução. O tempo de execução do filtro FNLM foi nove vezes mais rápido que o filtro NLM convencional.

Bertolla e Cruvinel desenvolveram um método para filtragem de imagens digitais afetadas por ruídos não estacionários (BERTOLLA; CRUVINEL, 2021). Na abordagem, os autores adicionaram ruído do tipo Gaussiano com diferentes níveis de intensidades em imagens de pragas agrícolas. A partir da imagem original, os autores dividiram essa imagem em quatro quadrantes, onde para cada quadrante foi adicionado ruído Gaussiano com nível de intensidade diferente. A partir da imagem ruidosa esses autores aplicaram os filtro passa-baixa e o filtro passa-alta, com diferentes frequências de corte. Essa abordagem possibilitou a observação de imagens de pragas do milho som sinais de ruído aleatório para o seu processamento.

1.3.3 Abordagem baseada em segmentação de imagens

Com base nas palavras-chave: *image segmetation*, *color spaces segmentation*, *Otsu segmentation* e *threshold segmetation*, houve o levantamento em estado da arte no tema segmentação de imagens.

Com o objetivo de identificar pragas e doenças da cultura do algodão, He e colaboradores propuseram um estudo para comparação dos seguintes padrões de cores: *Red*, *Green*, *Blue* (RGB); *Hue*, *Saturation*, *Intensity* (HSI) e YCbCr. A equalização do histograma e filtros lineares espaciais foram aplicados para eliminação de áreas que não eram de interesse e eliminação de ruídos, respectivamente (HE et al., 2013). Após esses processos os autores identificaram, dentre os padrões de cores selecionados para a pesquisa, quais componentes sofreram menor efeito causado por ruídos e iluminação para o processo de segmentação. No padrão de cor RGB, destacou-se o componente G, já no padrão de cor HSI, destacou-se o componente I e, por fim, no padrão de cor YCbCr, houve destaque no componente Y. Como resultado, apresentaram uma folha atacada por uma doença e outra por uma determinada praga. O modelo proposto mostrou que o padrão de cor RGB foi capaz de identificar que a folha atacada pela praga teve um dano de 5,29% e a folha atacada pela doença 81,60% de sua área total. O padrão de cor HSI identificou um dano de 4,90% e 43,15% nas folhas atacadas pela praga e pela doença, respectivamente. Já o padrão de cor YCbCr, identificou 5,93% e 58,40% da área total da folha atacada pela praga e pela doença respectivamente. Desta forma, os autores concluíram que o modelo proposto foi capaz de detectar os danos cousados tanto pela praga quanto pela doença.

Para a detecção de pequenos insetos-praga em imagens de baixa resolução, Xia e colaboradores propuseram segmentar imagens por meio do método *watershed* (XIA et al., 2015). Para a realização dos experimentos, os autores utilizaram imagens dos seguintes insetos-praga: mosca branca (*Bemisa tabaci Glover*), piolho-do-algodão (*Aphis gossypii Glover*) e tripés do fumo (*Thrips tabaci L.*). Inicialmente as imagens foram convertidas para o padrão de cor YCbCr e então segmentadas, visando extrair o componente Cb de cada imagem. Cada inseto-praga de cada

imagem foi então representado pela média de cor no padrão HSV. Os resultados se mostraram satisfatórios nas imagens de baixa resolução, principalmente nas imagens com resolução de 640x430 *pixel*. Já nas imagens de alta resolução, ocorreram os chamados falso-positivos causados pela super segmentação.

Kumar e colaboradores utilizaram a técnica de segmentação de imagens conhecida como limiarização adaptativa para a detecção e contagem de pragas. Esse método consistiu em computar o limiar de cada *pixel* da imagem por interpolação dos resultados das sub imagens (KUMAR et al., 2017). Para validação do método foram processadas 500 imagens. À essas imagens foi aplicado o filtro da mediana para eliminação de ruídos e então as mesmas foram convertidas para escala de tons de cinza. A limiarização adaptativa é foi então aplicada *pixel a pixel* às imagens, segmentando assim os objetos (pragas) individualmente. Os autores obtiveram acurácia entre 95% à 97% inclusive.

Sriwastawa e colaboradores compararam a segmentação baseada em cores (SRIWASTWA et al., 2018) com os métodos da segmentação de Otsu (OTSU, 1979) e de detecção de bordas. As imagens utilizadas foram, primeiramente, convertidas em escala de tons de cinza. Foi então aplicado o filtro do tipo *Top Hat*, para problemas de iluminação, além de filtros para ajustes de contraste. Os experimentos foram inicialmente realizados com a praga pirila (*Pyrrilla*), encontrada no cultivo da cana-de-açúcar (*Saccharum officinarum*). Posteriormente os mesmos métodos foram aplicados às imagens de cupins (*Isoptera*), encontrados no cultivo do milho. A segmentação de Otsu e por detecção de bordas foram então aplicadas nas imagens e comparadas. Esses métodos apresentaram problemas quando existiam sombras dos insetos nas imagens. As sombras eram detectadas como se fossem outros insetos. Já para a segmentação baseada em cores, as imagens foram convertidas para o espaço de cor L^*a^*b (LUO; HUNT, 1997) e segmentadas por meio do algoritmo de agrupamento *K-means* (MACQUEEN, 1967). Os resultados mostraram uma melhor eficiência, já que o algoritmo *K-means* segrega as cores de maneira eficiente.

1.3.4 Abordagem baseada em extração de características

Com base nas palavras-chave: *image feature extraction*, *HOG features* e *Hu invariant moments*, houve o levantamento em estado da arte no tema extração de características.

Para a classificação de doenças em folhas de orquídeas, aplicou uma rede neural artificial com algoritmo *backpropagation* para a classificação das doenças podridão mole (BSR), mancha marrom (BBS) e podridão negra (PBR) (HUANG, 2007). As imagens foram obtidas no espaço de cor RGB e de dimensão 640x480 *pixels*. As características de cor e de texturas da área da lesão ocasionada pelas doenças foram extraídas por meio de uma matriz de co-ocorrência (HARALICK et al., 1973). Como características de cor, foram extraídas as médias de cada valor R (vermelho), G (verde) e B (azul). Como características de textura foram extraídas a área, perímetro e diâmetro médio, contraste, uniformidade, probabilidade máxima, homogeneidade, diferença da inversa, diferença da variância, diferença da variância diagonal, entropia e a diferença da entropia das

bandas G e B da área lesionada para avaliar a textura da área da lesão da folha. Para a montagem da matriz de coocorrência foi utilizada a distância de 1 *pixel* e os ângulos de 0°, 45°, 90° e 135°. As classes de saídas foram definidas como plantas saudáveis, BSR, BBS e PBR. Para validação do método proposto, foram utilizadas 145 imagens de plantas doentes e 40 imagens de plantas saudáveis. Dessa maneira, a rede neural artificial classificou as doenças citadas com acurácia de 97,2%, obtendo assim, resultado considerado satisfatório de acordo com o autor.

Sette e Maillard propuseram um método baseado em análise de textura de imagens georreferenciadas para monitoramento de uma determinada região da Mata Atlântica localizada no Estado da Bahia, a partir de imagens analisadas no espectro visível (RGB) (SETTE; MAILLARD, 2011). Métricas como o contraste, entropia, correlação, momento de diferença inversa e segundo momento angular foram extraídas a partir da matriz de co-ocorrência (HARALICK et al., 1973). Para a classificação desses dados foi utilizado o algoritmo de classificação Máxima Verossimilhança (VICKERS; MODESTINO, 1982), que apresentou uma acurácia de 46,4% na classificação de imagens no espectro visível, ou seja, analisando somente as componentes R, G e B. Já a classificação de imagens por meio das características de textura obteve acurácia de 55,6% e a classificação de imagens por meio da adição das características de textura e espectrais obteve acurácia de 83,9%.

1.3.5 Abordagem baseada em aprendizado de máquina

Com base nas palavras-chave: *machine learning pests image classification, supervision classification e SVM kernels comparisons*, houve o levantamento em estado da arte no tema aprendizado de máquina.

Ahmed e colaboradores (AHMED et al., 2008) desenvolveram um sistema de visão de máquina em tempo real para reconhecimento e controle de plantas invasoras. O sistema proposto também tinha o objetivo de reconhecer e classificar as plantas invasoras nas classes folhas largas e folhas estreitas. Essa classificação se deve ao fato de que existem herbicidas específicos para cada tipo de planta invasora. O sistema baseou-se na medição da densidade das plantas por meio de operações de máscara. As imagens a serem analisadas eram decompostas em canais de cores (vermelho, verde e azul) e binarizada onde os *pixels* pertencentes às plantas invasoras correspondessem aos *pixels* brancos e os *pixels* correspondentes ao fundo fossem pretos. Dessa forma, a porcentagem de plantas invasoras foi calculada, assim como a densidade das mesmas. Posteriormente foi realizado o processo de classificação, as imagens foram classificadas em “plantas invasoras de folhas estreitas”, “mistura de plantas invasoras” e “plantas invasoras de folhas largas”. De acordo com os resultados apresentados pelos autores, o sistema obteve uma taxa de acerto de 99% para imagens sem ou com plantas invasoras e de 95% para plantas invasoras de folhas estreitas e largas. Assim os autores concluíram a eficiência do sistema proposto.

Guerrero e colaboradores propuseram um método baseado no classificador SVM para

identificação de ervas daninha na cultura do milho (GUERRERO et al., 2012). De acordo com os autores, o componente G (verde) do padrão de cor RGB está associado às folhas das plantas de milho, enquanto o componente R (vermelho) está associado aos materiais presentes no solo como água, ervas daninha e palhas secas das plantas do milho. Para o processo de classificação, utilizaram classificadores SVM com funções *kernel* polinomial, RBF e sigmoide. Ainda de acordo com os autores, os resultados dos experimentos mostraram melhor acurácia do classificador SVM com função *kernel* RBF, com acurácia de 84,2%. O classificador com função *kernel* polinomial apresentou acurácia de 79,1%, seguido pelo classificador com função *kernel* sigmoide, com acurácia de 74%.

Para a classificação de diferentes folhas, Lee e colaboradores utilizaram uma rede neural convolucional (CNN) baseada na rede neural AlexNet (KRIZHEVSKY et al., 2012), e uma rede deconvolucional para observação da transformação das características das folhas (LEE et al., 2015). Para o treinamento da segunda CNN foram realizados procedimentos como extração do fundo da imagem, rotação da imagem em diferentes ângulos e seleção manual da região interna da folha. Para treinamento da primeira CNN foram utilizadas 2288 imagens e 528 para testes, atingindo assim, acurácia de 97,7%. A segunda CNN foi treinada com 34672 regiões das imagens e testada com 8800, atingindo acurácia de 99,5%

Uma metodologia baseada em técnicas de aprendizado de máquina para a detecção de pragas e doenças do cultivo do tomate foi proposta por Fuentes e colaboradores (FUENTES et al., 2017). A metodologia proposta se dividiu em quatro etapas: *Data Collection*, *Data Annotation*, *Data Augmentation* e *Disease and Pest Detection*. Na etapa *Data Collection*, foram coletadas imagens de diferentes resoluções contendo diferentes pragas e doenças da cultura do tomate, em seus diferentes estágios de evolução. Na etapa *Data Annotation*, as imagens foram rotuladas por especialistas. Na etapa *Data Augmentation*, foram utilizadas técnicas para aumento do conjunto de imagens. Essas técnicas consistiram em transformações geométricas, como tamanho e rotação por exemplo, e de intensidade, como brilho, contraste, cor e ruídos. A etapa de *Disease and Pest Detection*, teve como objetivo além de detectar pragas e doenças, identificar sua localização na planta (folha, em qual dado da folha, galho etc.). Dessa forma, foram utilizados três modelos de redes neurais para executarem essa tarefa. Para reconhecimento das pragas e doenças (objetos de interesse) e da sua localização na planta, foram utilizadas *Faster Region-based Convolutional Neural Network* (F-CNN) e *Region-based Fully Convolutional Neural Network* (R-FCN). Com o objetivo alcançado, os autores utilizaram *Single Shot Multibox Detector* (SSD) para classificar os objetos de diferentes tamanhos. Para a validação da metodologia, esses autores trabalharam com cerca de cinco mil imagens, sendo que deste total, 80% foi utilizada na fase de treinamento, 10% para testes e 10% para validação das redes neurais.

Evangelista em 2019 apresentou um estudo para classificar moscas e mosquitos por meio da frequência do batimento de suas asas (EVANGELISTA, 2019). Um classificador Bayesiano (JOHN; LANGLEY, 1995) foi utilizado para classificar os insetos em duas classes, ‘moca

caseira' e 'mosquito'. Em seus experimentos, o autor separou moscas e mosquitos em recipientes distintos, cada um deles utilizando um sensor infravermelho para a captação da frequência das batidas das asas desses insetos. Foi aplicada a Transformada Rápida de Fourier (BERGLAND, 1969) para a representação das frequências dos sinais captados. A partir dos dados coletados foi possível verificar que os batimentos de asas das moscas variaram entre 200 Hz e 340 Hz, enquanto a frequência dos batimentos das asas dos mosquitos variam entre 350 Hz e 800 Hz. A partir desses dados foram construídos histogramas para que se pudesse identificar um limiar. Foram identificados os limiares de 100 Hz, 168 Hz, 340 Hz e 800 Hz. Frequências abaixo de 180 Hz foram consideradas inválidas por identificarem tanto moscas como mosquitos e serem pouco representativas. Frequências entre 180 Hz e 340 Hz foram classificadas como 'mosca caseira'. Já as frequências entre 340 Hz e 800 Hz foram classificadas como 'mosquito'. Ao final dos experimentos o classificador possibilitou obter uma acurácia de 91,97% na classificação de moscas caseiras e 92,85% na classificação de mosquitos.

Com o objetivo de comparar a eficiência do uso de classificadores SVM com parametrização otimizada em relação ao uso desses mesmos classificadores, porém, com parametrização *default*, Grama e colaboradores (GRAMA et al., 2017) apresentaram um estudo para otimização e parametrização de classificadores SVM com diferentes funções *kernel*. Como validação do estudo, os autores utilizaram um banco de dados contendo sons de armas de fogo, pássaros, motosserra, vozes humanas e tratores. Como funções *kernel* utilizaram função linear, RBF e sigmoide. Para otimização da classificação, os autores também utilizaram validação cruzada. De acordo com os autores, os resultados mostraram que a correta parametrização dos classificadores com suas respectivas funções *kernel* apresentaram significativo aumento de performance e acurácia em relação ao uso de parâmetros *default*, sendo que o classificador SVM com função *kernel* sigmoide obteve o melhor índice de performance.

Nanda e colaboradores propuseram um método para detectar cupins por meio de classificadores SVM (NANDA et al., 2018). Como experimentos, os autores utilizaram pedaços de madeira e previamente separaram em duas classes, infestado e não infestado. Dessa forma, utilizando microfones os autores capturaram sons dos cupins e como características para identificar a infestação utilizaram entropia e energia. Para validação do método, os autores realizaram experimentos com classificadores SVM com funções *kernel* linear, RBF, polinomial e sigmoide. De acordo com os autores a classificação por meio do uso da função *kernel* polinomial apresentou a melhor taxa de acurácia e menor taxa de erro, 91,88% e 12%, respectivamente. Os classificadores com funções *kernel* RBF, linear e sigmoide apresentaram, respectivamente, as seguintes taxas de acurácia e erro: 91,48% e 15%; 89,56% e 17%; 88,72% e 16%.

Liu e colaboradores apresentaram uma metodologia de extração de dados de camadas intermediárias de redes neurais convolucionais com o propósito de usar esses dados para treinamento de um classificador e assim torná-lo mais robusto (LIU et al., 2019). Segundo os autores, características extraídas de camadas intermediárias de uma rede neural convolucional foram

representativas e puderam melhorar significativamente a acurácia do classificador. Para testar a eficácia do método, os autores utilizaram as seguintes CNNs para extração de características: AlexNet, VggNet e ResNet (KRIZHEVSKY et al., 2012). As características extraídas foram utilizadas para treinamento dos seguintes classificadores: *Support Vector Machine* (SVM) (HARRALICK et al., 1973), *Naive Bayes* (JOHN; LANGLEY, 1995), *Linear Discriminant Analysis* (LDA) e *Decision Tree* (RIVEST, 1987). As imagens utilizadas foram adquiridas dos bancos de imagens: Caltech-10, Cifar-10 e Stl-10. Para a realização dos experimentos, os autores dividiram seu conjunto de imagem da seguinte maneira: 70% das imagens para treinamento e 30% para teste. Para cada classificador foram testadas cada CNN para cada banco de imagens. Os resultados mostraram que as características extraídas das camadas intermediárias das CNNs foram informativas e puderam ajudar no desempenho e na classificação final. Esse método desenvolvido mostrou ser mais eficiente e mais escalável que métodos tradicionais

Li e colaboradores apresentaram uma metodologia para classificação de grãos de milho por meio de técnicas de visão computacional e aprendizado de máquina (LI et al., 2019). Um estimador de máxima verossimilhança (VICKERS; MODESTINO, 1982) foi treinado para classificar os grãos de milho em ‘normal’ e ‘danificado’. Esse último com sete subclassificações. No processo de segmentação foi aplicado o método de binarização de Otsu (OTSU, 1979), para separar os grãos do fundo da imagem. Posteriormente, foi aplicado o algoritmo *line profile-based segmentation* (LPSA) (MAHDI; QIN, 2017), para separar os grãos que estivessem, de alguma maneira, sobrepostos ou em contato com outros e que pudessem ser confundidos como um único grão. As características de forma, como por exemplo, perímetro, circunferência, alongamento e cor foram consideradas para o processo de extração de características. Nesse processo as imagens foram trabalhadas no espaço de cor RGB e HSI. Considerando que das sete subclassificações de grãos danificados, três possuíam características semelhantes (danos por aquecimento, danos no gérmen e podridão), elas foram consideradas como uma única característica em um primeiro momento. No primeiro experimento foram separados 400 grãos, sendo que 280 para treinamento e os 120 restantes, para testes. Os resultados apresentados pelos autores mostraram que a acurácia do classificador foi de 96,67%. Em um segundo experimento, foram consideradas as sete subclassificações que representam os grãos danificados. Nesse experimento foram utilizados 490 grãos para a etapa de treinamento e 210 para a etapa de testes. Desta vez os resultados apresentados mostraram uma acurácia de 67,48%. Essa diminuição no desempenho ocorreu devido às semelhanças entre as subclasses dos grãos danificados. Por fim, os autores concluíram que o resultado não foi satisfatório para a classificação do grão de milho quando se considerou sete categorias de grãos de milho danificados. Porém, quando tratado como apenas uma categoria, o método se mostrou robusto e com acurácia satisfatória.

Em 2019 Abdelghafour e colaboradores apresentaram um *framework* para classificação da cobertura de parreiras em seus diferentes estágios fenológicos, ou seja, folhagem, pedúnculo e fruto (ABDELGHAFOUR et al., 2019). A estrutura do *framework* apresentada foi dividida em quatro etapas. A primeira etapa consistiu na aquisição das imagens e pré-processamentos para

obtenção dos objetos de interesse. Durante a obtenção das imagens, foi utilizado um *flash* com período curto de exposição, porém, com alto poder de iluminação, resultando em uma imagem com a folhagem brilhante e um fundo escuro. Dessa maneira, para a remoção do fundo da imagem, que incluía objetos sem interesse ao objetivo, as imagens foram convertidas para o espaço de cor HSV. Para cada componente de cor, foi construído um histograma permitindo assim, que se chegasse a um limiar para cada componente. Esse limiar foi utilizado para a separação dos objetos de interesse (folhas, frutos, entre outros) dos demais objetos presentes nas imagens. A segunda etapa consistiu na extração de características. Nessa etapa esses autores utilizaram um extrator de características geométricas (formas e texturas), *Local Structure Tensor*. Informações de cores também foram extraídas no espaço de cor RGB. A terceira etapa consistiu na classificação das informações extraídas dos objetos de interesse. Para essa tarefa, os autores utilizaram um classificador Bayesiano, aplicando também um estimador probabilístico *Maximum a Posteriori* (MAP). A quarta e última etapa do *framework* consistiu na etapa de pós-processamento e regularização da estimativa do classificador. De acordo com os autores, essa última etapa se fez necessária devido ao fato de que, no *framework*, o estimado MAP foi configurado para atuar individualmente em cada *pixel*, sem considerar os valores dos *pixel* vizinhos. Dessa forma, esses autores melhoraram a acurácia do *framework*, levando à resultados de 85% e podendo atingir 95% de sucesso na classificação da cobertura de parreiras em seus diferentes estágios de desenvolvimento.

1.4 Objetivo geral

Esse trabalho tem como objetivo geral apresentar um método baseado em uso de técnicas de PIS e aprendizado de máquina para a classificação de padrões da Lagarta do cartucho (*Spodoptera frugiperda*) na cultura do milho.

1.5 Objetivos específicos

Para se alcançar o objetivo geral deste trabalho, os seguintes objetivos específicos foram considerados:

- Levantar padrões para identificação de cada estágio da praga selecionada;
- Comparar o sistema de representação de cores RGB com sistemas pouco sensíveis à luminância, como por exemplo, HSV e CIE L*a*b*, para o refinamento do método de segmentação e reconhecimento dos padrões;
- Analisar, comparar e aplicar diferentes técnicas de processamento de imagens digitais, visando a organização de descritores para o reconhecimento dos padrões e a classificação;

- Estabelecer técnica de aprendizado de máquina para a identificação da presença e classificação dos principais estágios de desenvolvimento da Lagarta do Cartucho (*Spodoptera frugiperda*) em cultura do milho, com adequado nível de acurácia e precisão.

1.6 Organização do documento

Este documento está estruturado em cinco Capítulos, sendo este primeiro dedicado à introdução, revisão da literatura e principalmente a motivação para o desenvolvimento desta pesquisa; o Capítulo 2 que apresenta materiais e métodos, incluindo a fundamentação teórica sobre as técnicas utilizadas; o Capítulo 3 que apresenta resultados e discussões; e finalmente, o Capítulo 4 que apresenta as conclusões e sugestões de trabalhos futuro.

Capítulo 2

MATERIAIS E MÉTODOS

Este Capítulo tem por objetivo apresentar a estruturação do método para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho. Desta forma, são apresentados a base de imagens gerada a partir de consultas em banco de imagens, assim como a técnica utilizada para aumento do banco de imagens, técnica conhecida como *data augmentation*. São também apresentados a fundamentação teórica e os pseudocódigos desenvolvidos com base nas técnicas utilizadas, tais como, processo de redimensionamento de imagens, filtragem de ruídos, operações de padrões de cores, segmentação de imagens, reconhecimento de padrões e aprendizagem de máquina.

2.1 Abordagem metodológica e principais materiais

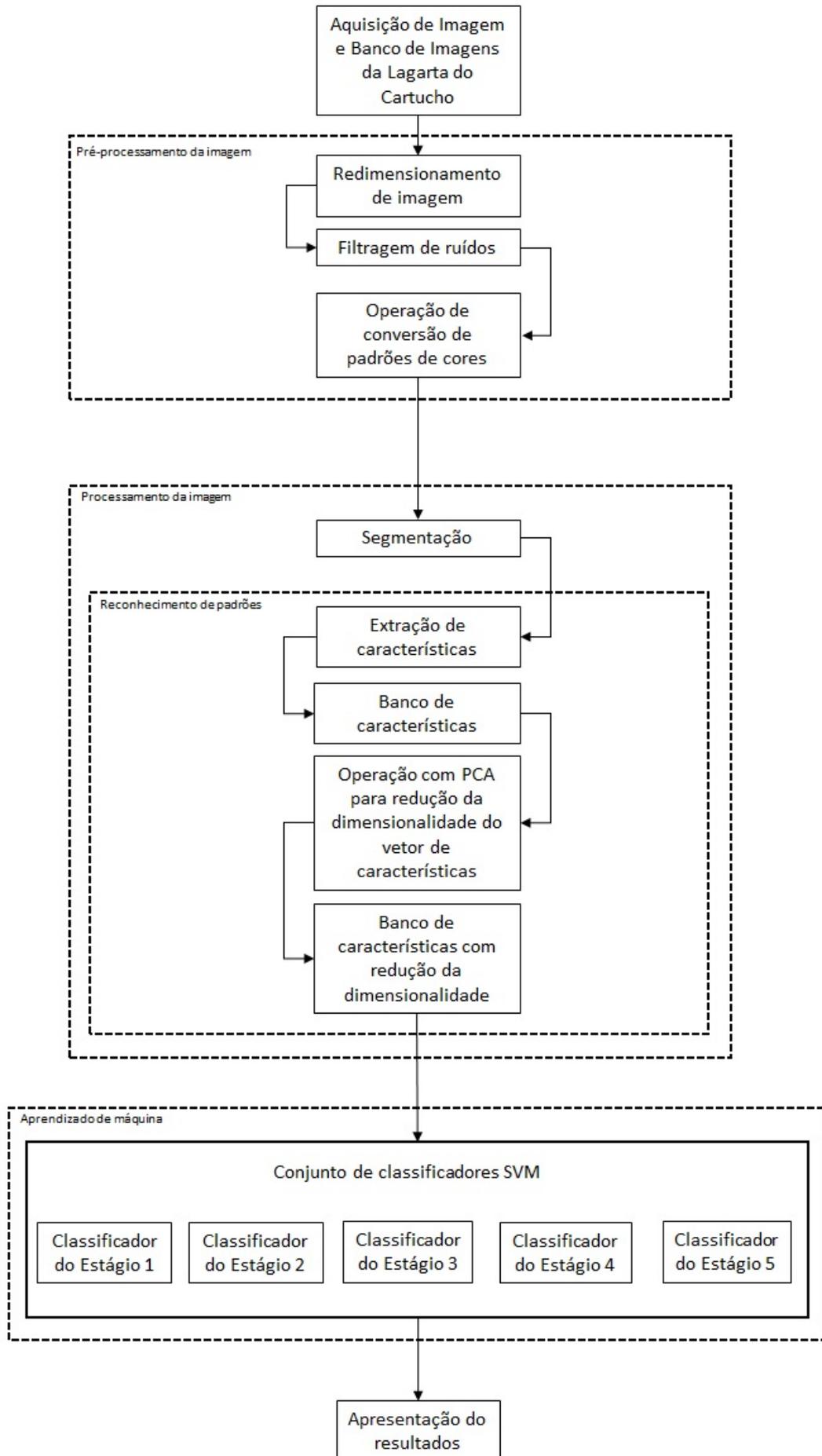
A Figura 4 ilustra em diagrama de blocos a organização estabelecida para o método de classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*).

Quanto a aquisição e banco de imagens da Lagarta do Cartucho em cultura do milho, existem diferentes maneiras para se obter uma imagem digital, porém, a mais direta é por meio de uma câmera digital. Câmeras digitais possuem sistemas de captura de quadros integrados e a conversão analógico-digital é feita no próprio sensor. Esses sensores são conhecidos como Dispositivos com Acoplamento por Carga (CCD – *Charged-Coupled Device*) ou Semicondutor de Metál-Óxido Complementar (CMOS – *Complementary Metal-Oxide Semiconductor*) (SOLOMON; BRECKON, 2000).

A primeira fotografia digital que se tem conhecimento data de 1957 e foi obtida por meio de um *scanner* desenvolvido por Russel Kirsch (KIRSCH et al., 1957; KIRSCH, 1998). Kirsch conseguiu gerar uma imagem digital formada por uma matriz quadrada de 176 *pixels*.

A convenção criada por Kirsch é utilizada até os dias atuais e define uma imagem digital como uma função bidimensional $f(x, y)$ de intensidade de luz nas posições (x, y) , conhecidas como *pixel* (do inglês *picture element*) ou *pel* (SIEBRA, 2013). Uma imagem digital pode ser representada tanto na forma de uma matriz de duas dimensões (2D), como na forma de uma

Figura 4 – Diagrama em blocos do método para classificação de padrões de da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho.



matriz de três dimensões (3D), podendo ser em escala de tons de cinza, em pseudo-cores ou também em cores reais.

Imagens digitais podem ser obtidas em diferentes formatos, tipos ou padrões de cores. A Tabela 4 apresenta os formatos de imagens digitais mais comuns e utilizados hoje em dia e suas características.

Tabela 4 – Principais formatos de imagens digitais (Fonte: Adaptado de Solomon e Breckon (2000))

Formato	Denominação	Propriedades
GIF	Graphics Interchange Format – Formato para intercâmbio de imagens	Limitado a apenas 265 cores (8 bits). Compressão com perda
JPEG/JPG	Joint Photographic Experts Group – Grupo Misto de Especialistas Fotográficos	Formato de uso mais comum. Compressão com perda, porém, existe variante sem perda
BMP	Bit Map Picture – Imagem de Mapa de Bit	Formato básico de imagem. Compressão sem perda. Existem variantes com perda
PNG	Portable Network Graphics – Imagens Portáteis para Redes	Projetado para substituir o GIF. Compressão sem perda
TIF/TIFF	Tagged Image (File) Format – Formato (de Arquivo) de Imagem com Rótulo	Formato muito flexível, detalhado e adaptável. Existem variantes comprimidas/descomprimidas

As imagens utilizadas para o desenvolvimento da pesquisa compõem o banco de imagens *Insect Images*¹. A escolha dessas imagens se caracteriza, principalmente, pela qualidade e diversidade, assim como sua disponibilidade para o uso na pesquisa.

O banco de imagens *Insect Images* é um subgrupo do sistema de banco de imagens online *Bugwood Image Database System*². Atualmente o *Bugwood Image Database System* é composto de mais de 300 mil imagens separadas em mais de 27 mil subgrupos e com contribuições de mais de 2.500 pesquisadores e fotógrafos. Além do *Insect Images*, outros bancos de imagens que compõem o *Bugwood Image Database System* são: *Forestry Images*³, *IPM Images*⁴, *Weed Images*⁵ e *Invasive.org*⁶.

¹ O banco de imagens *Insect Images* é um projeto conjunto das instituições *The University of Georgia, Warnell School of Forestry, Natural Resources and College of Agricultural and Environmental Sciences, The Center for Invasive and Ecosystem Health, USDA National Institute of Food and Agriculture, Southern Integrated Pest Management Center, Southern Plant Diagnostic Network* e *USDA Identification Technology Program*. Disponível em: <<https://www.insectimages.org>>

² Disponível em: <<https://images.bugwood.org/>>

³ Disponível em: <<https://www.forestryimages.org/>>

⁴ Disponível em: <<https://www.ipmimages.org/>>

⁵ Disponível em: <<https://www.weedimages.org/>>

⁶ Disponível em: <<https://www.invasive.org/>>

Outro fator de importância para a utilização desse banco de imagens é que em sua grande maioria, as imagens foram capturadas em campo, ou seja, sofreram influência de iluminação não controlada, possuem variações de escala e tamanho, entre outras influências decorrentes da aquisição em ambiente real.

A Tabela 5 exibe as principais informações de caracterização dessas imagens que estão disponíveis no referido banco de imagens e que foram utilizadas para a validação do método desenvolvido.

Tabela 5 – Características das imagens da Lagarta do Cartucho obtidas pelo banco de imagens *Insect Image*

Tipo do arquivo	<i>JPG/JPEG</i>
Espaço de cor	<i>RGB</i>
Largura da imagem	<i>3072 pixels</i>
Altura da imagem	<i>2048 pixels</i>
Resolução	<i>72 ppi (pixels per inch)</i>
Tamanho do pixel	<i>≈ 0,35mm</i>

A infraestrutura computacional utilizada para o desenvolvimento do método foi composta por um computador com processador *intel i9* e 16GB de memória RAM e placa de vídeo dedicada *NVIDIA GeForce GTX 1660* com 6 *GBytes* de memória dedicada. O código do sistema computacional foi desenvolvido em linguagem de programação *Python 3.7*⁷. Foi utilizado como ferramenta de desenvolvimento *PyCharm Community Edition*⁸, última atualização 2021.2. Foram utilizadas as bibliotecas de processamento de imagens, aprendizado de máquina, manipulação e visualização de dados em *Python*, como: *OpenCV*⁹ e *scikit-image*¹⁰, *scikit-learn*¹¹, *pandas*¹², *seaborn*¹³ e *matplotlib*¹⁴.

Como técnica adicional para aquisição de imagens, utilizou-se neste trabalho a técnica de *data augmentation*. No que se refere ao processamento de imagens digitais, *Data augmentation* pode ser descrito como um processo de manipulação e geração de imagens por meio de transformações morfológicas e quando utilizada na etapa de treinamento de classificadores, permite uma melhor generalização e previne *overfitting* (MIKOŁAJCZYK; GROCHOWSKI, 2018).

As operações mais comuns em algoritmos de *data augmentation* são rotação, translação, sobreposição de imagens, cisalhamento, corte (*cropping*), conversão de padrões de cores, adição de ruídos, entre outras operações (SHORTEN; KHOSHGOFTAAR, 2019). Operações de conversão de padrões de cores e adição de ruídos serão discutidas nas próximas seções.

⁷ Disponível em: <www.python.org>

⁸ Disponível em: <www.jetbrains.com/pycharm/>

⁹ Disponível em: <<https://opencv.org/>>

¹⁰ Disponível em: <<https://scikit-image.org/>>

¹¹ Disponível em: <<https://scikit-learn.org/>>

¹² Disponível em: <<https://pandas.pydata.org/>>

¹³ Disponível em: <<https://seaborn.pydata.org/>>

¹⁴ Disponível em: <<https://matplotlib.org/>>

Em operações de rotação, como o próprio nome sugere, é possível rotacionar uma imagem tanto para a esquerda quanto para a direita a partir de um ângulo θ entre 1 e 159 graus. Entretanto, em operações cuja tarefa é o reconhecimento de dígitos (números) são sugeridos ângulos entre 1 e 20 graus, ou então, ângulos entre -1 e -20 graus. Uma vez que à medida que o ângulo de rotação aumenta, o rótulo dos dados não é preservado (CLARO et al., 2020). Um bom exemplo desse problema é a operação de rotação do dígito 6 em um ângulo de maior que 90 graus, passando a ser confundido com o dígito 9. O mesmo ocorre para o dígito 9.

Em operações de translação é possível a alteração da posição da imagem, à direita, à esquerda, para cima ou para baixo, por exemplo. Esse método é muito apropriado durante a etapa de treinamento de redes neurais convolucionais, uma vez que força a rede neural a procurar objetos em todos os pontos de uma imagem.

Operações de cisalhamento permitem a rotação de um dos eixos da imagem, porém, é diferente da operação de rotação. Enquanto a operação de rotação permite a mudança dos eixos (x, y) de uma imagem, a operação de cisalhamento fixa um dos eixos, por exemplo o eixo x , e rotaciona o outro eixo, por exemplo o eixo y , resultando em uma imagem com uma perspectiva de inclinação.

Operações de rotação, translação e cisalhamento podem ser compreendidas como transformações de forma de uma imagem. A forma de uma imagem pode ser definida como conjuntos ordenados de N duplas de coordenadas euclidianas $(x_1, y_1) \dots (x_N, y_N)$ são descritas como colunas de uma matriz S (SOLOMON; BRECKON, 2000), conforme a Equação 2.1:

$$S = \begin{bmatrix} x_1 & x_3 & x_2 & \cdots & x_N \\ y_1 & y_3 & y_2 & \cdots & y_N \end{bmatrix} \quad (2.1)$$

Dessa forma, o resultado da operação de uma matriz de transformação T à matriz S , pode ser obtido pela operação de multiplicação de matrizes, de acordo com a Equação 2.2.

$$S' = TS \quad (2.2)$$

onde S' corresponde ao novo conjunto de coordenadas. Entretanto, a exceção seria a operação de translação, que requer uma operação de adição vetorial.

A Equação 2.3 expressa a operação de rotação, enquanto as Equações 2.4 e 2.5 expressam a operação de cisalhamento no eixo x e y de uma imagem, respectivamente.

$$S' = T_\theta S = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.3)$$

onde θ corresponde ao ângulo de rotação.

$$S' = T_x S = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.4)$$

$$S' = T_y S = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.5)$$

onde α corresponde ao fator de cisalhamento.

A operação de translação é expressa pela Equação 2.6:

$$S' = T_{(x,y)} + S = \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.6)$$

onde α corresponde ao fator de translação.

A operação de corte, aleatório ou não, pode ser executada de maneira a obter resultados distintos. Assim a operação de corte pode ser executada de modo a remover uma determinada região da imagem. Por outro lado, a operação de corte pode ser executada com o interesse de se obter uma sub-imagem a partir da imagem original, como pode ser observado na Figura 5.

Existem algoritmos específicos, principalmente no se refere a redes neurais convolucionais, para o uso de *data augmentation* (FAWZI et al., 2016; CLARO et al., 2020) no processamento de imagens digitais. Porém, conforme descrito acima, não se faz necessário o uso de técnicas complexas para o aumento do banco de imagens neste trabalho.

Assim, o uso da técnica de *data augmentation* utilizado neste trabalho e ilustrado pela Figura 5 é baseado nos critérios onde já existe o conhecimento prévio de qual estágio de desenvolvimento da Lagarta do Cartucho a imagem representa, locais onde a Lagarta do Cartucho seria encontrada na planta, tais como o colmo, folha, cartucho e a espiga, permitindo assim, o uso da técnica de sobreposição de imagens. Por fim, o uso da técnica de rotação aleatória da imagem referente a Lagarta do Cartucho e da técnica de corte (*cropping*), permitem a geração de sub-imagens.

O pseudocódigo 1 representa processo de *data augmentation* ilustrado pela Figura 5.

Algorithm 1 Pseudocódigo para aumento do banco de imagens da Lagarta do Cartucho

Entrada: Imagem da lagarta (I_l), Imagem de fundo (I_f), lista graus rotação (L_g)

1 **início**

2 **para** cada subimagem(I_f) **faça**

3 n = randon(tamanho(L_g)-1)

4 I_{lr} = rotacionar(I_l , n)

5 I_s = sobrepor(I_f , I_{lr})

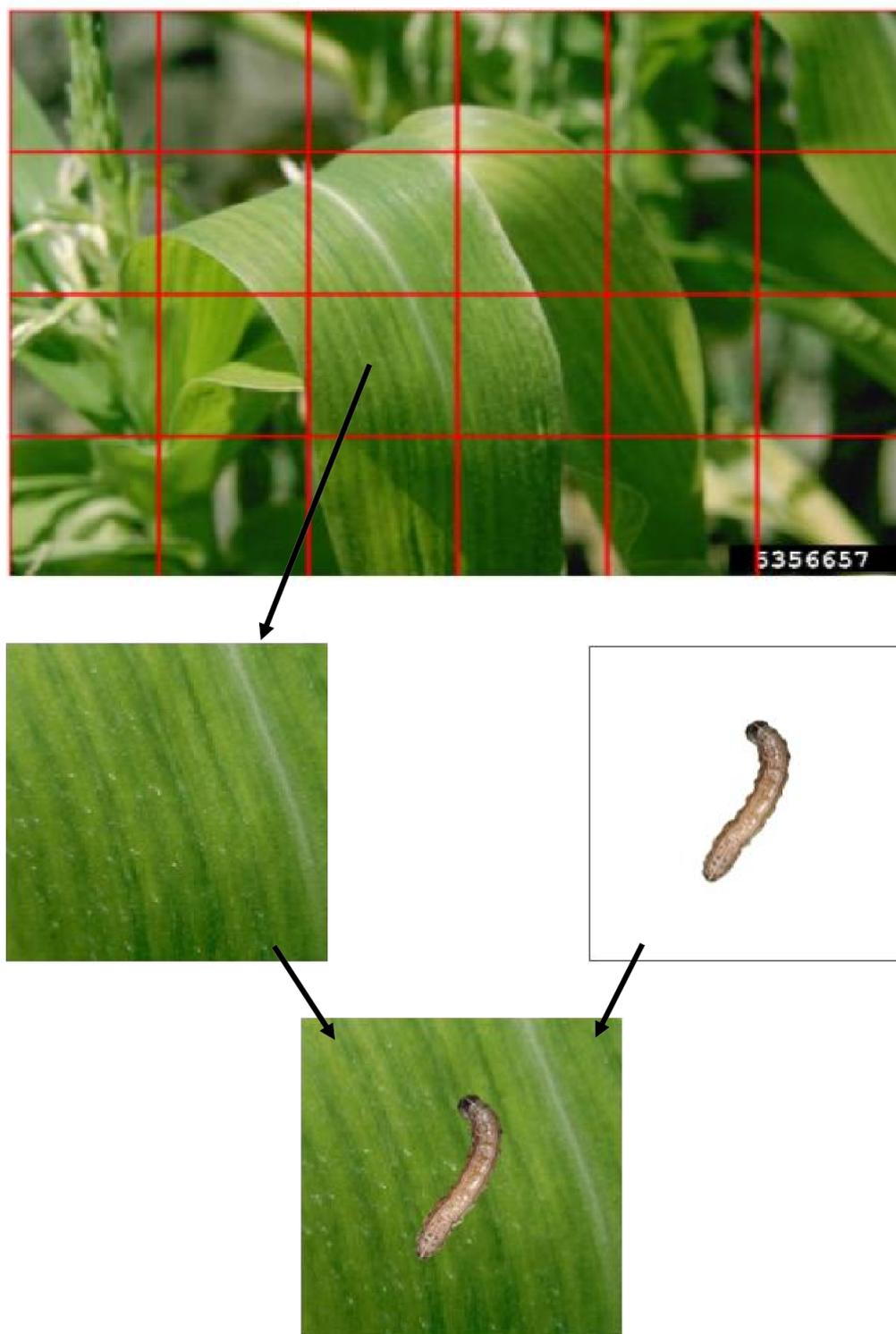
6 **fim**

7 **fim**

Saída: Sub imagem do fundo com a imagem da lagarta sobreposta e rotacionada (I_s)

Também se faz necessário incluir incluir uma etapa de redimensionamento de forma a garantir uma padronização na dimensionalidade das imagens, principalmente por conta da etapa

Figura 5 – Processo de aumento do banco de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*).



de extração de características, onde o tamanho do vetor de características obtido por meio da aplicação do descritor HOG é influenciado diretamente pela dimensionalidade da imagem.

O pseudocódigo 2 ilustra o processo de redimensionamento das imagens da Lagarta do Cartucho em cultura do milho.

Algorithm 2 Pseudocódigo para redimensionamento de imagens

Entrada: Imagem original I_o , dimensões(x, y)

8 **início**

9 | $I_r = \text{redimensionar}(I_o, \text{dimensões}(x, y))$

10 **fim**

Saída: Imagem redimensionada (I_r)

2.2 Pré-processamento das imagens da Lagarta do Cartucho em cultura do milho

Quanto à filtragem de ruídos das imagens adquiridas, foram avaliados considerando a presença de ruídos, principalmente por ruídos do tipo aleatório, como por exemplo, ruído do tipo Gaussiano ou ruído do tipo impulsivo (ABDURRAZZAQ et al., 2020). No conceito do processamento imagens digitais, ruídos podem ser definidos como qualquer mudança no sinal que acarrete degradação ou perda de informação do sinal original, podendo ser causado por condições de iluminação da cena e ou objeto, temperatura do sensor de captura do sinal durante a aquisição da imagem, transmissão da imagem, entre outros fatores (BERTOLLA; CRUVINEL, 2021).

Conforme acima descrito, ruídos do tipo aditivo normalmente ocorrem por conta da temperatura do sensor de captura da imagem e também por influência das condições de iluminação. Esse tipo de ruído pode ser representado pela Equação 2.7:

$$f'(x, y) = \eta(x, y) + f(x, y) \quad (2.7)$$

onde $f(x, y)$ representa a imagem original, $\eta(x, y)$ o ruído adicionado à imagem original e $f'(x, y)$ a imagem afetada pelo ruído.

Segundo Jain e Tyagi (2013) e Mythili e Kavitha (2011), o ruído do tipo aditivo mais utilizado na literatura é o ruído do tipo Gaussiano. Conforme o próprio no e sugere, esse tipo de ruído segue a distribuição Gaussiana, também conhecida como distribuição normal. A função de densidade de probabilidade (*Probability Density Function* - PDF) P de uma variável s é dada pela Equação 2.8:

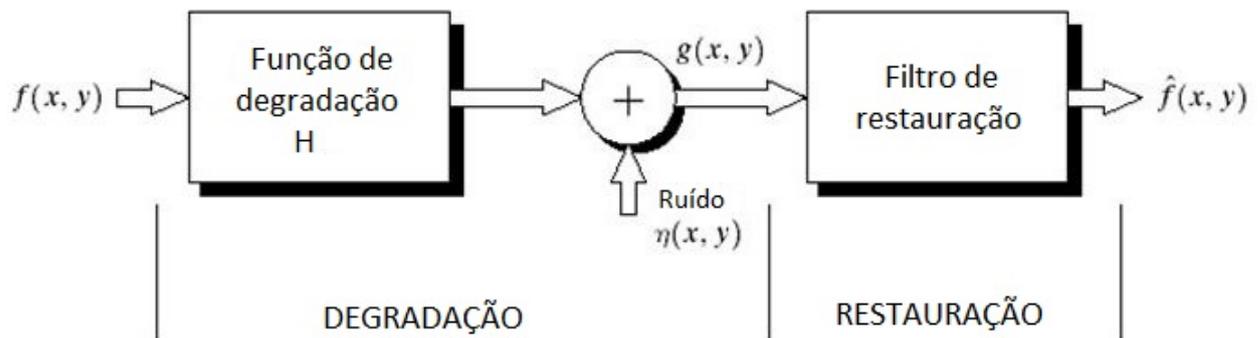
$$P(s) = \frac{1}{\sqrt{\sigma^2 2\pi}} e^{-\frac{(s - \mu)^2}{2\sigma^2}} \quad (2.8)$$

onde s representa a intensidade, μ a média do valor de s e σ o desvio padrão.

Ensaio representando a degradação da imagem por ruídos do tipo aditivos foram realizados com o ruído do tipo Gaussiano.

O processo de filtragem de ruídos, também chamado de restauração de imagens, tem como principal objetivo a restauração de uma imagem degradada por algum tipo de ruído para o seu estado original. A Figura 6 ilustra o processo de restauração, onde um ruído $\eta(x, y)$ é adicionado a uma imagem $f(x, y)$ a partir de uma função de degradação H . O resultado obtido a partir de um filtro $g(x, y)$ é uma imagem restaurada $\hat{f}(x, y)$.

Figura 6 – Processo de restauração de imagem. (Fonte: Adaptado de Gonzalez e Woods (2009))



O processo de restauração de imagens ruidosas pode ser aplicado tanto no domínio espacial quanto no domínio da frequência. O termo filtragem no domínio espacial da imagem indica que o processo de filtragem ocorrerá diretamente sobre os valores dos *pixels* de uma determinada imagem. Já a filtragem no domínio da frequência, ocorre por meio de modificações e operações no domínio da frequência.

Neste trabalho, a restauração de imagens foram realizadas no domínio espacial. Assim, como principais filtros para restauração de imagens afetadas por ruído do tipo Gaussiano, foram aplicados os filtros do tipo Gaussiano (SOLOMON; BRECKON, 2000) e *Non-local Means* (BUADES et al., 2005).

A aplicação do filtro Gaussiano tem o efeito de suavizar uma imagem sendo o grau de suavização controlado pelo desvio padrão (σ). Seu *kernel* segue o modelo matemático de uma função gaussiana bidimensional, conforme a Equação 2.9:

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (2.9)$$

onde x e y representam as dimensões do *kernel* do filtro e σ o valor do desvio padrão da função gaussiana com $\sigma > 0$.

Já a aplicação filtro *Non-local Means* (NLM) é baseado em médias não locais e tem grande aplicação também para a filtragens e imagens com ruídos do tipo aditivo (Gaussiano). Basicamente, o algoritmo do filtro NLM baseia-se no fato de que imagens digitais possuem características que não se repetem de forma local, mas sim, de forma global na imagem (BUADES

et al., 2005). Assim, o filtro NLM busca o valor estimado da intensidade de cada *pixel* (i) em uma determinada região da imagem (f), calculando então a média ponderada dessa região. A similaridade é estimada conforme uma imagem com ruído g , na forma $g(i)|i \in f, NLM_{[g](s)}$, onde NLM representa o valor estimado de um determinado *pixel* i (BEN et al., 2016), conforme a Equação 2.10:

$$NLM_{[g](s)} = \sum_{j \in f} \omega(i, j)g(j) \quad (2.10)$$

onde $\omega(i, j)$ representa a similaridade dos pesos dos *pixels* i e j , de forma a satisfazer as condições $0 \leq \omega(i, j) \leq 1$ e $\sum_{j \in f} \omega(i, j) = 1$. A similaridade entre os *pixels* i e j está relacionada a intensidade dos vetores de níveis de cinza $g(N_i)$ e $g(N_j)$, respectivamente. Em complemento, o cálculo do peso do $\omega(i, j)$ é calculado de acordo com a Equação 2.11:

$$\omega(i, j) = \frac{1}{Z(i)} \exp \frac{-\|N_i - g(N_j)\|_{2, \tau}^2}{\sigma^2} \quad (2.11)$$

onde $Z(i)$ refere-se à função de normalização e pode ser obtido a partir da Equação 2.12:

$$Z(i) = \sum_j \exp \frac{-\|N_i - g(N_j)\|_{2, \tau}^2}{\sigma^2} \quad (2.12)$$

onde $\|N_i - g(N_j)\|_{2, \tau}^2$ representa o peso da distância Euclidiana e σ ao desvio padrão do ruído Gaussiano (BRITO, 2020).

Os pseudocódigos 3 e 4 representam, respectivamente, o processo de filtragem de uma imagem pelo filtro Gaussiano e pelo filtro *Non-local Means*.

Algorithm 3 Pseudocódigo para filtragem de imagens pelo filtro Gaussiano

Entrada: Imagem ruidosa (I_r), σ , $kernel(x, y)$

11 **início**

12 | $I_f = \text{filtroGasussiano}(I_r, \sigma, kernel(x, y))$

13 **fim**

Saída: Imagem filtrada (I_f)

Algorithm 4 Pseudocódigo para filtragem da imagem pelo filtro *Non-local Means*

Entrada: Imagem ruidosa (I_r), tamanho *patch*, altura *patch*, σ , $kernel(x, y)$

14 **início**

15 | $I_f = \text{filtroNLM}(I_r, \text{tamanho } patch, \text{distancia } patch, \sigma, kernel(x, y))$

16 **fim**

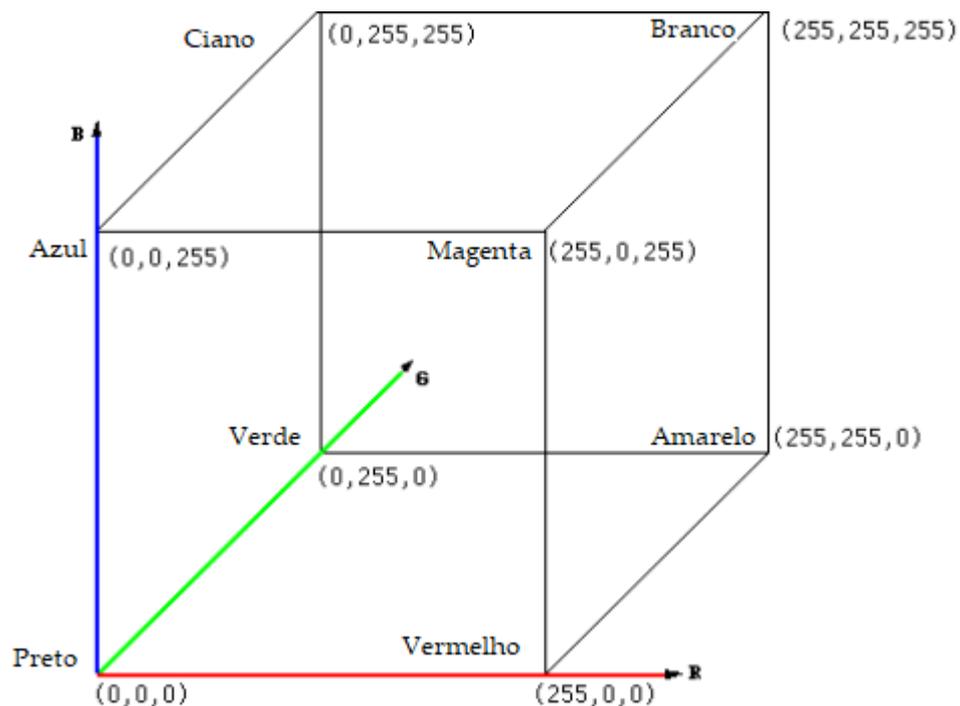
Saída: Imagem filtrada (I_f)

Quanto à operação de conversão de padrões de cores, a mesma é baseada nas cores primárias vermelha, verde e azul, o padrão de cor RGB (do inglês *Red, Green e Blue* - Vermelho, Verde e Azul) é aplicado, principalmente, em dispositivos de captura e exibição de imagens coloridas

(monitores, televisão, máquinas fotográficas, por exemplo), computação gráfica, armazenamento de imagens digitais, entre outras aplicações (IBRAHEEM et al., 2012).

Graficamente o padrão RGB pode ser representado na forma de um cubo 3D (Figura 7), onde cada componente de cor é representado por um eixo do cubo. Cada eixo tem a mesma faixa de valores, que variam de 0 a 255 na representação de 1 *byte*.

Figura 7 – Representação gráfica do padrão RGB. (Fonte: Adaptado de Ibraheem et al. (2012))



Em referência ao processamento de imagens digitais (Figura 3), o padrão RGB aplica-se, principalmente, às etapas de aquisição de imagens e visualização do resultado. Porém, devido a sua baixa capacidade de captar a variação da intensidade nos componentes de cores, seu uso não é recomendado nas demais etapas do processo (SARAVANAN et al., 2016).

Outros padrões de cores que têm importante aplicação no processamento de imagens digitais são os modelos baseados em tonalidade (ou matiz), saturação e intensidade (HSV - do inglês *Hue Saturation Intensity*) e suas variações, como luminosidade (HSL - *Lightness*) e luminância (HSV - *Value*). Uma característica importante desses padrões é a representação das cores, que são representadas em uma forma mais natural em relação à percepção da visão humana.

Assim como no padrão RGB, as imagens digitais nos padrões de cores HSI, HSL e HSV também são representadas na forma de uma matriz 3D. Já a representação gráfica é na forma de um semi cone para HSV, conforme ilustrado pela Figura 8 e um cone duplo para os modelos HSI e HSL (Figura 9).

Figura 8 – Representação gráfica do padrão de cor HSV. (Fonte: Adaptado de Ribeiro e Gomes (2010))

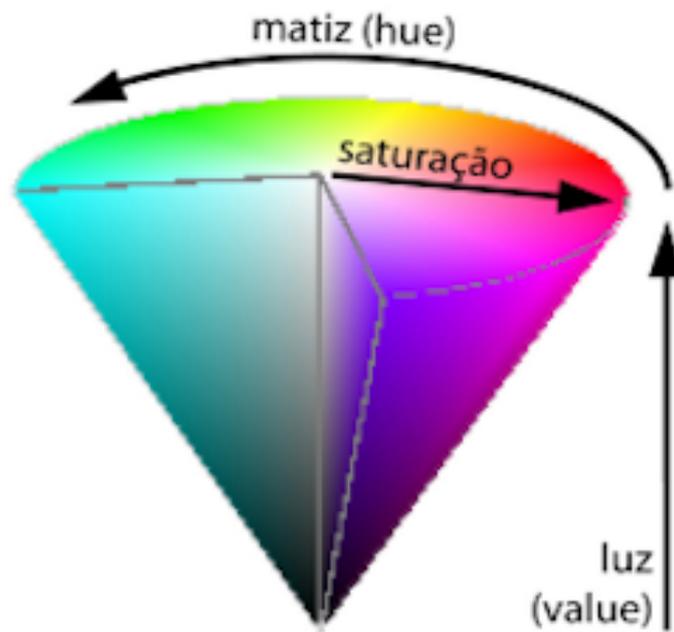
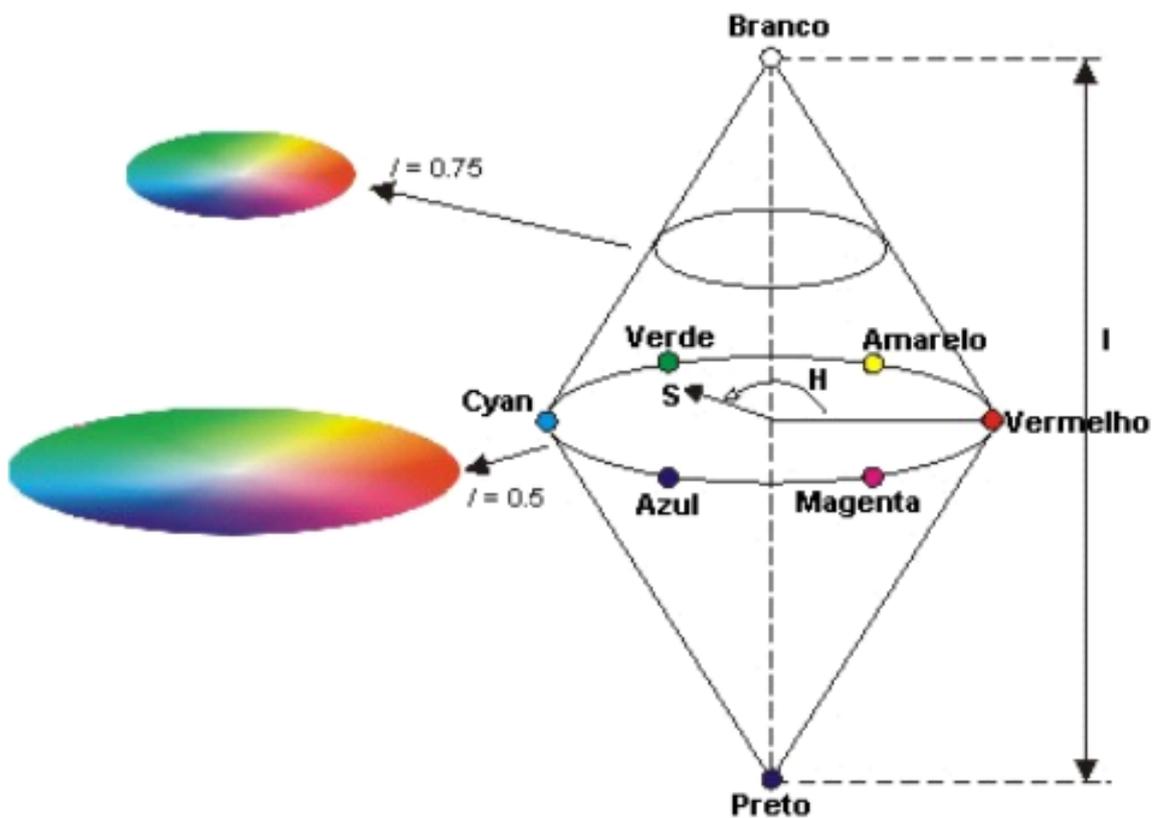


Figura 9 – Representação gráfica dos padrão de cor HSI. (Fonte: Adaptado de Matos et al. (2007))



Outra importante característica é que ao se analisar individualmente cada componente,

percebe-se que o objeto-imagem está contido de forma mais consistente no canal matiz (ou tonalidade). Essa característica torna tais espaços de cores muito utilizados na etapa de segmentação de imagens, por meio da técnica conhecida como decomposição de cores. Conforme já abordado, a captura de imagens em aplicações de processamento de imagens digitais são realizadas no padrão RGB, porém, para operações de segmentação neste padrão de cor não é recomendado. Portanto, é possível realizar a operação de conversão do padrão RGB para os padrões HSI, HSL e HSV, e vice-versa.

A Equação 2.13 demonstra a conversão do padrão de cor RGB para HSI.

$$H = \begin{cases} \theta & \text{se } B \leq G \\ 306 - \theta & \text{se } B > G \end{cases}$$

$$\text{onde } \theta = \cos^{-1} \left(\frac{0,5[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)+(G-B)^2}} \right) \quad (2.13)$$

$$S = 1 - \frac{3}{(R+G+B)}[\min(R, G, B)]$$

$$I = \frac{1}{3}(R + G + B)$$

A conversão inversa, ou seja, de HSI para RGB é realizada de acordo com o valor do componente H, conforme a Equação 2.14:

para $0^\circ \leq H < 120^\circ$, $H = H - 120^\circ$

$$R = I \left(1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right)$$

$$G = 3I - (R + G)$$

$$B = I(1 - S)$$

para $120^\circ \leq H < 240^\circ$, $H = H - 120^\circ$

$$R = I(1 - S)$$

$$G = \left(1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right) \quad (2.14)$$

$$B = 3I - (R + G)$$

para $240^\circ \leq H < 360^\circ$, $H = H - 240^\circ$

$$R = 3I - (R + G)$$

$$G = I(1 - S)$$

$$B = \left(1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right)$$

Para realizar a conversão do padrão RGB para HSL, primeiramente é necessário normali-

zar cada componente R, G e B, de acordo com a Equação 2.15:

$$\begin{cases} R' = \frac{R}{255} \\ G' = \frac{G}{255} \\ B' = \frac{B}{255} \end{cases} \quad (2.15)$$

Após a normalização de cada componente é necessário então encontrar seu valores máximos e mínimos para então realizar a conversão, conforme as Equações 2.16 e 2.17, respectivamente.

$$\begin{aligned} C_{max} = \max(R', G', B') \quad C_{min} = \min(R', G', B') \\ \Delta = C_{max} - C_{min} \end{aligned} \quad (2.16)$$

$$H = \begin{cases} 60^\circ - \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & \text{se } C_{max} = R' \\ 60^\circ - \left(\frac{B' - R'}{\Delta} + 2 \right) & \text{se } C_{max} = G' \\ 60^\circ - \left(\frac{R' - G'}{\Delta} + 4 \right) & \text{se } C_{max} = B' \end{cases} \quad (2.17)$$

$$S = \begin{cases} 0 & \text{se } \Delta = 0 \\ \frac{\Delta}{1 - |2L - 1|} & \text{se } \Delta \neq 0 \end{cases}$$

$$L = (C_{max} + C_{min})/2$$

A conversão do padrão HSL para RGB deve seguir a condições $0 \leq H < 360^\circ$. Primeiramente deve-se encontrar o valor do componente croma C , conforme a Equação 2.18:

$$C = (1 - |2L - 1|) S \quad (2.18)$$

Com o valor do componente croma definido, deve-se encontrar para cada componente R_1 , G_1 e B_1 , em suas respectivas faces do cubo RGB, sobre os valores similares de tonalidade (matiz) e croma como cores, conforme a Equação 2.19:

$$\begin{aligned} X = C(1 - \left| \left(\frac{H}{60^\circ} \right) \bmod 2 - 1 \right|) \\ (R_1, G_1, B_1) = \begin{cases} (C, X, 0) & \text{se } 0^\circ \leq H < 60^\circ \\ (X, C, 0) & \text{se } 60^\circ \leq H < 120^\circ \\ (0, C, X) & \text{se } 120^\circ \leq H < 180^\circ \\ (0, X, C) & \text{se } 180^\circ \leq H < 240^\circ \\ (X, 0, C) & \text{se } 240^\circ \leq H < 300^\circ \\ (C, 0, X) & \text{se } 300^\circ \leq H < 360^\circ \end{cases} \end{aligned} \quad (2.19)$$

Por fim, adicionando uma variável m equivalente a luminosidade a cada componente R, G e B, é possível obter seus respectivos valores por meio da Equação 2.18:

$$\begin{aligned} m = L - \frac{C}{2} \\ (R, G, B) = ((R_1 + m), (G_1 + m), (B_1 + m)) \end{aligned} \quad (2.20)$$

Assim como na conversão do padrão RGB para HSL, na conversão do RGB para o HSV também é necessário realizar a normalização dos valores dos componentes R, G e B, e encontrar seus respectivos valores máximos e mínimos para depois então realizar a conversão, conforme as Equações 2.21 e 2.22, respectivamente.

$$\begin{cases} R' = \frac{R}{255}, \\ G' = \frac{G}{255}, \\ B' = \frac{B}{255}, \end{cases} \quad (2.21)$$

$$C_{max} = \max(R', G', B') \quad C_{min} = \min(R', G', B') \\ \Delta = C_{max} - C_{min}$$

$$H = \begin{cases} 60^\circ - \left(\frac{G' - B'}{\Delta} \bmod 6 \right) & \text{se } C_{max} = R' \\ 60^\circ - \left(\frac{B' - R'}{\Delta} + 2 \right) & \text{se } C_{max} = G' \\ 60^\circ - \left(\frac{R' - G'}{\Delta} + 4 \right) & \text{se } C_{max} = B' \end{cases} \quad (2.22)$$

$$S = \begin{cases} 0 & \text{se } \Delta = 0 \\ \frac{\Delta}{C_{max}} & \text{se } \Delta \neq 0 \end{cases}$$

$$V = C_{max}$$

A conversão do padrão HSV para RGB, segue praticamente os mesmos princípios que a conversão do padrão HSI para RGB. Portanto, é possível realizar essa conversão conforme as Equações 2.23, 2.24 e 2.25, respectivamente.

$$C = V \times S \quad (2.23)$$

$$X = C(1 - |(\frac{H}{60^\circ}) \bmod 2 - 1|)$$

$$(R_1, G_1, B_1) = \begin{cases} (C, X, 0) & \text{se } 0^\circ \leq H < 60^\circ \\ (X, C, 0) & \text{se } 60^\circ \leq H < 120^\circ \\ (0, C, X) & \text{se } 120^\circ \leq H < 180^\circ \\ (0, X, C) & \text{se } 180^\circ \leq H < 240^\circ \\ (X, 0, C) & \text{se } 240^\circ \leq H < 300^\circ \\ (C, 0, X) & \text{se } 300^\circ \leq H < 360^\circ \end{cases} \quad (2.24)$$

$$m = V - C \quad (2.25) \\ (R, G, B) = ((R_1 + m), (G_1 + m), (B_1 + m))$$

Fundada em 1913, a *Commission Internationale de l'Eclairage* (CIE, do francês Comissão Internacional de Iluminação) define a sensação de cor baseada nos elementos de luminosidade,

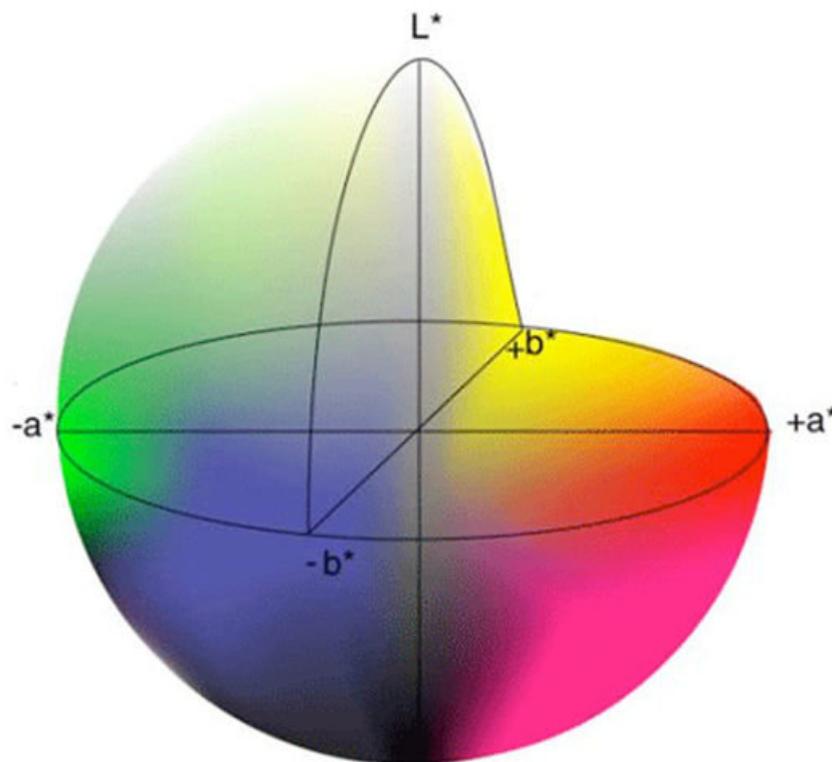
matiz e cromaticidade, definindo a condição de existência de cor a partir de três elementos, iluminante, objeto e observador (SANGWINE; HORNE, 1998).

Baseados nestes três elementos, a CIE criou, em 1964, o padrão de cor conhecido como CIE Lab, onde o componente L representa a luminosidade podendo variar de 0 até 100, o componentes 'a' representa a cromaticidade variando do verde (valores negativos) até o vermelho (ou magenta, para valores positivos) e o componente 'b' também representando a cromaticidade, porém, podendo variar do amarelo (para valores negativos) até o azul (ou ciano, para valores positivos) (BANSAL; AGGARWAL, 2011).

Em 1976, foi criado o padrão CIE L*a*b* a partir de melhorias do CIE Lab criado em 1964. O novo padrão fornece uma diferenciação de cores mais precisa em relação à percepção humana (DURMUS, 2020). Porém, como o antigo padrão ainda é muito utilizado foi adotado a utilização do asterisco (*) na nomenclatura do novo padrão, assim como, é muito comum encontrar na literatura a notação 'CIE Lab 1964', se referendo ao padrão criado em 1964.

A representação gráfica do padrão CIE L*a*b* pode ser representada por uma esfera, conforme ilustrado pela Figura 10.

Figura 10 – Representação gráfica do padrão CIE L*a*b*. (Fonte: Adaptado de Moghadam et al. (2013))



Uma vez que o padrão CIE L*a*b*, assim como seu antecessor, é baseado no padrão de

cor CIE XYZ, a conversão a partir de imagens RGB se dá em duas etapas (KAUR; KRANTHI, 2012). Primeiro a imagem RGB é convertida para o padrão CIE XYZ, conforme a Equação 2.26:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,4125 & 0,3576 & 0,1804 \\ 0,2127 & 0,7152 & 0,0722 \\ 0,0193 & 0,1192 & 0,9502 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.26)$$

Uma vez que a imagem foi convertida para o padrão CIE XYZ, ela agora pode ser convertida para o padrão CIE L*a*b*, conforme a Equação 2.27:

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 & \text{se } \frac{Y}{Y_n} > 0,008856 \\ 903,3 \left(\frac{Y}{Y_n} \right) & \text{se } \frac{Y}{Y_n} \leq 0,008856 \end{cases} \quad (2.27)$$

$$a^* = 500 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} \right]$$

$$b^* = 200 \left[\left(\frac{X}{X_n} \right)^{\frac{1}{3}} - \left(\frac{Z}{Z_n} \right)^{\frac{1}{3}} \right]$$

Os pseudocódigos 5 e 6 representam, respectivamente, os processos de conversão do padrão de cor RGB para HSV e CIE L*a*b*.

Algorithm 5 Pseudocódigo para conversão do padrão de cor RGB para o padrão HSV

Entrada: Imagem rgb (I_{rgb})

17 **início**

18 | $I_{hsv} = \text{converterRGBParaHSV}(I_{rgb})$

19 **fim**

Saída: Imagem hsv (I_{hsv})

Algorithm 6 Pseudocódigo para conversão do padrão de cor RGB para o padrão CIE L*a*b*

Entrada: Imagem rgb (I_{rgb})

20 **início**

21 | $I_{lab} = \text{converterRGBParaLAB}(I_{rgb})$

22 **fim**

Saída: Imagem CIE L*a*b* (I_{lab})

2.3 Segmentação da imagem

A etapa de segmentação tem por objetivo dividir, ou isolar, regiões de uma imagem, as quais sejam possíveis a rotulação de objetos de primeiro e segundo plano. Objetos de primeiro plano são assim denominados os objetos de interesse (ROI, do inglês *Region Of Interest*) da

imagem, ou seja, os objetos que se buscam encontrar na imagem. Já os objetos de segundo plano são quaisquer objetos que não são de interesse, ou simplesmente o fundo da imagem (BAXES, 1994).

Algoritmos de segmentação baseiam-se geralmente em duas propriedades básicas, descontinuidade e similaridade. Detecção de bordas por meio de identificação de fronteiras entre regiões é um exemplo de descontinuidade, já a similaridade aloca *pixel* em uma determinada região de acordo com suas similaridades (GONZALEZ; WOODS, 2009).

Exemplos de algoritmos de segmentação que aplicam a propriedade de descontinuidade, ou detecção de bordas são: limiarização (*thresholding*) (KULKARNI, 2012), Otsu (OTSU, 1979) e Canny (CANNY, 1986). Já exemplos de algoritmos que aplicam a propriedade de similaridade são: Algoritmos de crescimento de região (KAMDI; KRISHNA, 2012), divisores de água, também conhecido como *watershed* (VICENT; SOILLE, 1991)..

Neste trabalho buscou-se a separação da Lagarta do Cartucho (*Spodoprepera frugiperda*) do fundo da imagem, esse podendo ser representado ou por folha ou pela espiga do milho.

Considerada a forma mais simples de segmentação de imagens, a limiarização usa somente a intensidade absoluta de cada *pixels*. Esse método pode ter como entrada tanto uma imagem binária quanto uma imagem em escala de tons de cinza, porém, seu resultado é uma imagem binária onde a representação dos *pixels* com valor 0 (preto) normalmente representam o fundo da imagem e os *pixels* com valor 1 (branco) o(s) objeto(s) de interesse (KULKARNI, 2012).

A imagem binária $b(x, y)$ pode ser criada por meio da aplicação um limiar T aplicado no histograma da imagem $f(x, y)$, conforme a Equação 2.28:

$$b(x, y) = \begin{cases} 0, & \text{se } f(x, y) < T \\ 1, & \text{se } f(x, y) \geq T \end{cases} \quad (2.28)$$

O valor do limiar pode ser escolhido por meio de uma análise manual a partir do histograma de uma imagem, por meio de algum algoritmo de seleção automática de limiar. Neste caso, foram utilizados *pixels* sementes.

O pseudocódigo 7 representa o método de segmentação por limiarização e *pixels* sementes.

Adicionalmente, considerado um método de análise de discriminante não paramétrico e não supervisionado, o método de Otsu realiza a seleção automática de um limiar ótimo a partir dos valores de intensidade dos *pixels* de uma imagem digital, permitindo assim, a máxima separação de classes (OTSU, 1979). Entende-se por classes, a região de interesse e o fundo da imagem.

Algorithm 7 Pseudocódigo para segmentação de imagens da Lagarta do Cartucho por meio da técnica de limiarização e *pixels* sementes

Entrada: Imagem rgb (I_{rgb}), *pixels* sementes, limiar L

```

23 início
24    $I_{hsv}$  = converter  $I_{rgb}$  de RGB para HSV
25    $I_h$  = componente H de  $I_{hsv}$ 
26   plotar pixels sementes em  $I_h$ 
27   obter valores dos pixels sementes de  $I_h$ 
28    $L$  = calcular média dos valores dos pixels sementes
29    $\sigma$  = desvio padrão dos valores dos pixels sementes
30    $L_{min}$  =  $L - \sigma$ 
31    $L_{max}$  =  $L + \sigma$ 
32   se fundo == folha então
33     |  $I_s$  = segmentar  $I_h$  acima de  $L_{min}$  e abaixo de  $L_{max}$ 
34   fim
35   se fundo == espiga então
36     |  $I_s$  = segmentar  $I_h$  acima de  $L_{min}$ 
37   fim
38 fim
Saída: Imagem segmentada ( $I_s$ )

```

Em uma imagem digital 2D de dimensões $N \times M$, com níveis de intensidade L , onde n_i denote o número de *pixels* de intensidade i e MN o número total de *pixels*, o histograma é normalizado conforme a Equação 2.29 (GONZALEZ; WOODS, 2010):

$$p_i = \frac{n_i}{N} \quad (2.29)$$

onde $p_i \geq 0$ e $\sum_{i=1}^L p_i = 1$.

A operação de um limiar $T(k) = k$ onde $0 < k < L - 1$, que separa os L níveis de intensidade de uma imagem em duas classes, C_1 e C_2 , representando o objeto de interesse e o fundo da imagem, respectivamente, onde C_1 é constituída por todos os *pixels* no intervalo $[0, k]$ e C_2 no intervalo $[k + 1, L - 1]$, a probabilidade $P_1(k)$ de um *pixel* ser atribuído à classe C_1 é dada pela Equação 2.30:

$$P_1(k) = \sum_{i=0}^k p_i \quad (2.30)$$

Da mesma maneira, a probabilidade de ocorrência da classe C_2 é dada pela Equação 2.31:

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k) \quad (2.31)$$

Os valores das intensidades médias das classes C_1 e C_2 são dados pelas Equações 2.32 e 2.33, respectivamente:

$$m_1(k) = \sum_{i=0}^k iP(i|C_1) = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i \quad (2.32)$$

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i|C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i \quad (2.33)$$

A intensidade média acumulada é dada pela Equação 2.34. Já a média de toda a imagem (média global) é dada pela Equação 2.35:

$$m(k) = \sum_{i=0}^k ip_i \quad (2.34)$$

são momentos cumulativos zero e de primeira ordem do histograma de níveis de tons de cinza. A média total dos níveis de tons de cinza μ_t é expressa pela Equação 2.35:

$$m_G = \sum_{i=0}^{L-1} ip_i \quad (2.35)$$

O limiar ótimo pode ser obtido por meio da minimização de uma das funções discriminantes descritas na Equação 2.36:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \quad \eta = \frac{\sigma_B^2}{\sigma_G^2}, \quad \kappa = \frac{\sigma_T^2}{\sigma_W^2} \quad (2.36)$$

onde σ_W^2 é uma variância dentro da classe, σ_B^2 a variância entre classes e σ_G^2 a variância global. A variância entre classes é definida pela Equação 2.37, enquanto a variância global é definida pela Equação 2.38:

$$\sigma_B^2 = P_1P_2(m_1 - m_2)^2 = \frac{(m_GP_1 - m)^2}{P_1(1 - P_1)} \quad (2.37)$$

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 pi \quad (2.38)$$

A partir da Equação 2.37 é possível verificar que quanto maior a diferença dos valores das médias m_1 e m_2 , maior será a variância entre classes (σ_B^2), confirmando assim, ser uma medida de separabilidade (GONZALEZ; WOODS, 2010). Da mesma forma, sendo σ_G^2 uma constante, é possível verificar que η também é uma medida de separabilidade e maximizar essa métrica equivale a maximização de σ_B^2 . Assim, o objetivo é determinar o valor de k de forma a maximizar a variância entre classes. Portanto, o limiar k que maximiza a função η , é selecionado conforme as Equações 2.39 e 2.40:

$$\eta(k) = \frac{\sigma_B^2(k)}{\sigma_T^2} \quad (2.39)$$

$$\sigma_B^2(k) = \frac{[m_GP_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]} \quad (2.40)$$

Assim, o limiar ótimo de k^* é definido pela Equação 2.41:

$$\sigma_B^2(k^*) = \max_{1 \leq k \leq L-1} \{\sigma_B^2(k)\} \quad (2.41)$$

Por fim, ao obter o limiar ótimo k^* , a imagem original $f(x, y)$ é segmentada a partir da Equação 2.42:

$$b(x, y) = \begin{cases} 0, & \text{se } f(x, y) \leq k^* \\ 1, & \text{se } f(x, y) > k^* \end{cases} \quad (2.42)$$

A utilização do método de Otsu visou automatizar o processo de segmentação das imagens contendo a Lagarta do Cartucho em folhas e em espigas de milho.

O pseudocódigo 8 representa o procedimento para segmentação da Lagarta do Cartucho em imagens contendo folhas como fundo.

Algorithm 8 Pseudocódigo para segmentação de imagens da Lagarta do Cartucho em folhas pelo método de Otsu

Entrada: Imagem rgb (I_{rgb})

39 **início**

40 $I_{lab} = \text{converterRGBParaLAB}(I_{rgb})$

41 $I_a = \text{componente A de } I_{lab}$

42 $I_b = \text{componente B de } I_{lab}$

43 $L_a = \text{metodoOtsu}(I_a)$

44 $L_b = \text{metodoOtsu}(I_b)$

45 $I_{as} = \text{segmentarAcima}(I_a, L_a)$

46 $I_{bs} = \text{segmentarAbaixo}(I_b, L_b)$

47 $I_{abs} = I_{as} \cap I_{bs}$

48 $I_s = \text{segmentar imagem } I_{rgb} \text{ pela imagem } I_{abs}$

49 **fim**

Saída: Imagem segmentada (I_s)

O pseudocódigo 9 representa o procedimento para segmentação da Lagarta do Cartucho em imagens contendo espigas de milho como fundo.

Algorithm 9 Pseudocódigo para segmentação de imagens da Lagarta do Cartucho em espigas de milho pelo método de Otsu

Entrada: Imagem rgb (I_{rgb})

50 **início**

51 $I_{lab} = \text{converterRGBParaLAB}(I_{rgb})$

52 $I_b = \text{componente B de } I_{lab}$

53 $\sigma = \text{desviopadrode } I_b$

54 $L_b = \text{metodoOtsu}(I_b)$

55 $I_{bs} = \text{segmentarAbaixo}(I_b, L_b - (\sigma * 0.25))$

56 $I_s = \text{segmentar imagem } I_{rgb} \text{ pela imagem } I_{bs}$

57 **fim**

Saída: Imagem segmentada (I_s)

2.4 Extração de características

Quanto a extração de características, a mesma tem por objetivo principal o processamento de uma imagem digital de modo que suas propriedades possam ser extraídas para subsequente reconhecimento e classificação (SOLOMON; BRECKON, 2000). Características podem ser interpretadas com internas, quando os *pixels* representam uma determinada região, como por exemplo, ou seja, cor e ou textura. Já as características externas dizem respeito às fronteiras de uma região, ou seja, a atenção primária está voltada para características de forma e tamanho (GONZALEZ; WOODS, 2010).

Independentemente do tipo de características que se queira extrair de uma imagem, é importante que o algoritmo escolhido seja invariante às transformações geométricas, como tamanho, forma, rotação, entre outras. Por fim, como resultado das técnicas de análise de características, é obtido um conjunto de dados na forma vetorial, chamado de vetor de características.

2.4.1 Histograma de Gradientes Orientados

Proposto por Dalal e Triggs (2005) com a finalidade de detectar pedestres em imagens digitais, o descritor *Histogram of Oriented Gradient* (HOG) calcula a distribuição da intensidade do gradientes e também a direção das bordas das células de uma imagem.

O descritor HOG é aplicado, basicamente, em cinco etapas. Normalização de cores, cálculo da intensidade dos gradientes, agrupamento dos *pixels* da imagem em células, agrupamento dessas células em blocos e por fim, a extração de características da magnitude do gradiente (DALAL; TRIGGS, 2005; CHU et al., 2019).

A normalização de cores converte uma imagem para escala de tons de cinza. Posteriormente, é realizado o cálculo da intensidade do gradiente em cada *pixel* da imagem.

A imagem é então uniformemente dividida em células, sendo então, realizado o cálculo da magnitude do gradiente de acordo com seu ângulo de orientação. É então realizada a normalização para que o resultado seja invariante à iluminação.

Por fim, a partir dos histogramas gerados é obtido o descritor com sua estrutura de dados representado na forma de um vetor, onde cada posição contém as características da imagem.

2.4.2 Momentos invariantes de Hu

Utilizado principalmente como descritor de características relacionadas à geometria de uma imagem digital, como por exemplo forma e tamanho, os momentos invariantes foram introduzidos por Hu (1962) a partir da teoria de momentos invariantes algébricos.

Porém, para que seja possível o cálculo dos sete momentos invariantes de Hu, é necessário, a priori, calcular os momentos bidimensionais, centrais e centrais normalizados (ZHAO; WANG,

2010). Entende-se por momentos bidimensionais funções polinomiais projetadas em uma imagem 2D, $f(x, y)$, e tamanho $M \times N$ e de ordem $(p + q)$, são definidos por meio da Equação 2.43:

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x, y) \quad (2.43)$$

onde $p = 0, 1, 2, \dots$ e $q = 0, 1, 2, \dots$ são inteiros.

Momentos centrais de ordem $(p + q)$ e invariantes a transformações de translação e rotação, são definidos pelas Equações 2.44 e 2.45:

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad (2.44)$$

onde $p = 0, 1, 2, \dots$ e $q = 0, 1, 2, \dots$ são inteiros:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad e \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (2.45)$$

onde \bar{x} e \bar{y} são coordenadas do centro de massa da imagem $f(x, y)$.

Os momentos centrais normalizados permitem que os momentos centrais definidos pela Equação 2.44 sejam invariantes à transformações de escala e é definido pela Equação 2.46:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (2.46)$$

onde γ é definido pela Equação 2.47:

$$\gamma = \frac{p + q}{2} + 1 \quad (2.47)$$

para $p + q = 2, 3, \dots$

Por meio da combinação linear dos momentos centrais, Hu (1962) definiu um conjunto de sete momentos invariantes à translação, rotação, escala e espelhamento, os quais são chamados de momentos invariantes de Hu (HU, 1962; ZHAO; WANG, 2010). Assim, esses sete momentos invariantes são definidos a partir da Equação 2.48 até a Equação 2.54:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (2.48)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2.49)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.50)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.51)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \quad (2.52)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.53)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (2.54)$$

Uma vez estabelecido o processo de segmentação, as técnicas de extração de características da Lagarta do Cartucho foram avaliadas. Assim, os descritores de características referentes a textura da imagem são obtidos por meio da utilização do descritor HOG. Para obter características referentes a tamanho e forma, é utilizado o descritor de momentos invariantes de Hu.

O pseudocódigo 10 representa o procedimento de extração de características referentes a textura por meio do descritor HOG.

Algorithm 10 Pseudocódigo para extração de características de imagens baseado no descritor HOG

Entrada: Imagem em escala de tons de cinza (I_c)

58 **início**

59 | definir tamanho da janela J

60 | definir tamanho do bloco por janela B_j

61 | definir orientações O

62 | $V_{hog}, I_{hog} = \text{HOG}(I_c, J, B_j)$

63 **fim**

Saída: vetor de características V_{hog} , Imagem HOG I_{hog}

O pseudocódigo 11 representa o procedimento de extração dos momentos invariantes de Hu.

Após a extração de características tanto pelo descritor HOG quanto pelo método de Hu, os vetores de características gerados por cada descritor são concatenados de maneira que se tornem um único vetor de características, contendo tanto características de textura como de forma e tamanho, na forma Vetor[HOG,Hu].

Por fim, para que as características sejam corretamente utilizadas para a etapa de classificação, é necessário realizar a normalização do vetor de características, de forma que os valores que representam as características de textura e forma e tamanho estejam na mesma escala. O pseudocódigo 12 representa o processo de normalização do vetor de características.

Algorithm 11 Pseudocódigo para extração de características de imagens baseado no descritor de momentos invariantes de Hu

Entrada: Imagem em escala de tons de cinza (I_c)

```

64 início
65   obter momentos centrais ( $I_c$ )
66   obter momentos invariantes de Hu ( $I_c$ )
67   para cada momento invariante de Hu faça
68      $V_{hu}[i]$  = normalização do momento invariante de Hu
69   fim
70 fim

```

Saída: vetor de características V_{hu}

Algorithm 12 Pseudocódigo para normalização do vetor de características Vetor[HOG,Hu]

Entrada: Vetor de características ($V_{[hog,hu]}$)

```

71 início
72    $m$  = média( $V_{[hog,hu]}$ )
73    $\sigma$  = desviopadro( $V_{[hog,hu]}$ )
74   para cada  $c$  em  $V_{[hog,hu]}$  faça
75      $v$  =  $V_{[hog,hu]}[c]$ 
76      $V_{[hog,hu]}[c]$  =  $(v - m)/\sigma$ 
77   fim
78 fim

```

Saída: vetor de características $V_{[HOG,hu]}$

Para validação da normalização do vetor de características, basta que a média dos valores seja igual a 0 e o desvio padrão igual a 1. Após a normalização do vetor de características, o mesmo é salvo em disco de modo a criar um banco de características da Lagarta do Cartucho.

2.4.3 Redução de dimensionalidade do vetor de características

É comum em sistemas de reconhecimento de padrões que utilizam imagens a ocorrência de um número elevado de atributos. Por exemplo, em uma imagem de 1024 x 1024 *pixels*, cada *pixel* é considerado como um atributo, a imagem teria mais de um milhão de atributos. Este problema se refere à dimensionalidade (FACELI et al., 2011).

Para que dados com número elevado de atributos possam ser utilizados, a quantidade de atributos deve ser reduzida. Uma das principais técnicas de redução de dimensionalidade utilizada é a Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) (HONGYU et al., 2016). A PCA é uma técnica de análise multivariada inicialmente descrita por Pearson (1901) e utilizada por Hotelling (1933) com o propósito de analisar as estruturas de correlação. Assim, considerando uma matriz X de dados com n amostras que representam o

número de observações e m variáveis independentes, conforme a Equação 2.55:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \quad (2.55)$$

Para se obter os componentes principais, inicialmente padronizam-se os dados da matriz X , de modo que seja gerada uma nova matriz Z e então calculá-se a matriz de correlação (VARELLA, 2008). Assim, os valores dos elementos z_{ij} são calculados de acordo com a Equação 2.56:

$$z_{ij} = \frac{x_i - \bar{x}_j}{S_j}, \quad i=1, 2, \dots, n \text{ e } j=1, 2, \dots, m \quad (2.56)$$

onde \bar{x}_j representa a média estimada da variável j , conforme a Equação 2.57:

$$\bar{x}_j = \frac{\sum_{i=1}^n X_{ij}}{n}, \quad i=1, 2, \dots, n \text{ e } j=1, 2, \dots, m \quad (2.57)$$

e S_j representa o desvio padrão da variável j , conforme a Equação 2.58:

$$S_j = \sqrt{\hat{V}ar(x_j)}, \quad j=1, 2, \dots, m \quad (2.58)$$

Dessa forma, é obtida a matriz de dados Z :

$$Z = \begin{bmatrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{bmatrix} \quad (2.59)$$

A partir da obtenção da matriz Z é possível calcular a matriz de correlação R , conforme a Equação 2.60:

$$R = \frac{1}{p-1}(Z^t Z) \text{ para } p \neq 1 \quad (2.60)$$

Adicionalmente, são calculados os autovalores e autovetores a fim de determinar os componentes principais. Uma vez dados os autovalores $\lambda_1, \lambda_2, \dots, \lambda_m$, são determinados sob a matriz de correlação $m \times m$, conforme a Equação: 2.62:

$$\det = |R - \lambda I| = 0 \quad (2.61)$$

onde I representa a matriz identidade de ordem m .

Posteriormente, são calculados os autovetores $\mu_1, \mu_2, \dots, \mu_m$, conforme a Equação ??:

$$R\mu = \lambda\mu \quad (2.62)$$

onde para cada autovalor λ_i existe um autovetor μ_i correspondente.

Por fim, é realizada a normalização dos autovetores, conforme a Equação 2.63:

$$\vartheta_j = \frac{1}{\mu_j^t \mu_j} \mu_j \quad (2.63)$$

Ao realizar a normalização dos autovetores, os m autovetores normalizados representam os componentes principais.

O pseudocódigo 13 representa o procedimento de PCA utilizado nesse trabalho.

Algorithm 13 Pseudocódigo para redução de dimensionalidade do vetor de características
Vetor[HOG,Hu]

Entrada: Vetor de características $V_{[hog,hu]}$, número do componentes principais $n = 128$

79 **início**

80 | $V'_{[HOG,hu]} = \text{PCA}(V_{[HOG,hu]}, n)$

81 **fim**

Saída: vetor de características $V'_{[HOG,hu]}$ com n posições

2.5 Reconhecimento de padrões e aprendizado de máquina

Memorizar, observar e explorar situações para aprender fatos, organizar conhecimento novo, entre outros, podem ser considerados atividades relacionadas ao aprendizado (FACELI et al., 2011). Assim, a capacidade que um computador tem de melhorar o desempenho na realização de alguma tarefa baseada na experiência, é chamada de aprendizado de máquina (AM), do inglês *machine learning* (ML).

As técnicas de AM podem ser consideradas em uma subárea da inteligência artificial (IA), que tem como objetivo, o desenvolvimento de métodos computacionais voltados ao auxílio da tomada de decisão, baseados em conhecimento existentes. Porém, outras áreas de pesquisas como probabilidade e estatística, teoria da computação, neuro ciência, entre outras, têm contribuições significativas para o avanço da IA e AM (MITCHELL, 1997). Técnicas de classificação são relacionadas ao AM em duas categorias, aprendizado supervisionado e aprendizado não supervisionado.

Aprendizado supervisionado baseia-se na existência de padrões já existentes e classificados, servindo como exemplos de treinamento para que um classificador possa ser generalizado de forma eficiente para novos conjuntos de dados (DUDA et al., 2012). Aprendizado não supervisionado baseia-se em exemplos não rotulados, buscado o agrupamento (*clustering*) a partir do corpo total do conjunto de dados (FERNÁNDEZ, 2011).

O reconhecimento de padrões é considerado uma descrição estrutural de um objeto ou região de interesse em uma imagem. Classe de padrões é um grupo de padrões que compartilha características em comum (AUGUSTO, 2012). Desta forma, o processo de reconhecimento de padrões (RP) envolve a atribuição desses padrões à suas respectivas classes (GONZALEZ;

WOODS, 2009). De modo geral, cada padrão é formado por um vetor de características da seguinte forma:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_{\dots} \\ x_n \end{bmatrix} \quad (2.64)$$

onde cada componente x_i corresponde ao i -ésimo descritor e n o número referente ao descritor.

A partir dos vetores de características, a classificação ocorre de acordo com o posicionamento desses vetores no espaço de características, onde grupos compostos por características similares são identificados.

Conforme exposto anteriormente, RP e classificação são técnicas relacionadas ao AM. Adicionalmente, os classificadores podem ser agrupados em diferentes tipos, como por exemplo, baseado em distância (vizinhos mais próximos – KNN) (BEYER et al., 1999), baseado em procura (árvores de decisão e regressão) (RIVEST, 1987), probabilístico (classificador Bayesiano) (MITCHELL, 1997), baseados em otimização (máquina de suporte de vetores) (CORTES; VAPNIK, 1995), entre outros. Para a realização deste trabalho será abordado o classificador baseado em máquina de vetores de suporte, *Support Vector Machine* (SVM) (HAYKIN, 1994).

2.5.1 Máquina de vetores de suporte

Embasada pela teoria do aprendizado estatístico, a máquina de vetores de suporte (SVM – *Support Vector Machine*) é um classificador supervisionado utilizado na etapa de reconhecimento de padrões e classificação (CORTES; VAPNIK, 1995).

Dentre as principais vantagens do classificador SVM destacam-se sua alta capacidade de generalização, evitando assim *overfitting*, convexidade da função objetiva, ou seja, possui um ótimo global apenas, possui ampla e estabelecida teoria matemática e apresenta soluções para problemas lineares e não lineares (SÁ, 2016). O classificador SVM tem por objetivo a separação de um conjunto de dados de treinamento em duas classes por um hiperplano, chamado de hiperplano ótimo, de modo a maximizar a separação entre as classes.

Vetores de suporte (SV – *support vector*) são assim denominados os vetores de características de cada classe localizados mais próximos ao hiperplano, representando assim, uma margem de separação entre as classes (BURGES, 1998). Essas margens podem ser definidas como margens rígidas ou margens suaves.

2.5.2 SVM linear com margens rígidas

SVMs lineares com margens rígidas definem um hiperplano que maximiza a separação entre duas classes a partir de um conjunto de dados de treinamento x com n objetos $x_i \in X$ e seus

respectivos rótulos $y_i \in Y$, tal que X representa o conjunto de dados de entrada e $Y = (-1, +1)$ as possíveis classes (FACELI et al., 2011). Tal hiperplano é definido pela Equação 2.65:

$$w \cdot x + b = 0 \quad (2.65)$$

onde w é o vetor normal ao hiperplano, $w \cdot x$ é o produto escalar dos vetores w e x , e b é um termo de ajuste. Dessa forma, a Equação 2.65 divide o espaço de entrada X em duas regiões conforme a Equação 2.66:

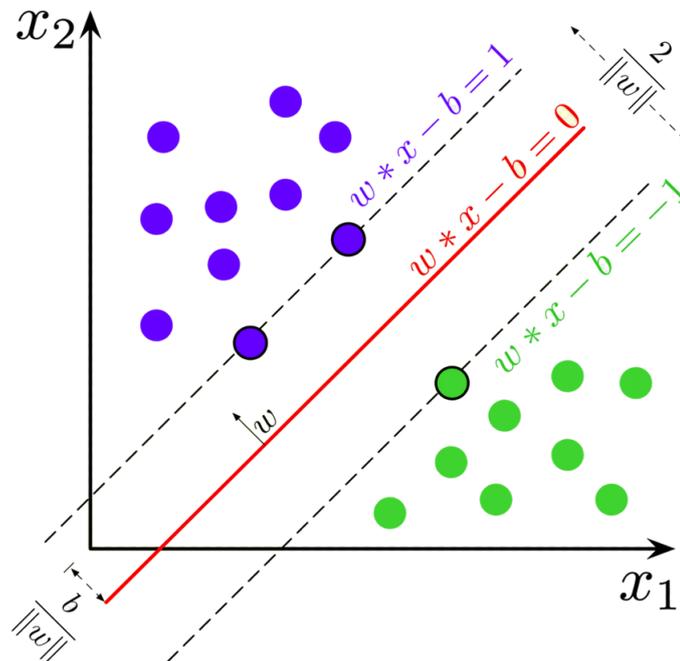
$$\begin{aligned} w \cdot x_i + b &\geq 1 \quad \text{se } y_i = +1 \\ w \cdot x_i + b &\leq -1 \quad \text{se } y_i = -1 \end{aligned} \quad (2.66)$$

que pode ser resumida conforme a Equação 2.67:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \quad \forall (x_i, y_i) \in X \quad (2.67)$$

Assim, assegura-se que nenhum dado de treinamento ultrapasse as margens de separação das classes, caracterizando assim, a SVM linear com margens rígidas (LORENA; CARVALHO, 2007). A Figura 11 ilustra uma SVM linear com margens rígidas.

Figura 11 – Máquina de vetores de suporte com margens rígidas. (Fonte: Noyum et al. (2021))



A maximização da margem de separação dos dados em relação a $w \cdot x + b = 0$ pode ser obtida por meio da minimização de $\|w\|$ (CAMPBELL, 2001), conforme a Equação 2.68:

$$\text{minimizar}_{w,b} = \frac{1}{2} \|w\| \quad (2.68)$$

com as restrições da Equação 2.69:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \quad \forall_i = 1, \dots, n \quad (2.69)$$

O problema de minimização é quadrático, visto que a função objeto é quadrática e as restrições são lineares. Esse problema pode ser solucionado com a introdução da função de Lagrange (CORTES; VAPNIK, 1995), conforme a Equação 2.70:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 \quad (2.70)$$

onde α é denominado como multiplicadores de Lagrange.

A função de Lagrange de ser minimizada em relação a w e b , implicando na maximização das variáveis α_i . Desta forma, obtém-se um ponto de sela conforme a Equação 2.72:

$$\frac{\partial L}{\partial b} = 0 \quad e \quad \frac{\partial L}{\partial w} = 0 \quad (2.71)$$

cuja solução das equações obtém-se:

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (2.72)$$

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad (2.73)$$

onde w é uma combinação linear dos vetores de treinamento.

Assim substituindo as equações 2.72 e 2.73 na equação 2.70, o seguinte problema de otimização é obtido:

$$\text{maximizar}_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2.74)$$

com as restrições:

$$\alpha_i \geq 0, \forall_i = 1, \dots, n \quad e \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.75)$$

Essa formulação é denominada como forma dual, enquanto o problema original é referenciado como forma primal. A forma dual apresenta restrições mais simples e permite representação do problema de otimização em termos de produtos internos entre dados, que será útil para a não linearização das SVMs. Vale ressaltar também, que o problema dual utiliza apenas dados de treinamento e seus rótulos (FACELI et al., 2011).

Neste sentido, suponha que α^* seja a solução do problema dual w^* e b^* as soluções da forma primal. Obtido o valor de α^* , por meio da equação 2.73 pode-se obter o valor de w^* . O valor de b^* é definido por α^* e por condições de Kühn-Tucker (KUHN; TUCKER, 2014), procedentes da teoria de otimização com restrições, que devem ser satisfeitas no ponto ótimo (SMOLA; SCHÖLKOPF, 2004). Desta forma, tem-se:

$$\alpha_i^* (y_i (w^* \cdot x_i + b^*) - 1) = 0, \forall_i = 1, \dots, n \quad (2.76)$$

Pode-se observar na equação 2.75 que α_i^* pode ser diferente de 0 somente para os dados que se localizam sobre os hiperplanos H1 e H2, ou seja, exatamente sobre as margens, sendo os objetos mais próximos ao hiperplano que separa as classes.

Dessa forma, os dados que possuem $\alpha_i^* > 0$ são assim denominados vetores de suporte (BURGES, 1998), conforme citado na introdução do capítulo.

A partir dos SVs e das condições da equação 2.76, o valor de b^* é calculado conforme a equação:

$$b^* = \frac{1}{n_{SV}} \sum_{x_j \in SV} \frac{1}{y_j} - w^* \cdot x_j = \frac{1}{n_{SV}} \sum_{x_j \in SV} \left(\frac{1}{y_j} - \sum_{x_i \in SV} \alpha_i^* y_i x_i \cdot x_j \right) \quad (2.77)$$

onde n_{SV} corresponde ao número de vetores de suporte e SV ao conjunto dos vetores de suporte. Dessa maneira, tem-se como resultado final o classificador $g(x)$, apresentado na equação:

$$g(x) = \text{sgn}(h(x)) = \text{sgn} \left(\sum_{x_i \in SV} y_i \alpha_i^* x_i \cdot x + b^* \right) \quad (2.78)$$

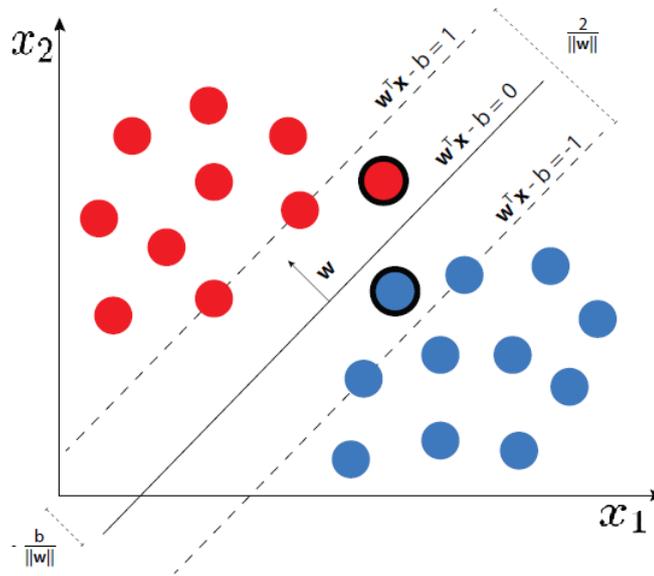
2.5.3 SVM linear com margens suaves

Os classificadores SVM lineares apresentados na seção anterior podem ser estendidos para lidar com conjuntos de dados mais gerais (LORENA; CARVALHO, 2007). Devido a fatores como ruídos, *outliers*, ou até mesmo a própria natureza do problema, que pode ser não linear, permite-se que alguns dados possam violar a restrição da equação 2.69, por meio da introdução da variável de folga (ou relaxamento - *slack variables*) ξ_i , para todo $i = 1, \dots, n$ (CORTES; VAPNIK, 1995), relaxando as restrições da otimização do problema primal (SMOLA; SCHÖLKOPF, 2004), conforme demonstrado na Equação 2.79.

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, \quad \forall_i = 1, \dots, n \quad (2.79)$$

A aplicação deste procedimento (ilustrado na Figura 12) permite que alguns dados permaneçam entre os hiperplanos H1 e H2, assim como a ocorrência de alguns erros de classificação (FACELI et al., 2011). A esse tipo de SVM dá-se o nome de SVM linear com margens suaves.

Figura 12 – Máquina de vetores de suporte com margens suaves. (Fonte: Pecha e Horák (2020))



SVMs com margens suaves tem por objetivo gerar o menor número de erros possíveis e maximizar a margem de separação entre as classes. Desta forma, delimitam-se as variáveis de relaxamento ξ_i com valores mínimos para o primeiro e minimiza-se o erro para o segundo (BURGES, 1998). Assim, reformula-se a função objetivo da Equação 2.68 para a Equação 2.80:

$$\text{minimizar}_{w,b,\xi} = \frac{1}{2} \|w\|^2 + C \left(\sum_{i=1}^n \xi_i \right) \quad (2.80)$$

onde C é uma constante que impõem peso na minimização de erros no conjunto de dados de treinamento.

Da mesma maneira que nas SVMs lineares com margens rígidas, o problema de otimização obtido é quadrático, com restrições lineares (LORENA; CARVALHO, 2007). Porém, da mesma maneira que foi apresentado anteriormente, a solução envolve a introdução da função de Lagrange, tendo como resultado o seguinte problema dual:

$$\text{maximizar}_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2.81)$$

com as restrições:

$$0 \leq \alpha_i \leq C, \forall_i = 1, \dots, n \text{ e } \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.82)$$

onde $\alpha_i^* \neq 0$ e as variáveis ξ_i^* podem ser calculadas como:

$$\xi_i^* = \max \left\{ 0, 1 - y_i \sum_{j=1}^n y_j \alpha_j x_j \cdot x_i + b \right\} \quad (2.83)$$

onde b^* provem de α^* e de condições de Kühn-Tucker:

$$\alpha_j^*(y_i(w^* \cdot x_i + b^*) - 1 + \xi_i^*) = 0(C - \alpha_i^*)\xi_i^* = 0 \quad (2.84)$$

em que os pontos x_i para os quais $\alpha_i^* \geq$ são denominados SVs.

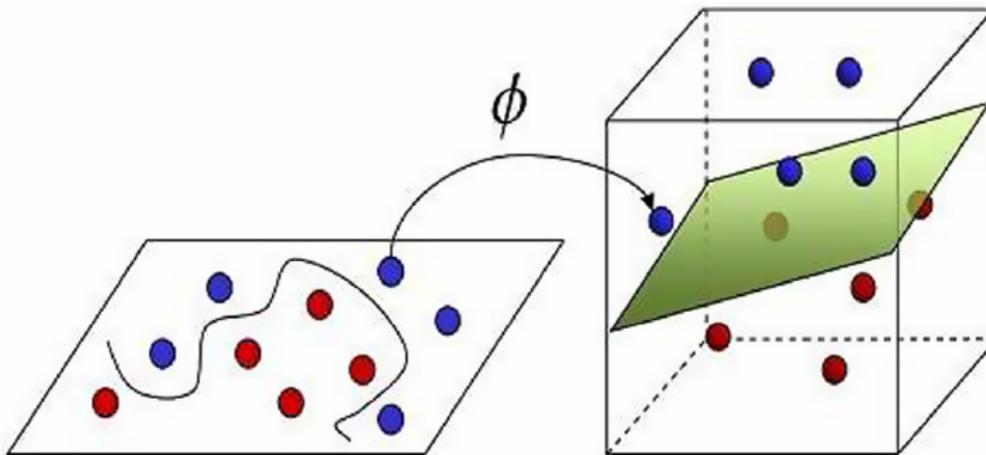
2.5.4 SVM não linear

Conjuntos de dados linearmente separáveis são classificados com eficiência por SVMs lineares e com certa tolerância à ocorrência de erros, no caso de uma SVM linear com margens suaves. Porém, em diversos casos não é possível classificar, de forma eficiente, dados de treinamento por meio de um hiperplano (FACELI et al., 2011).

Dessa forma, é possível que SVMs lidem com problemas não lineares por meio de uma função Φ , mapeando o conjunto de dados de seu espaço original (espaço de entradas), para um espaço de maior dimensão (espaço de características) (HEARST et al., 1998), caracterizando um classificador SVM não linear.

A Figura 13 ilustra um conjunto de dados não linearmente separáveis transformado em um espaço de maior dimensão onde os dados são linearmente separáveis.

Figura 13 – Conjunto de dados não linearmente separáveis (espaço de entrada) convertido para em espaço de maior dimensão (espaço de características). (Fonte: Adaptado de Nalini e Palanivel (2016))



Assim, a partir da escolha de Φ o conjunto de dados de treinamento x , em seu espaço de entrada R^2 , é dimensionado para o espaço de características R^3 , conforme as Equações 2.85 e 2.86:

$$\Phi(x) = \Phi(x_1, x_2) = (x^2, \sqrt{2}x_1x_2, x_2^2) \quad (2.85)$$

$$h(x) = w \cdot \Phi(x) + b = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 + b = 0 \quad (2.86)$$

Dessa maneira, os dados são inicialmente mapeados para um espaço de maior dimensão, sendo então aplicada um SVM linear sobre o novo espaço. Encontra-se então um hiperplano com maior margem de separação, garantindo melhor generalização (LORENA; CARVALHO, 2007).

Para lidar com ruídos e *outliers*, é usada uma SVM linear de margens suaves, aplicando ϕ à equação 2.81, conforme a equação:

$$maximizar_{\alpha} = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\Phi(x_i) \cdot \Phi(x_j)) \quad (2.87)$$

com as restrições da equação 2.82. Assim o classificador obtido se torna:

$$g(x) = sgn(h(x)) = sgn \left(\sum_{x_i \in SV} \alpha_i^* y_i \Phi(x_i) \cdot \Phi(x) + b^* \right) \quad (2.88)$$

em que b^* é calculado como:

$$b^* = \frac{1}{n_{SV:\alpha^* < C}} \sum_{x_j \in SV:\alpha_j^* < C} \left(\frac{1}{y_j} - \sum_{x_i \in SV} \alpha_i^* y_i \Phi(x_i) \cdot \Phi(x_j) \right) \quad (2.89)$$

Como o espaço de características pode ter uma dimensão muito alta, o cálculo de Φ pode ser extremamente custoso, ou até mesmo inviável. Porém, a única informação necessária sobre o mapeamento é o cálculo dos produtos escalares entre os dados no espaço de características, obtidos por meio de funções *kernels* (FACELI et al., 2011).

Uma função *kernel* K calcula, no espaço de características, o produto escalar de dois pontos, x_1 e x_2 , no espaço de entradas (HERBRICH, 2001). Tem-se então a Equação 2.90:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.90)$$

Para o mapeamento de dois dados, x_i e x_j em R^2 , o *kernel* é dado pela Equação 2.91:

$$K(x_i, x_j) = (X_i \cdot X_j)^2 \quad (2.91)$$

A Tabela 6 apresenta os tipos de *kernels* mais utilizados. É possível observar que cada um deles apresentam parâmetros que devem ser preenchidos pelo usuário.

Tabela 6 – Tipos de *Kenels* (Fonte Lorena e Carvalho (2007))

Tipo de <i>Kernel</i>	Função $K(x_i, x_j)$	Parâmetros
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$	δ, k, d
Função de base radial (RBF)	$exp(-\sigma \ x_i - x_j\ ^2)$	σ
Sigmoide	$tahn(\delta(x_i x_j) + k)$	δ e k

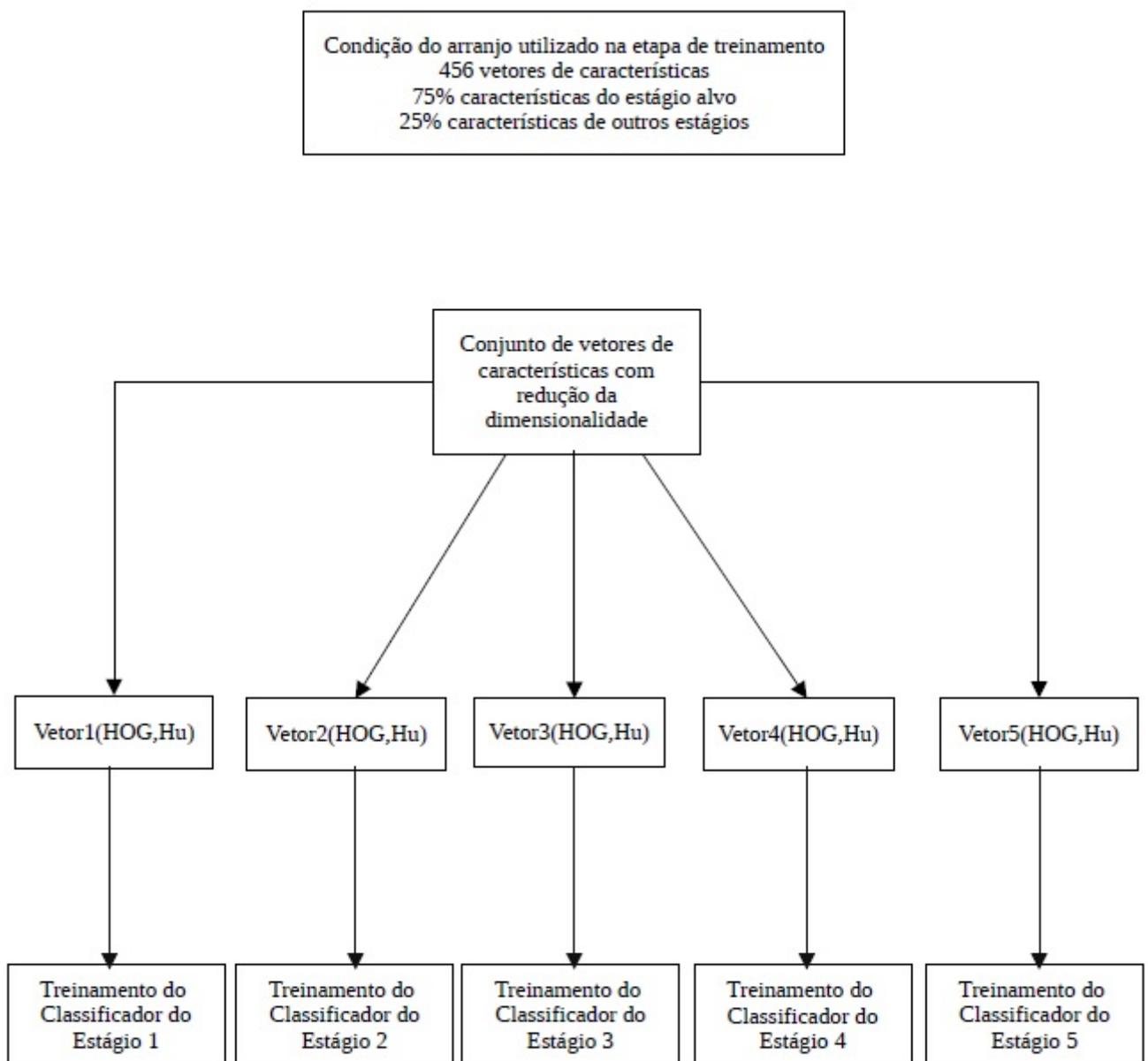
A escolha do tipo de *kernel* e de seus parâmetros afeta o desempenho do classificador. Uma boa escolha pode ser empregar inicialmente o *kernel* função de base radial (RBF - *Radial Basis Function*) (HSU et al., 2003), tendo em vista que o *kernel* linear é um caso especial da função RBF e o *kernel* sigmoide pode comportar-se como o RBF, dependendo da escolha de seus parâmetros (KEERTHI; LIN, 2003; LIN; LIN, 2003).

2.5.5 Conjunto de classificadores SMV

Para a classificação da Lagarta do Cartucho em seus diferentes estágios de desenvolvimento, foi considerado um conjunto de classificadores.

Desta forma, foram considerados classificadores SVM, sendo que cada classificador é responsável por classificar um determinado estágio da Lagarta do Cartucho. A Figura 14 ilustra o processo de treinamento dos classificadores SVM responsáveis pela classificação da Lagarta do Cartucho em seus cinco principais instares de desenvolvimento, ou seja, conhecidos como instar 1, instar 2, instar 3, instar 4 e instar 5.

Figura 14 – Processo de treinamento do conjunto de classificadores do tipo SVM.



Para um conjunto de vetores de características com redução de dimensionalidade, foram definidos arranjos de vetores para cada classificador SVM, de modo a representarem um determinado estágio de desenvolvimento da Lagarta do Cartucho. Dessa maneira, foram trabalhados um total de 456 vetores de características para cada um dos estágios considerados.

O pseudocódigo 14 representa o processo de classificação de padrões da Lagarta do Cartucho.

Algorithm 14 Pseudocódigo para classificação dos padrões de características da Lagarta do Cartucho

Entrada: Vetor de características $V_{[hog,hu]}$, *kernel*, constante de regularização C , coeficiente do *kernel* δ

```

82 início
83   para cada classificador faça
84      $V_{tr[hog,hu]}$  = vetor de treinamento
85      $V_{ts[hog,hu]}$  = vetor de teste
86     se kernel == sigmoid então
87       função kernel =  $\tanh(\delta(x_i \cdot x_j) + \kappa)$ 
88       treinar classificador SVM( $V_{tr[hog,hu]}$ ,  $V_r$ ,  $C$ , função kernel)
89     fim
90     se kernel == linear então
91       treinar classificador SVM( $V_{tr[hog,hu]}$ ,  $V_r$ ,  $C$ )
92     fim
93     testar classificador SVM( $V_{ts[hog,hu]}$ )
94   fim
95 fim
Saída: Acurácia, precisão, revocação, f1_score

```

Para as etapas de treinamento e teste do conjunto de classificadores SVM, foram realizados experimentos com as proporções 80% teste e 20% treinamento, 70% teste e 30% treinamento e 50% teste e 50% treinamento. Para a validação e qualificação do conjunto de classificadores foram utilizadas a Matriz de Confusão e também a curva de Característica de Operação do Receptor (ROC, do inglês *Receiving Operating Characteristics*).

A Tabela 7 ilustra um exemplo de uma matriz de confusão para solução de um problema binário.

Tabela 7 – Matriz de confusão para solução de problemas binários.

		Classe Preditiva	
		+	-
Classe Verdadeira	+	VP	FN
	-	FP	VN

Na Tabela 7, VP representa o número de verdadeiros positivos, ou seja, número de amostras que pertencem a classe positiva e classificadas corretamente. VN representa o número de

verdadeiros negativos, ou seja, o número de amostras negativas e que também foram classificadas corretamente. Já FP, falso positivo, representa o número de amostras correspondentes a classe negativa, mas classificadas como positivas, assim como FN, falso negativo, amostras da classe positiva mas classificadas como pertencentes a classe negativa.

É possível verificar também, que o número de amostras corretamente classificadas encontram-se na diagonal principal da matriz de confusão. Por fim, o número total de amostras é dado por $n = VP + VN + FP + FN$. Dessa forma, a partir da matriz de confusão, é possível verificar o desempenho do classificador a partir do cálculo da acurácia, precisão, revocação e f_1 score, entre outras métricas (VISA et al., 2011).

A acurácia trata do desempenho geral do classificador, conforme a Equação 2.92. Já a precisão trata da proporção de acertos do classificador dentre as amostras previstas como positivas, conforme a Equação 2.93. A revocação refere-se à taxa de acerto da classe positiva, conforme a Equação 2.94. De modo geral, o cálculo da precisão e revocação não são tratados isoladamente, mas sim combinadas em uma única medida, f_1 score, ou seja, a média harmônica da entre a precisão e a revocação. Dando o mesmo grau de importância tanto à revocação quanto à precisão, ou seja, peso igual a 1, chega-se à Equação 2.95.

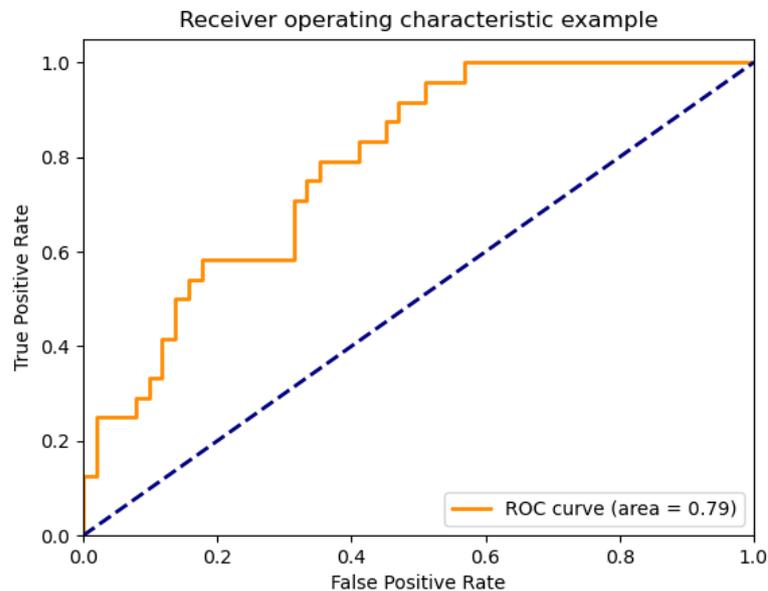
$$\text{acurácia} = \frac{FN}{VP + FN}, \text{ para } VP + FN > 0 \quad (2.92)$$

$$\text{precisão} = \frac{VP}{VP + FP}, \text{ para } VP + FP > 0 \quad (2.93)$$

$$\text{revocação} = \frac{VP}{VP + FN}, \text{ para } VP + FN > 0 \quad (2.94)$$

$$f_1 \text{ score} = 2 \left(\frac{\text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}} \right) \quad (2.95)$$

Em relação à curva ROC, seu uso inicial foi reportado na avaliação e comparação de algoritmos (SPACKMAN, 1989). A curva ROC é um gráfico bidimensional com eixos x e y representando, respectivamente, a taxa de falsos positivos e a taxa de verdadeiros positivos, conforme ilustrado na Figura 15.

Figura 15 – Exemplo de curva ROC. (Fonte: Scikit-learn webpage ¹⁵)

Para a comparação do desempenho de algoritmos de classificação é utilizado o cálculo da área abaixo da curva (AUC, do inglês *Area Under the Curve*). A media AUC produz valores entre 0 e 1, sendo que valores próximos a 1 são considerado melhores. As principais vantagens do uso da curva ROC são a possibilidade de realizar medidas de desempenho independentes de condições, como o limiar da classificação e de custos associados à classificação e à distribuição de classes (FACELI et al., 2011).

¹⁵ Disponível em: <<https://scikit-learn.org>>. Acessado em 20 de jan. de 2022.

Capítulo 3

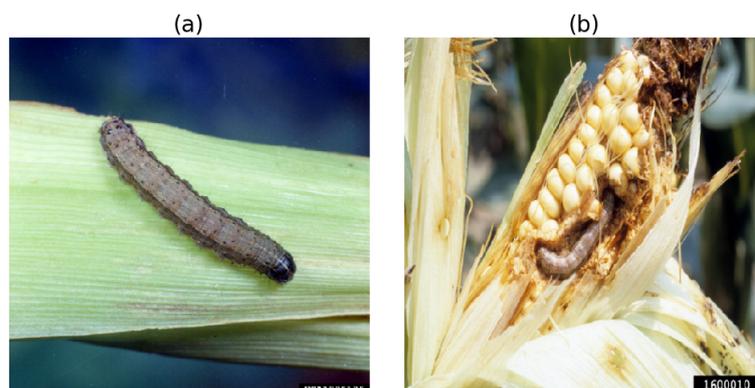
RESULTADOS E DISCUSSÕES

Este Capítulo tem por objetivo apresentar os resultados dos ensaios realizados para o desenvolvimento do método para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho com técnicas do processamento de imagens digitais. Primeiramente, é apresentada a qualificação do banco de imagens. Em seguida, são apresentados os resultados e discussões das etapas de filtragem das imagens e sua respectiva avaliação por meio das métricas MSE e PNSR, assim como das etapas de segmentação das imagens, extração de características e as respectivas classificações.

3.1 Qualificação do banco de imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em cultura do milho

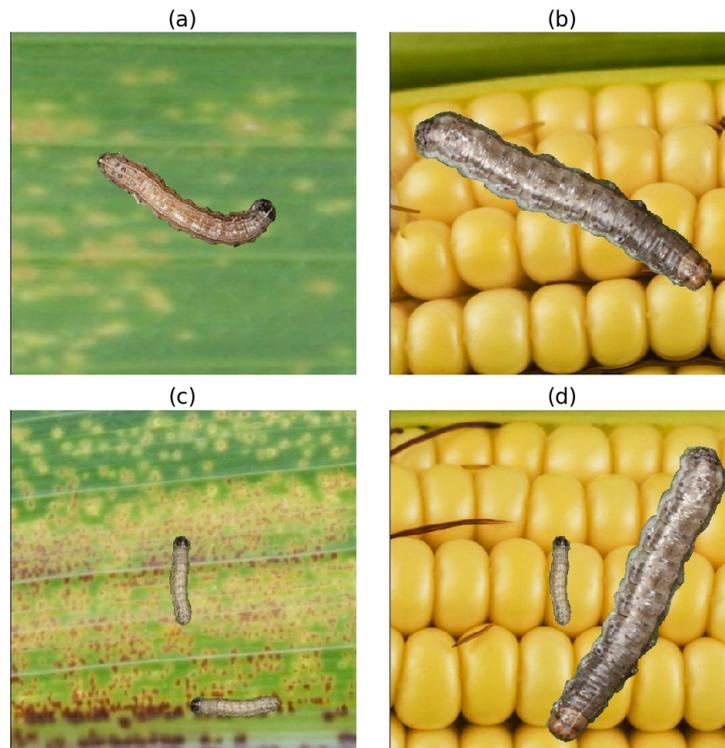
A partir das consultas realizadas, foi possível organizar um banco de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) em cultura do milho com um total de 90 imagens. A Figura 16 ilustra dois exemplos da Lagarta do Cartucho.

Figura 16 – Exemplos da Lagarta do Cartucho (*Spodoptera frugiperda*). (Fonte: (a) Clemson University - USDA Cooperative Extension Slide Series , Bugwood.org, (b) John C. French Sr., Retired, Universities:Auburn, GA, Clemson and U of MO, Bugwood.org)



A Figura 17 ilustra dois exemplos de imagens da Lagarta do cartucho (*Spodoptera frugiperda*) geradas por meio da técnica *data augmentation*.

Figura 17 – Exemplos da Lagarta do cartucho (*Spodoptera frugiperda*) geradas por meio da técnica *data augmentation*. (a) Imagem da lagarta gerada sobre uma folha, (b) imagem da lagarta gerada sobre uma espiga de milho, (c) imagem composta da lagarta gerada sobre uma folha e (d) imagem composta da lagarta gerada sobre uma espiga de milho.



Por meio da técnica utilizada para o aumento da base de imagens, foi possível obter um banco de imagens da Lagarta do Cartucho com um total de 2280 imagens representando seus cinco estágios de desenvolvimento, ou seja, foram geradas 456 imagens para cada estágio de desenvolvimento.

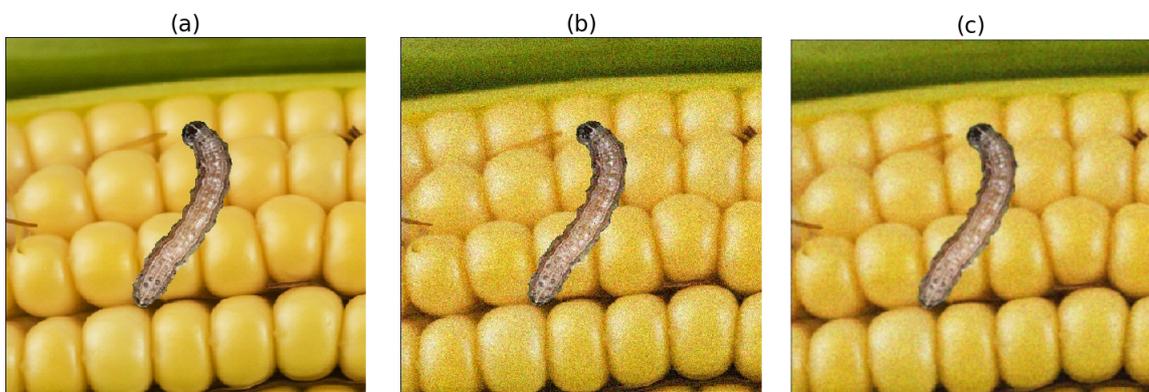
3.2 Resultados obtidos por meio da etapa de filtragem de ruídos

Os ensaios realizados com o filtro do tipo Gaussiano para suavização do ruído do tipo Gaussiano foram realizados de acordo com o pseudocódigo 3. A Tabela 8 apresenta os parâmetros utilizados para este processo de filtragem. Tais parâmetros foram selecionados de acordo com os ensaios realizados e com base na literatura utilizada como referencial bibliográfico.

Tabela 8 – Valores dos parâmetros para o filtro do tipo Gaussiano.

Parâmetros	Valores
Janela (<i>kernel</i>)	7x7
σ	1,0

A Figura 18 ilustra o resultado da suavização do ruído do tipo Gaussiano por meio do filtro Gaussiano em imagens da Lagarta do Cartucho (*Spodoptera frugiperda*).

Figura 18 – Resultado do filtro Gaussiano. (a) Imagem original, (b) imagem afetada por ruído do tipo Gaussiano e (c) imagem filtrada.

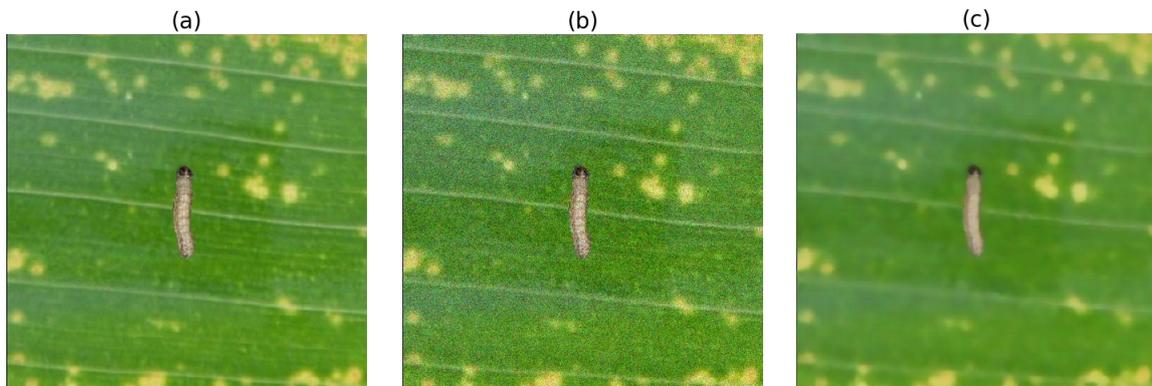
Os ensaios realizados com o filtro do tipo *Non-local Means* para suavização do ruído do tipo Gaussiano foram realizados de acordo com o pseudocódigo 4. A Tabela 9 apresenta os parâmetros utilizados para este processo de filtragem. Assim como a parametrização do filtro Gaussiano, a parametrização do filtro *Non-local Means* foi realizada de acordo com os ensaios e com base na literatura utilizada como referencial bibliográfico.

Tabela 9 – Valores dos parâmetros para o filtro do tipo *Non-local Means*.

Parâmetros	Valores
Janela (<i>kernel</i>)	7x7
Distância do <i>patch</i>	11
Altura do <i>patch</i>	11

A Figura 19 ilustra o resultado da suavização do ruído do tipo Gaussiano por meio do filtro *Non-local Means* em imagens da Lagarta do Cartucho (*Spodoptera frugiperda*).

Figura 19 – Resultado do filtro *Non-local Means*. (a) Imagem original, (b) imagem afetada por ruído do tipo Gaussiano e (c) imagem filtrada.



Para o desenvolvimento do código em linguagem *Python* referente ao processo de filtragem de ruídos foram utilizadas as bibliotecas *OpenCV* em sua versão 4.1 e *skimage* em sua versão 0.16.

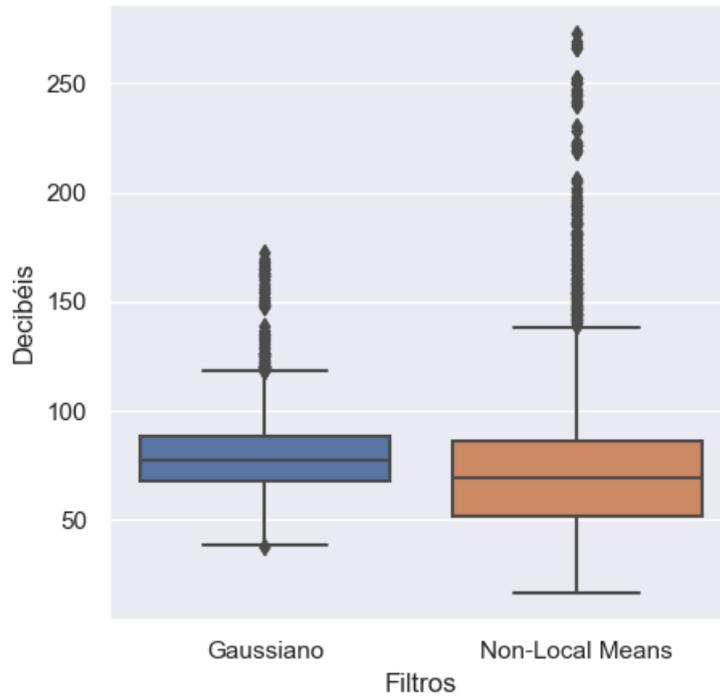
3.2.1 Avaliação da etapa de filtragem de ruídos

Para avaliar o processo de filtragem do ruído do tipo Gaussiano pelos filtros do tipo Gaussiano e do tipo *Non-local Means*, são aplicadas duas métricas, MSE e PSNR. Como o próprio nome sugere, a métrica MSE é utilizada para calcular o erro quadrático médio da diferença dos valores dos *pixels* da imagem original e da imagem filtrada. Assim, quanto mais o valor resultante do cálculo da métrica MSE se aproximar de 0, pode-se dizer que a imagem filtrada se aproxima da imagem original.

Da mesma forma, a métrica PSNR é calculada a partir da imagem filtrada em relação a imagem original. Porém, de maneira inversa, a métrica PSNR calcula o pico da relação sinal-ruído. Isso significa que quanto maior o valor resultante da métrica PSNR, pode-se dizer que a imagem filtrada se aproxima da imagem original.

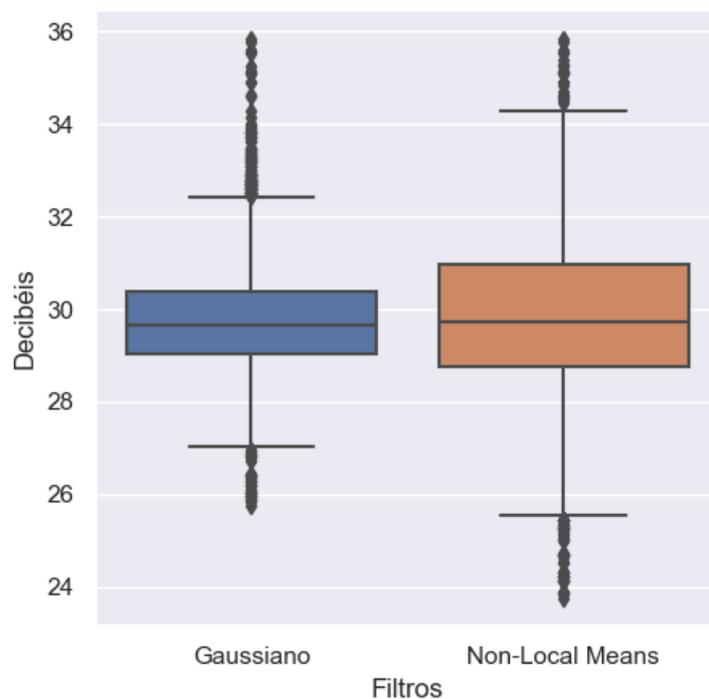
Vale ressaltar que optou-se pelo cálculo tanto da métrica MSE quanto da métrica PSNR para que fosse possível avaliar, com a mínima margem de erro, o processo de filtragem deste projeto de pesquisa. Assim, a Figura 20 apresenta o gráfico em linhas do resultado do cálculo da métrica MSE do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e *Non-local Means*.

Figura 20 – Resultado do cálculo da métrica MSE do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e *Non-local Means*.



A Figura 21 apresenta o gráfico em linhas do resultado do cálculo da métrica PSNR do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e *Non-local Means*.

Figura 21 – Resultado do cálculo da métrica PSNR do processo de filtragem do ruído do tipo Gaussiano por meio dos filtros Gaussiano e *Non-local Means*.



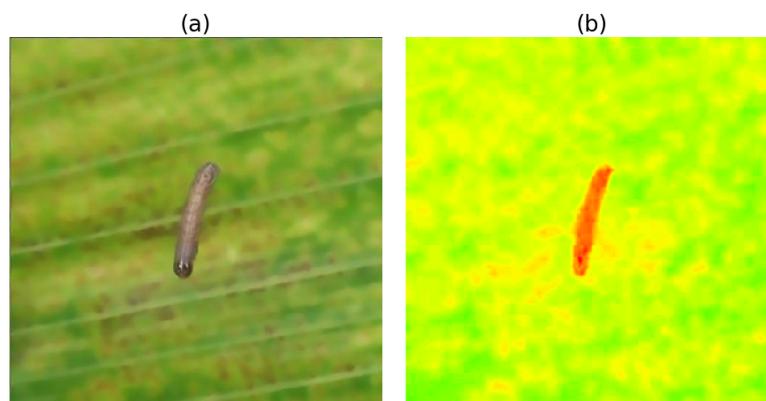
A partir da análise dos gráficos 20 e 21 é possível concluir que o filtro *Non-local Means* obteve um resultado superior ao filtro Gaussiano.

Assim, para o caso em questão e com base na Figura 6, ilustrada no Capítulo 2, o valor de degradação H é igual a 1, tendo em vista que houve apenas o emprego de filtragem espacial, ou seja, para a melhoria da relação (sinal/ruído). Portanto, não houve a necessidade do trabalho com funções especiais como melhoria de degradação nas informações geométricas.

3.3 Resultados obtidos por meio da etapa de operação de conversão de padrões de cores

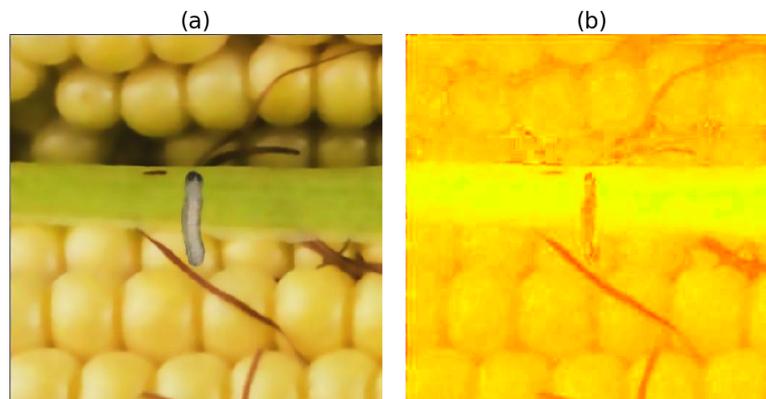
A Figura 22 ilustra o resultado da conversão do padrão de cor RGB para o padrão HSV, destacando o mapa de H. É possível verificar que, neste caso, a segmentação Lagarta do Cartucho (*Spodoptera frugiperda*), objeto, em relação a folha, fundo, pode ser realizada a partir da aplicação de apenas um limiar data a diferença entre os valores dos *pixels* tanto da lagarta como da folha.

Figura 22 – Resultado da conversão do padrão de cor RGB para o padrão HSV. (a) Imagem original, (b) mapa de H.



Porém, analisando a Figura 23, é possível verificar que o processo de segmentação da imagem pelo mapa de H não resultaria em uma segmentação de forma satisfatória, uma vez que os valores dos *pixels* tanto da lagarta quanto da espiga, são praticamente iguais.

Figura 23 – Resultado da conversão do padrão de cor RGB para o padrão HSV de uma imagem com fundo de espiga de milho. (a) Imagem original, (b) mapa de H.



Dada das devidas restrições verificadas em ensaios realizados com o mapa de H do padrão de cor HSV, decidiu-se por realizados ensaios também com o padrão de cor CIE $L^*a^*b^*$, com o objetivo de obter um processo de segmentação ideal. As Figuras 24 e 25 ilustram, respectivamente, a conversão do padrão de cor RGB para o padrão CIE $L^*a^*b^*$ de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma folha e em uma espiga, destacando-se os componentes a^* e b^* .

Figura 24 – Resultado da conversão do padrão de cor RGB para o padrão CIE $L^*a^*b^*$ de uma imagem com fundo de folha. (a) Imagem original, (b) mapa de a^* e (c) mapa de b^* .

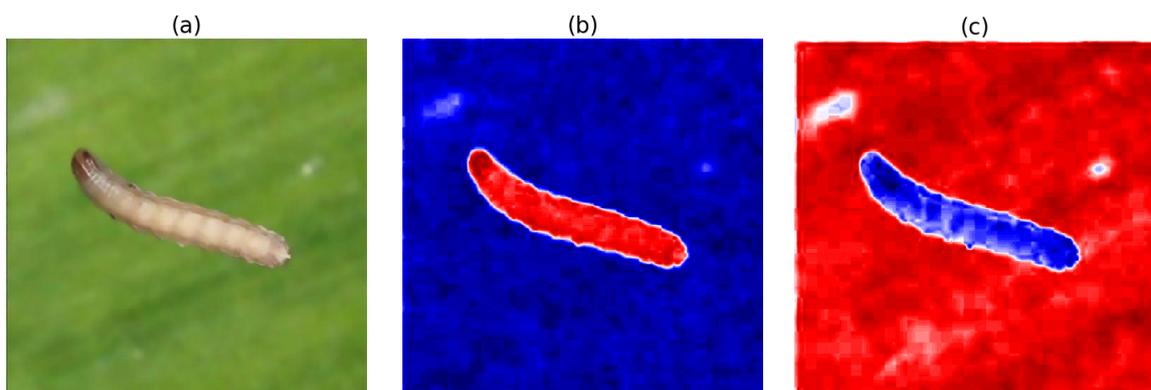
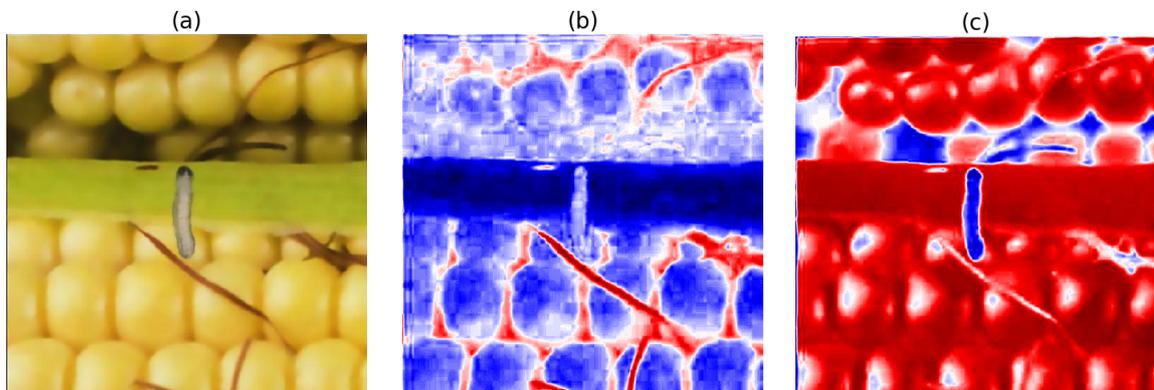


Figura 25 – Resultado da conversão do padrão de cor RGB para o padrão CIE L*a*b de uma imagem com fundo de espiga. (a) Imagem original, (b) mapa de a* e (c) mapa de b*.



Analisando a Figura 24, é possível verificar que os mapas de a* e b* referentes a imagem da lagarta em uma folha, permitem que o processo de segmentação seja executado de forma satisfatório. Já em relação a Figura 25, é possível verificar que utilizando apenas o mapa de b* obtém-se um processo de segmentação onde é possível remover a espiga (fundo) sem que a lagarta (objeto) seja afetada pela segmentação.

3.3.1 Avaliação da etapa de operação de conversão de padrões de cores

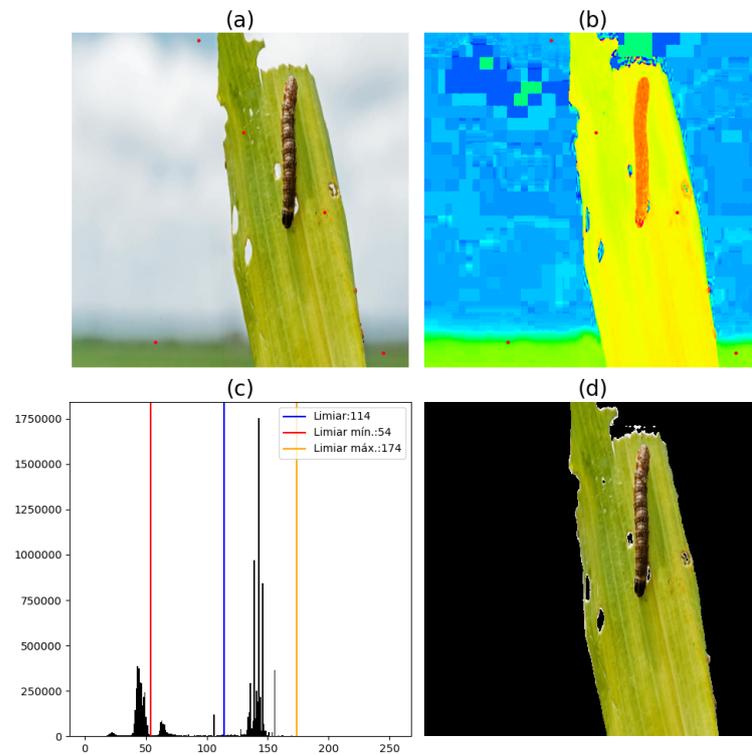
Conforme exposto nas seções anteriores, é possível concluir que a operação de conversão do padrão de cor RGB para o padrão CIE L*a*b* permitirá que a etapa de segmentação da imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) seja executada de forma a remover tanto a folha quanto a espiga (fundo), de forma a lagarta (objeto) não seja afetada pelo processo de segmentação, ou que seja o mínimo possível.

3.4 Resultados obtidos por meio da etapa de segmentação de imagens

A segmentação de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) utilizando a técnica de limiarização e *pixels* sementes foi realizada com base no pseudocódigo 7. Em referência ao cálculo dos limiares mínimo e máximo, foi utilizada proporção de $1,5\sigma$ e 2σ para imagens com fundo de folhas. Para imagens com fundo de espigas de milho, foram consideradas as proporção de $0,60\sigma$ e $0,70\sigma$.

A Figura 26 ilustra processo de segmentação por meio de limiarização e *pixels* sementes de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma folha, obtida no padrão de cor RGB (Figura 26 (a)), o mapa de H da respectiva imagem convertida para o padrão de cor

Figura 26 – Resultado da segmentação por limiarização e *pixels* sementes com proporção de $1,5\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H e (d) imagem segmentada.



HSV (Figura 26 (b)), o histograma do mapa de H (Figura 26 (c)) e a imagem no padrão de cor RBG segmentada (Figura 26 (d)) em função do mapa de H, utilizando a proporção de $1,5\sigma$.

A Figura 27 ilustra processo de segmentação por meio de limiarização e *pixels* sementes de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma folha, obtida no padrão de cor RGB (Figura 27 (a)), o mapa de H da respectiva imagem convertida para o padrão de cor HSV (Figura 27 (b)), o histograma do mapa de H (Figura 27 (c)) e a imagem no padrão de cor RBG segmentada (Figura 27 (d)) em função do mapa de H, utilizando a proporção de 2σ .

A Figura 28 ilustra processo de segmentação por meio de limiarização e *pixels* sementes de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma espiga, obtida no padrão de cor RGB (Figura 28 (a)), o mapa de H da respectiva imagem convertida para o padrão de cor HSV (Figura 28 (b)), o histograma do mapa de H (Figura 28 (c)) e a imagem no padrão de cor RBG segmentada (Figura 28 (d)) em função do mapa de H, utilizando a proporção de $0,60\sigma$.

A Figura 29 ilustra processo de segmentação por meio de limiarização e *pixels* sementes de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma espiga, obtida no padrão de cor RGB (Figura 29 (a)), o mapa de H da respectiva imagem convertida para o padrão de cor HSV (Figura 29 (b)), o histograma do mapa de H (Figura 29 (c)) e a imagem no padrão de cor RBG segmentada (Figura 29 (d)) em função do mapa de H, utilizando a proporção de $0,70\sigma$.

Figura 27 – Resultado por segmentação por limiarização e *pixels* sementes com proporção de 2σ . (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.

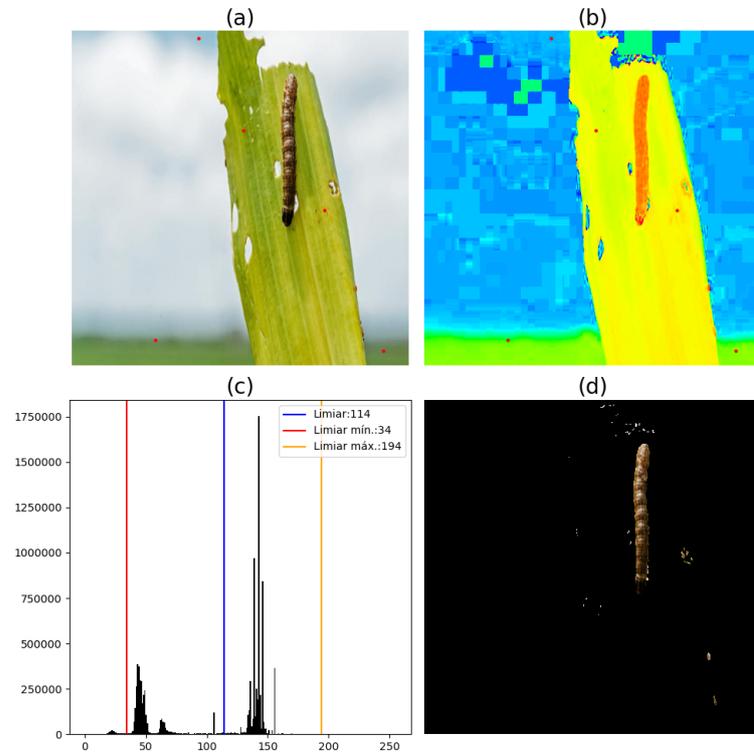


Figura 28 – Resultado da segmentação por limiarização e *pixels* sementes com proporção de $0,6\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.

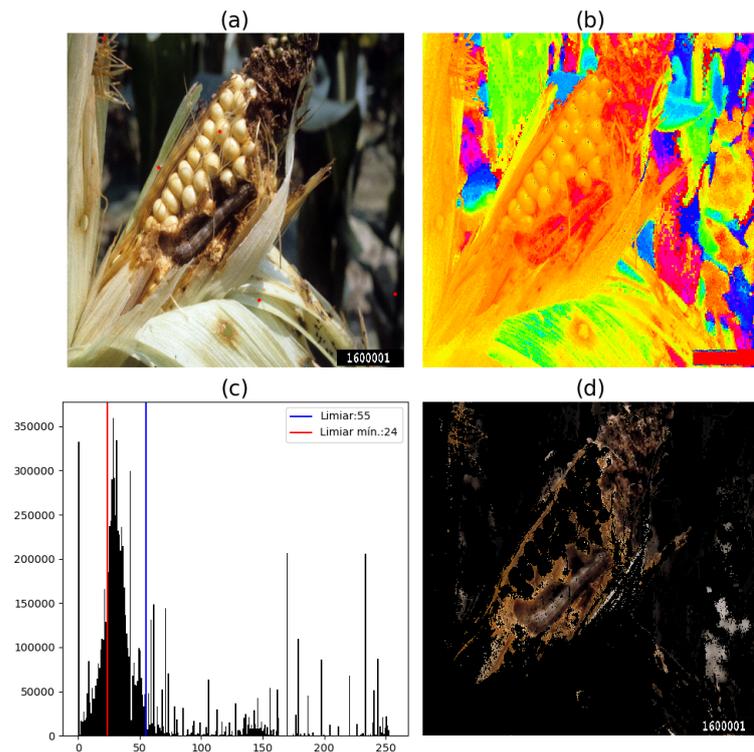
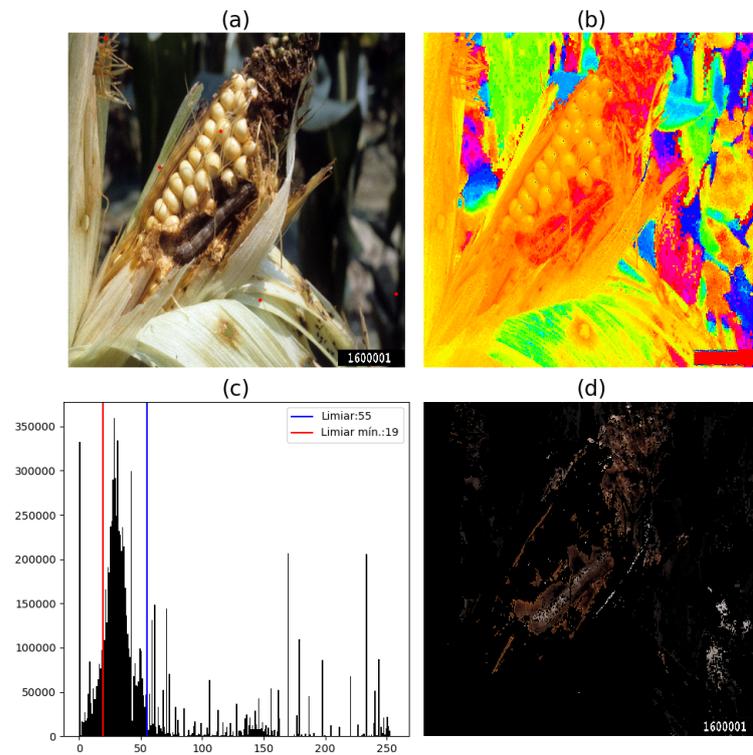


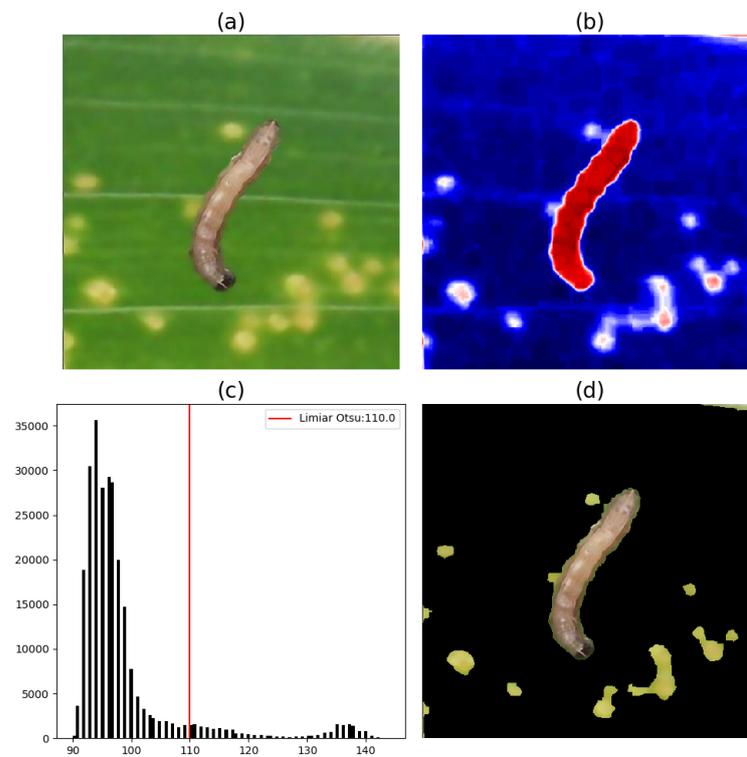
Figura 29 – Resultado da segmentação por limiarização e *pixels* sementes com proporção de $0,7\sigma$. (a) Imagem original, (b) mapa de H, (c) histograma do mapa de H, (d) imagem segmentada.



Conforme exposto no Capítulo anterior, a aplicação método de Otsu neste projeto de pesquisa visa a automação do processo de segmentação de imagens. Assim, essa etapa do método é baseada no pseudocódigo 8 elaborado para segmentação de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) em folhas. Para imagens da lagarta em espigas de milho, o método foi baseado no pseudocódigo 9.

A Figura 30 ilustra o resultado da segmentação pelo método de Otsu de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em uma folha. Essa imagem foi obtida por meio da técnica de *data augmentation*. Primeiramente, a imagem no padrão de cor RGB (Figura 30 a) foi convertida para o padrão CIE $L^*a^*b^*$. Conforme descrito no pseudocódigo 8, foi utilizado, inicialmente, o componente a^* (identificado na Figura 30 (a) como mapa de a^*) para a segmentação.

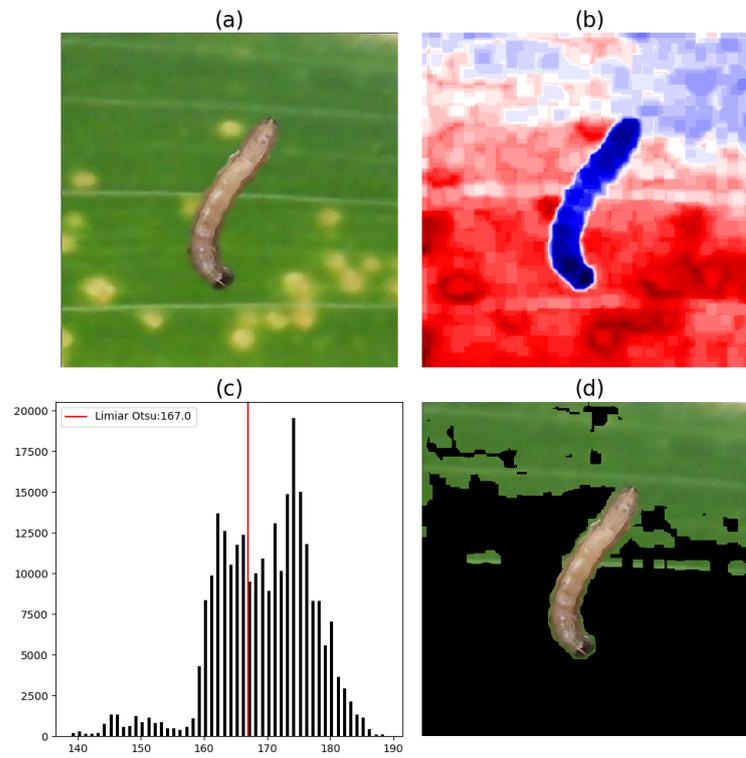
Figura 30 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) mapa de a^* , (c) histograma do mapa de a^* e (d) resultado da segmentação pelo mapa de a^* .



Ao analisar o histograma do mapa de a^* (Figura 30 c) observa-se que os *pixels* de menor valor são referentes aos *pixels* da folha, sendo representados pela cor azul na Figura 30 (b). De maneira oposta, os de *pixels* de maior valor representam a os *pixels* da lagarta (vermelho), assim como, de alguma anomalia presente folha. Portanto, para segmentar a lagarta do restante da imagem, foram considerados apenas os *pixels* com valores acima do limiar obtido pelo método de Otsu, 110.

Entretanto, a partir dos ensaios realizados para a segmentação de imagens pelo método de Otsu da Lagarta do Cartucho (*Spodoptera frugiperda*) em folhas utilizando apenas o componente a^* do padrão de cor CIE $L^*a^*b^*$, apesar de se mostrar eficiente, foi identificada a necessidade de uma segunda etapa de segmentação. Foram então realizados ensaios a partir do componente b^* . A Figura 31 ilustra o resultado da segmentação pelo método de Otsu da mesma imagem da Figura 30, porém, agora utilizando o componente b^* .

Figura 31 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) mapa de b^* , (c) histograma do mapa de b^* e (d) resultado da segmentação pelo mapa de b^*

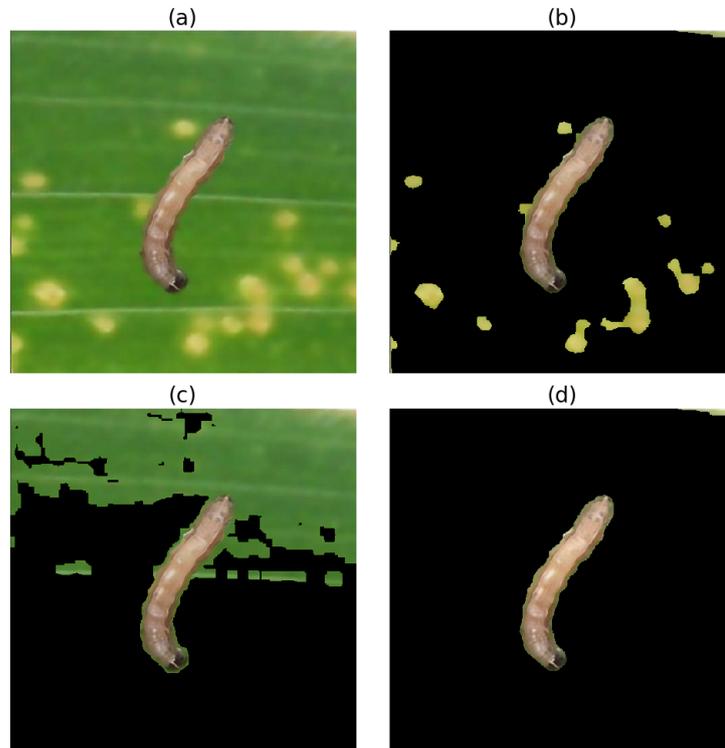


A partir da análise do histograma na Figura 31 c, observa-se que o *pixels* de menor valor representam os *pixels* da lagarta, assim como, de algumas partes da folha. Portanto, para segmentar a lagarta do restante da imagem, foram considerados apenas os *pixels* com valores abaixo do limiar obtido pelo método de Otsu, 167.

Assim como no processo de segmentação utilizando apenas o componente a^* , a segmentação pelo componente b^* também resultou em uma imagem ainda com partes da folha. Entretanto, foi possível verificar que, no domínio espacial da imagem, as partes não segmentadas referentes a folha na segmentação pelo componente a^* , não pertenciam às mesmas localizações espaciais do resultado da segmentação pelo componente b^* . Portanto, a partir dos resultados obtidos pela segmentação da imagem da lagarta em uma folha pelos componentes a^* e b^* do padrão de cor CIE $L^*a^*b^*$, ilustrados pelas Figuras 30 (d) e 31 (d), respectivamente, foi realizada uma nova etapa de segmentação, desta vez, na forma de intersecção.

O resultado da segmentação por intersecção acima descrito, é ilustrado pela Figura 32.

Figura 32 – Resultado da segmentação pelo método de Otsu de uma imagem da lagarta em uma folha. (a) Imagem original, (b) resultado da segmentação pelo mapa de a^* , (c) resultado da segmentação pelo mapa de a^* e (d) resultado final da segmentação.

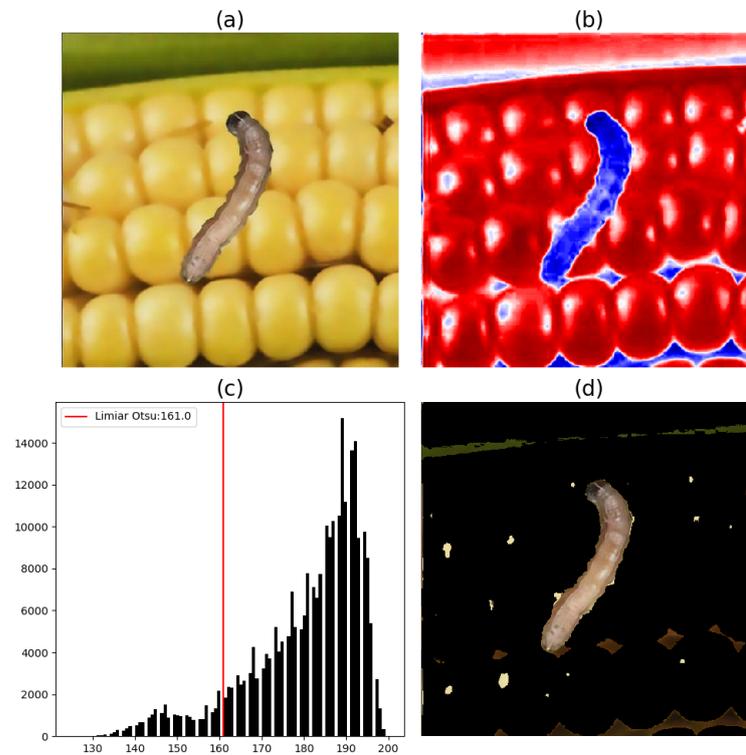


Observa-se pelo resultado da segmentação ilustrado na Figura 32 (c), que o resultado da segmentação por intersecção mostrou-se eficiente na segmentação de imagens da lagarta em folhas.

Para a segmentação da Lagarta do Cartucho (*Spodoptera frugiperda*) em espigas de milho por meio do método de Otsu, os resultados dos ensaios realizados para segmentação a partir do componente a^* do padrão de cor CIE $L^*a^*b^*$ se mostraram ineficientes para o processo de segmentação, dado o processo de conversão do padrão de cor RGB para o padrão CIE $L^*a^*b^*$, conforme abordado no Capítulo 3.

Dessa forma, para a segmentação da lagarta em espigas de milho, utilizado apenas o componente b^* . A Figura 33 ilustra o processo de segmentação de imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) em espigas de milho por meio do método de Otsu.

Figura 33 – Resultado da segmentação de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) pelo método de Otsu. (a) Imagem original, (b) mapa de b^* , (c) imagem segmentada.



A partir da análise do histograma do mapa de b^* (Figura 33 c), é possível verificar que os *pixels* de maior valor representam os *pixels* da espiga de milho no componente b^* (Figura 33 b). Assim, para a segmentação da lagarta do restante da imagem, foram considerados os *pixels* com valores abaixo do limiar obtido pelo método de Otsu, 161.

O resultado da segmentação ilustrado na Figura 33 (c), mostra que partes da espiga não foram totalmente segmentados. Esse resultado já era esperado tanto para segmentação de imagens com espigas, como em imagens contendo folhas, uma vez que durante os ensaios realizados para validação da etapa de segmentação do método proposto, foi constatado a complexidade da formação das imagens no que se refere a variação dos valores dos *pixels* que compõem tanto a Lagarta do Cartucho (*Spodoptera frugiperda*) quanto o fundo das imagens.

3.4.1 Avaliação da etapa de segmentação de imagens

A partir dos resultados obtidos pela segmentação por limiarização e *pixels* sementes utilizando o mapa de H do padrão de cor HSV, é possível avaliar que a técnica proposta foi eficaz em segmentar imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) em folhas, entretanto, para a segmentação da lagarta em espigas de milho não foi possível obter os mesmos resultados, uma vez que os valores dos *pixels* referentes a lagarta e a espiga de milho são bem próximos.

A automatização da etapa de segmentação de imagens pelo método de Ótsu a partir do

padrão de cor CIE $L^*a^*b^*$ se mostrou mais eficaz no processo de segmentação de imagens com fundo contendo folhas e espigas de milho em relação a limiarização e *pixels* sementes. Em relação a segmentação de imagens da lagarta sobre folhas, a divisão em três etapas, primeiramente a partir do componente a^* , depois pelo componente b^* e por fim, segmentar os resultados em uma segmentação no modelo de intersecção das imagens, mostrou muito eficiente e de alta performance.

A segmentação de imagens da lagarta em espigas de milho, apesar de não apresentar resultados semelhantes a segmentação de imagens da lagarta em folhas, apresentou resultados significativos dada a complexidade da formação dessas imagens e, conforme será apresentado no decorrer deste trabalho, foi eficaz no que se refere aos resultados esperados na validação do método proposto.

3.5 Resultados obtidos por meio da etapa de extração de características

A aplicação do descritor HOG foi executada de acordo com o pseudocódigo 10, de modo a extrair características de textura da Lagarta do Cartucho (*Spodoptera frugiperda*). A Tabela 10 exhibe a parametrização do descritor HOG.

Tabela 10 – Valores dos parâmetros do descritor HOG.

Parâmetros	Valores
<i>Pixels</i> por célula	16x16
Células por bloco	2x2
Número de orientações	9
Vetor de características	Verdadeiro
<i>Transform Sqrt</i>	Falso

Para a execução do descritor HOG as imagens, previamente segmentadas, foram redimensionadas de modo a obter uma resolução espacial de 256x256 *pixels*. Uma vez aplicados os parâmetros do descritor HOG nas imagens redimensionadas foi possível gerar um vetor de características de 8100 posições, da forma $V_{[HOG]}$, para cada imagem da Lagarta do Cartucho, conforme ilustrado na Figura 34

Uma vez obtido o vetor de características $V_{[HOG]}$ por meio da aplicação do descritor HOG, foi então aplicado o descritor de momentos invariantes de Hu nas imagens da Lagarta do Cartucho, conforme demonstrado no pseudocódigo 11 e ilustrado na Figura 35.

Figura 34 – Resultado da extração da características de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) pelo descritor HOG. (a) Imagem da lagarta segmentada, (b) imagem HOG da lagarta segmentada.

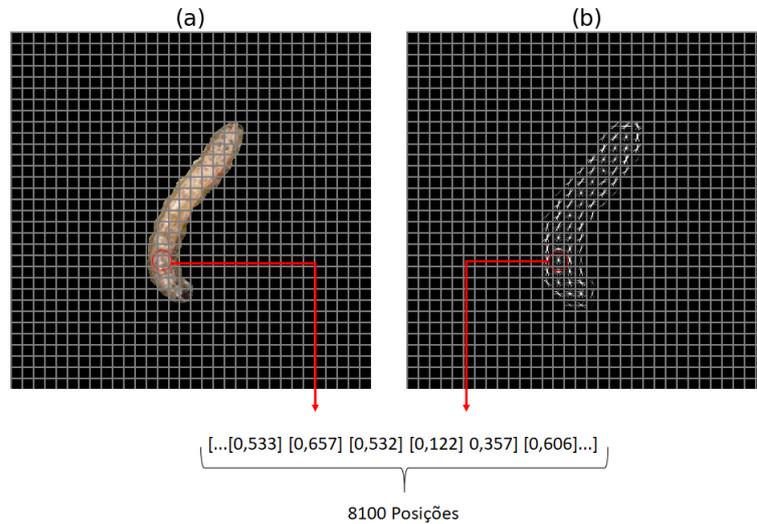
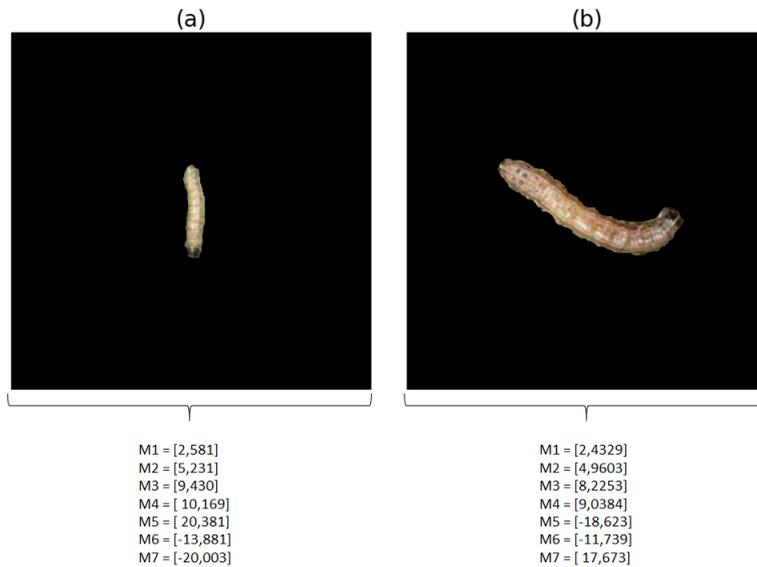


Figura 35 – Resultado da extração da características de uma imagem da Lagarta do Cartucho (*Spodoptera frugiperda*) pelos momentos invariantes de Hu. (a) Imagem da lagarta no instar 1 segmentada na posição vertical, (b) Imagem da lagarta no instar 3 segmentada na posição horizontal.



Assim, para cada imagem da Lagarta do Cartucho, foi gerado um vetor de características $V_{[Hu]}$, contendo os sete momentos invariantes de Hu, ou seja, características de tamanho de forma das lagartas. As características contidas nesses vetores permitirão a classificação da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento.

3.5.1 Avaliação da etapa de extração de características

Considerando os descritores aplicados nessa etapa, um se refere a características de textura (HOG) e outro se refere a forma e tamanho, ou seja, características geométricas. No que se refere ao desenvolvimento do método, ambos descritores foram considerados de modo a enriquecer o resultado do conjunto de classificadores, responsável por classificar os padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento.

Assim, após a extração de características pelo descritor HOG e pelo descritor dos momentos invariantes de Hu, os vetores de características $V_{[HOG]}$ e $V_{[Hu]}$ foram concatenados de maneira a gerar um único vetor de características $V_{[HOG,Hu]}$ com 8107 posições, conforme ilustrado na Figura 36.

Figura 36 – Exemplo do vetor de características $V_{[HOG,Hu]}$.

```
Vetor[HOGHu] = [[... 0,533 0,658 0,533 0,245 0,607 0,245 0,123 0,358 0,607 0,079 0,230 0,447 0,223
0,158 0,407 0,447 0,447 0,316 0,346 0,245 0,453 0,453 0,453 0,453 0,467 0,467
0,467 0,467 0,192 0,303 0,003 0,015 0,003 0,019 0,008 0,009 0,021 0,003 0,004
0,152 0,170 0,649 0,259 0,649 0,148 0,091 0,039 0,070 0,001 0,006 0,001 0,000
0,001 0,010 0,057 0,064 0,390 0,097 0,300 0,055 0,034 0,015 0,026 0,221 0,348
0,390 0,382 0,390 0,257 0,193 0,080 0,082 0,000 0,001 0,008 0,001 0,001 0,002
0,006 0,003 0,002 0,182 0,286 0,331 0,315 0,331 0,212 0,159 0,066 0,067 0,275
0,137 0,037 0,041 0,143 0,200 0,331 0,331 0,331 0,002 ...], [2,563 5,200 10,325
10,118 -20,341 12,756 21,600 ]]
```

Por fim, uma vez que os valores referentes a características de textura e de forma e tamanho estão em escalas diferentes, foi necessário realizar a normalização do vetor $V_{[HOG,Hu]}$ de acordo com o pseudocódigo 12, afim de obter um vetor normalizado. Na intenção de gerar um banco de dados de características da Lagarta do Cartucho (*Spodoptera frugiperda*), os vetores $V_{[HOG,Hu]}$ referentes a cada imagem foram salvos em disco.

3.6 Resultados obtidos por meio da etapa de redução de dimensionalidade do vetor de características

Para que fosse possível estabelecer o número de componentes principais, foi realizado o cálculo da porcentagem da variância de cada componente e dividido pela soma acumulada da variância explicada. Assim, foi utilizada a biblioteca *sklearn.PCA* da linguagem *Python*.

Portanto, para a redução da dimensionalidade do vetor de características de modo a permitir os treinamentos e testes do conjunto de classificadores SVM, foi considerado um valor no intervalo de 0,9 a 1, de modo que, a taxa ideal para se manter a variação dos dados originais para a etapa de classificação, é quando o número de autovetores corresponde a aproximadamente 95% da variância total (SWETS; WENG, 1996; BRITO, 2020).

Dessa forma, foi considerado número de 128 componentes principais, de modo a obter 98% da variabilidade dos dados originais. A Figura 37 ilustra o novo $Vetor_{[HOG,Hu]}$ obtido por meio da aplicação de PCA.

Figura 37 – Exemplo do vetor de características $Vetor_{[HOG,Hu]}$ obtido por meio da técnica de PCA.

$Vetor_{[HOG,Hu]} = [252,247$ 237,829 166,560 41,041 27,823 25,157 24,131 23,082 22,1012 19,779
 18,835 18,573 17,293 16,917 16,895 16,447 16,210 15,382 14,805 14,659 14,094
 13,981 13,823 13,404 13,114 12,843 12,594 12,356 12,239 12,026 11,903 11,749
 11,662 11,552 11,488 11,316 11,222 11,001 10,955 10,838 10,781 10,681 10,552
 10,494 10,377 10,351 10,292 10,284 10,172 10,094 10,024 9,993 9,878 9,791 9,769
 9,673 9,585 9,525 9,447 9,388 9,331 9,276 9,235 9,220 9,143 9,076 9,044 9,015 8,974
 8,925 8,885 8,784 8,767 8,710 8,700 8,665 8,613 8,551 8,491 8,446 8,404 8,344 8,298
 8,270 8,250 8,225 8,194 8,118 8,088 8,070 8,019 7,950 7,903 7,859 7,815 7,795 7,770
 7,666 7,627 7,621 7,582 7,531 7,506 7,442 7,403 7,379 7,316 7,308 7,263 7,222 7,209
 7,177 7,126 7,095 7,042 7,013 6,998 6,953 6,943 6,912 6,882 6,813 6,787 6,763 6,733
 6,657 6,596 6,540]

3.6.1 Avaliação da etapa de redução de dimensionalidade do vetor de características

De forma a remover informações duplicadas e também informações não significativas para a etapa de treinamento do conjunto de classificadores SVM, de modo a permitir a classificação dos padrões da Lagarta do Cartucho (*Spodoptera frugiperda*), nessa etapa foi aplicada a técnica de redução da dimensionalidade do vetor de características por meio de PCA. Tal técnica permitiu a redução da dimensionalidade do vetor de características de 2280 para 128 componentes principais de características, mantendo aproximadamente 98% da variabilidade dos dados originais.

Outra alternativa para a etapa de redução da dimensionalidade do vetor de características seria a Análise de Discriminante Linear (LDA, do inglês *Linear Discriminant Analysis*). Entretanto, o fato deste trabalho tratar de um problema de classificação entre duas classes para cada classificador SVM, assim como PCA é considerada mais eficiente do ponto de vista de compactação dos dados e também pela análise dos resultados gerados (mantendo aproximadamente 98% de variabilidade dos dados originais), optou-se pelo uso de PCA.

3.7 Resultados obtidos por meio da etapa de treinamento do conjunto de classificadores SVM

Para a realização dos ensaios nessa etapa, foram considerados os classificadores SVM com função *kernel* linear e função *kernel* sigmoide. Para validação de cada classificador SVM,

não apenas a acurácia foi considerada mas também a precisão para a classificação do estágio alvo da Lagarta do Cartucho (*Spodoptera frugiperda*).

Os ensaios para treinamento do conjunto de classificadores SVM foram realizados conforme ilustrado pela Figura 14. Assim, para os ensaios realizados nessa etapa, foi considerada a proporção de 75% de vetores de características relacionados ao estágio de desenvolvimento específico que se deseja classificar e 25% foi selecionada de forma aleatória de forma a representar os outros estágios de desenvolvimento da Lagarta do Cartucho (*Spodoptera frugiperda*). Por exemplo, supondo que se deseja treinar o classificador SVM ‘Classificador do Estágio 1’, do total de 456 vetores de características referentes a este estágio, 25% (114 vetores) são vetores de características escolhidos de forma aleatória que representam outros estágios. É importante destacar que a proporção de 75% e 25% não está relacionada a proporção utilizada para as etapas de treinamento e teste dos classificadores.

Para as etapas de treinamento e teste dos classificadores, foram consideradas as proporções de 50% teste e 50% treinamento, 70% teste e 30% treinamento e 80% teste e 20% treinamento

De forma a seguir os critérios listados acima, foi utilizada a biblioteca da linguagem *Python sklearn.model.selection.train_test_split*. Essa biblioteca permite, de forma confiável, a separação das amostras para as etapas de teste e treinamento do conjunto de classificadores SVM.

A Tabela 11 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores com função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 50% para treinamento e teste.

Tabela 11 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 50% para treinamento e 50% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,82	0,63	0,71	183	0,59
Classificador do Estágio 2	0,81	0,69	0,75	183	0,63
Classificador do Estágio 3	0,82	0,82	0,82	183	0,71
Classificador do Estágio 4	0,83	0,78	0,81	183	0,70
Classificador do Estágio 5	0,80	0,85	0,83	183	0,71

As Figuras 38, 40, 42, 44 e 46 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 11. As Figuras 39, 41, 43, 45 e 47 ilustram as curvas ROC referentes à mesma tabela.

Figura 38 – Matriz de confusão do Classificador 1 com função *kernel* linear e proporção 50% teste 50% treinamento.

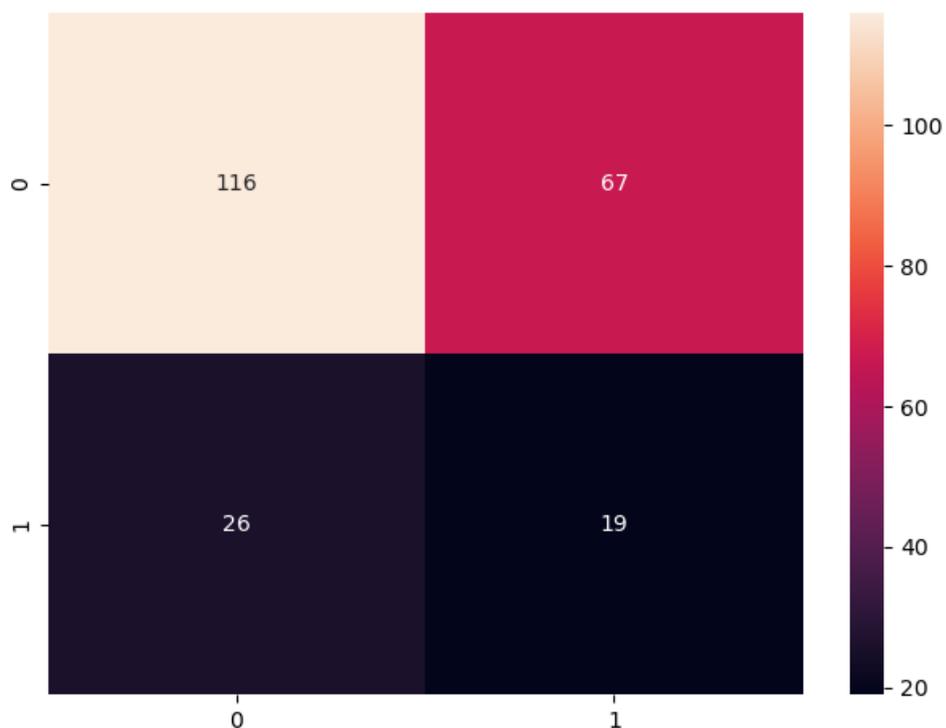


Figura 39 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 50% teste 50% treinamento.

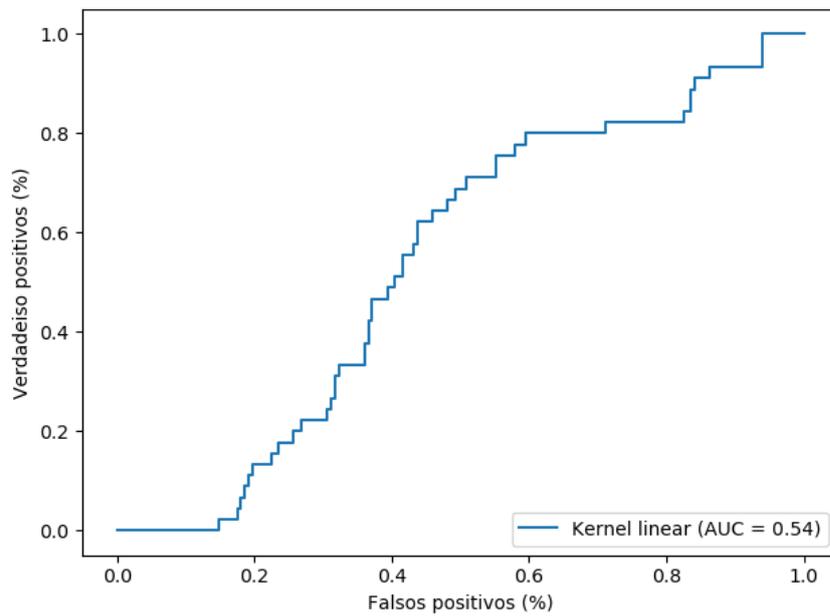


Figura 40 – Matriz de confusão do Classificador 2 com função *kernel* linear e proporção 50% teste 50% treinamento.

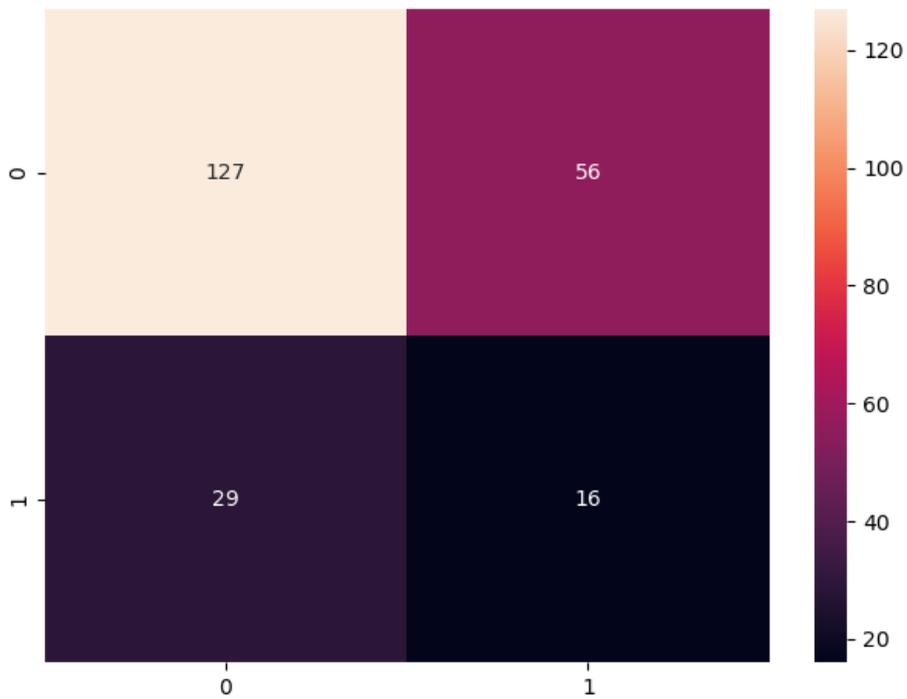


Figura 41 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 50% teste 50% treinamento.

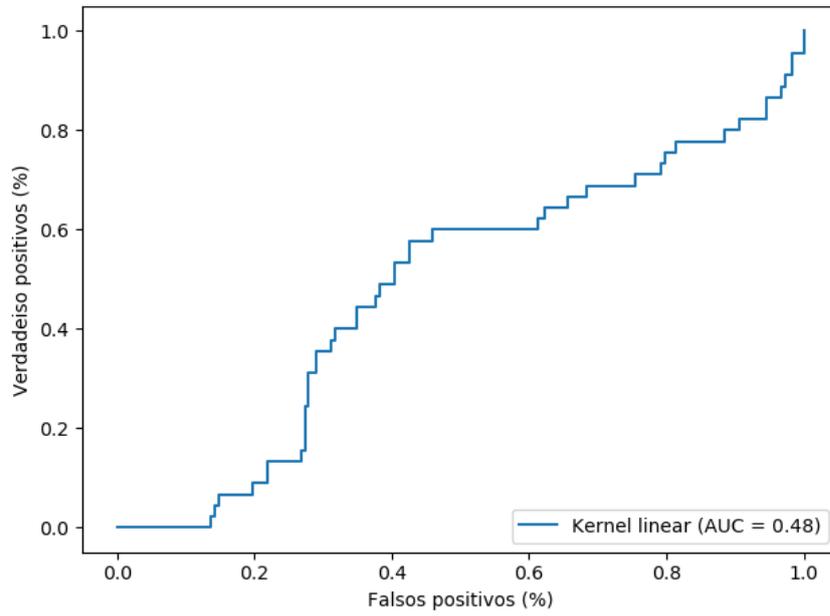


Figura 42 – Matriz de confusão do Classificador 3 com função *kernel* linear e proporção 50% teste 50% treinamento.

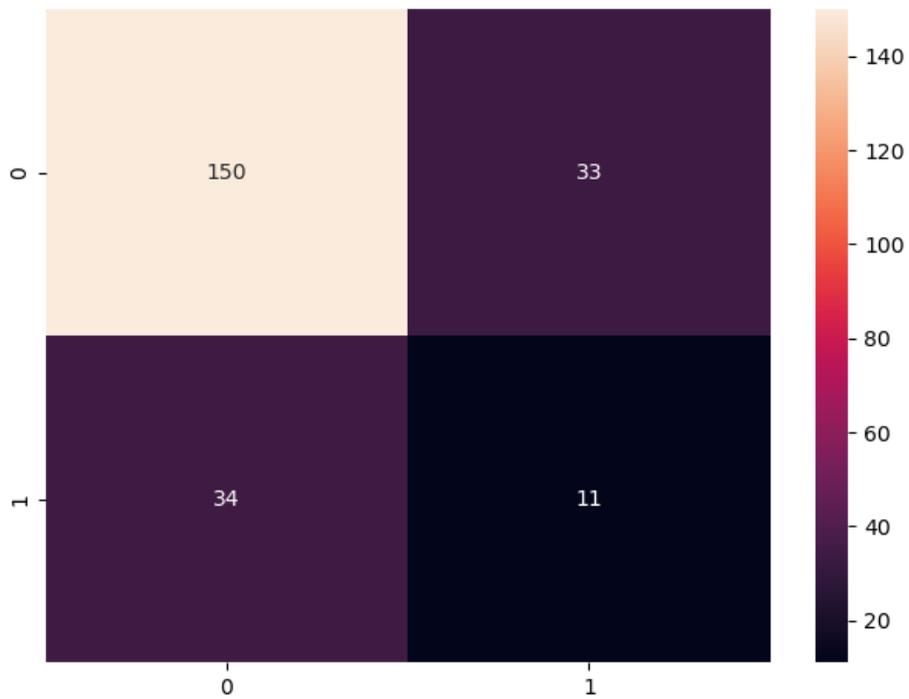


Figura 43 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 50% teste 50% treinamento.

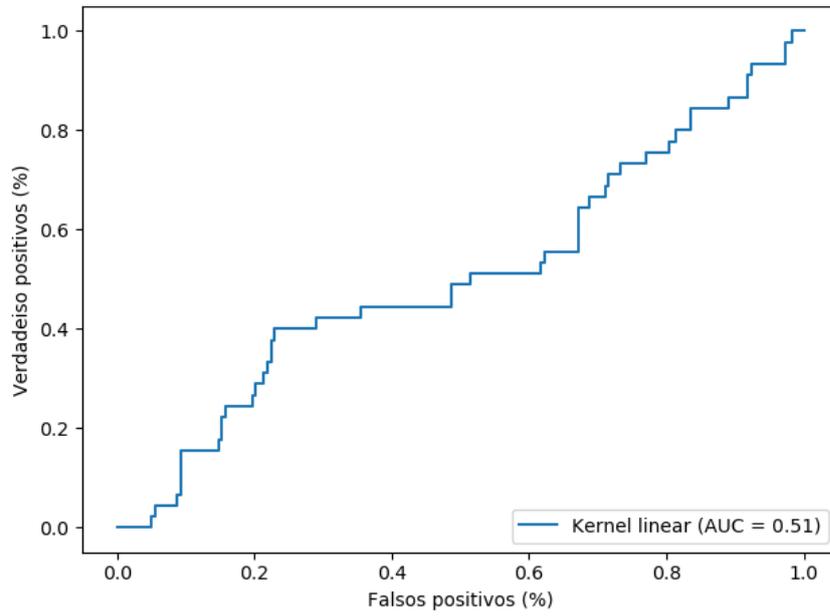


Figura 44 – Matriz de confusão do Classificador 4 com função *kernel* linear e proporção 50% teste 50% treinamento.

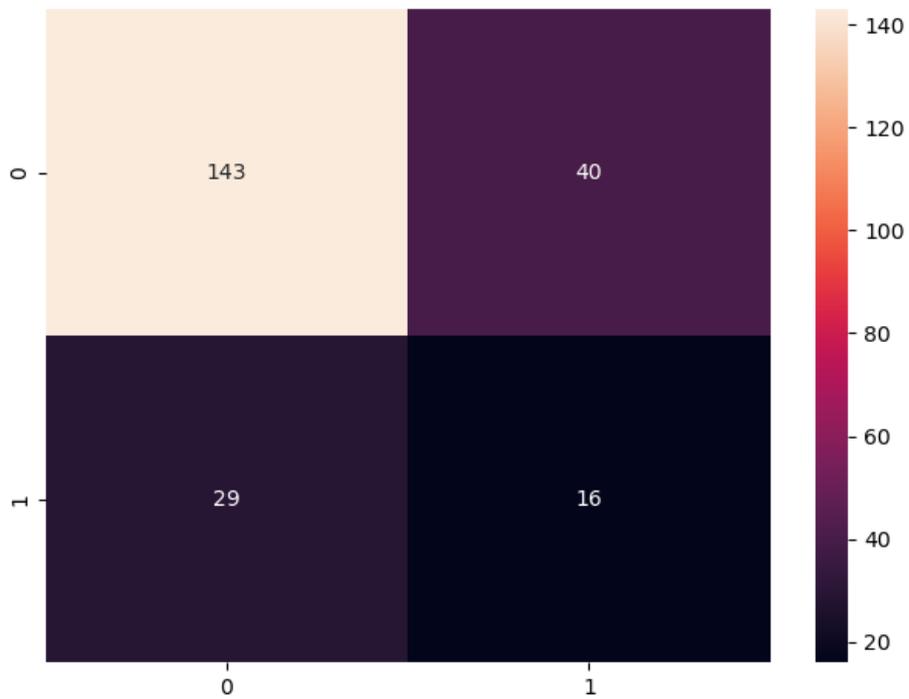


Figura 45 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 50% teste 50% treinamento.

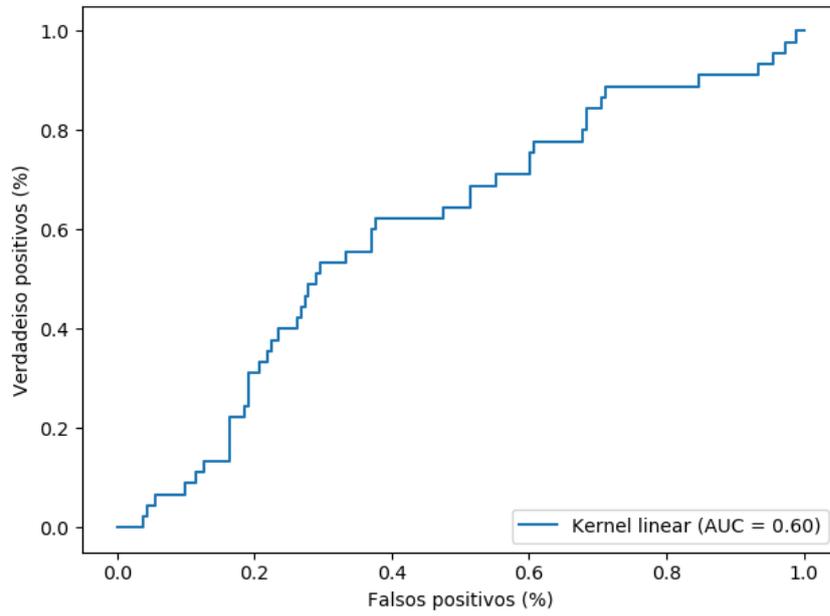


Figura 46 – Matriz de confusão do Classificador 5 com função *kernel* linear e proporção 50% teste 50% treinamento.

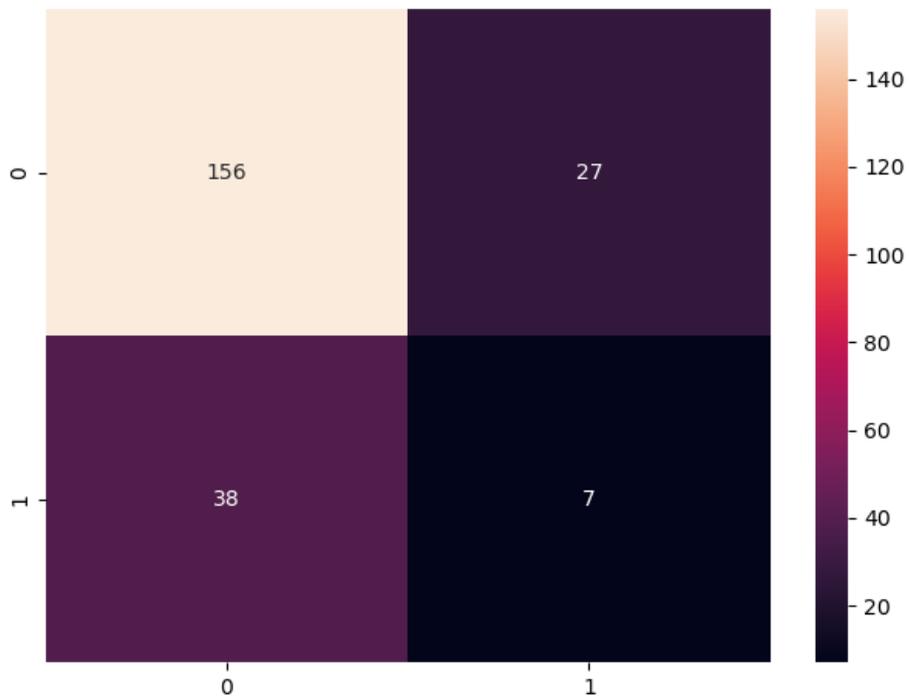
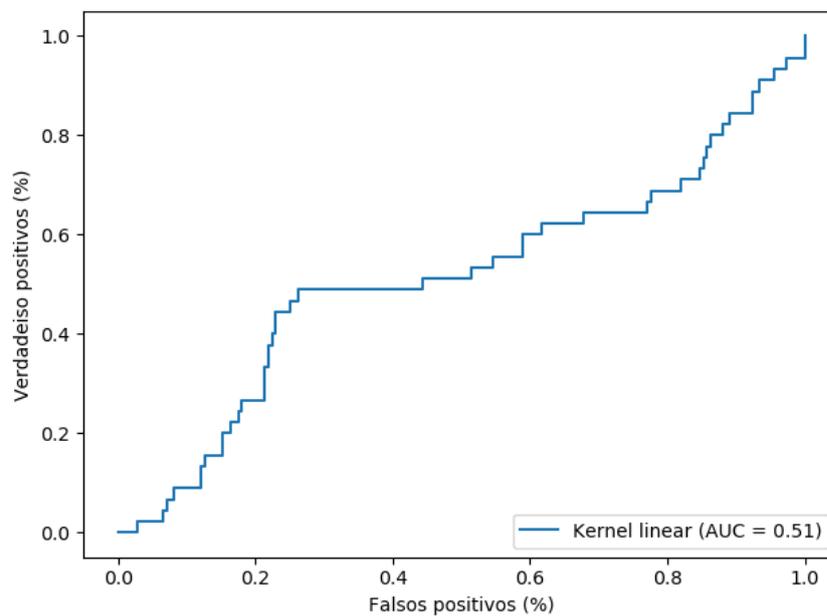


Figura 47 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 50% teste 50% treinamento.



A Tabela 12 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 50% para treinamento e teste.

Tabela 12 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 50% para treinamento e 50% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,81	0,63	0,71	183	0,59
Classificador do Estágio 2	0,83	0,69	0,75	183	0,64
Classificador do Estágio 3	0,73	0,79	0,83	183	0,73
Classificador do Estágio 4	0,85	0,79	0,82	183	0,72
Classificador do Estágio 5	0,85	0,72	0,78	183	0,68

As 48, 50, 52, 54, e 56 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 12. As Figuras 49, 51, 53, 57 e 57 ilustram as curvas ROC referentes à mesma tabela.

Figura 48 – Matriz de confusão do Classificador 1 com função *kernel* linear e proporção 50% teste 50% treinamento.

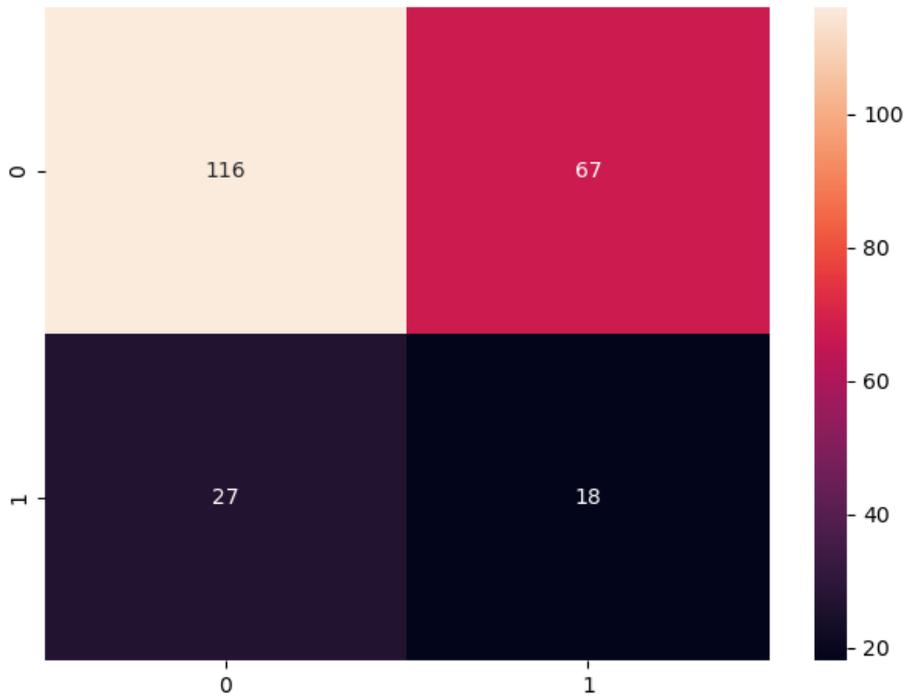


Figura 49 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 50% teste 50% treinamento.

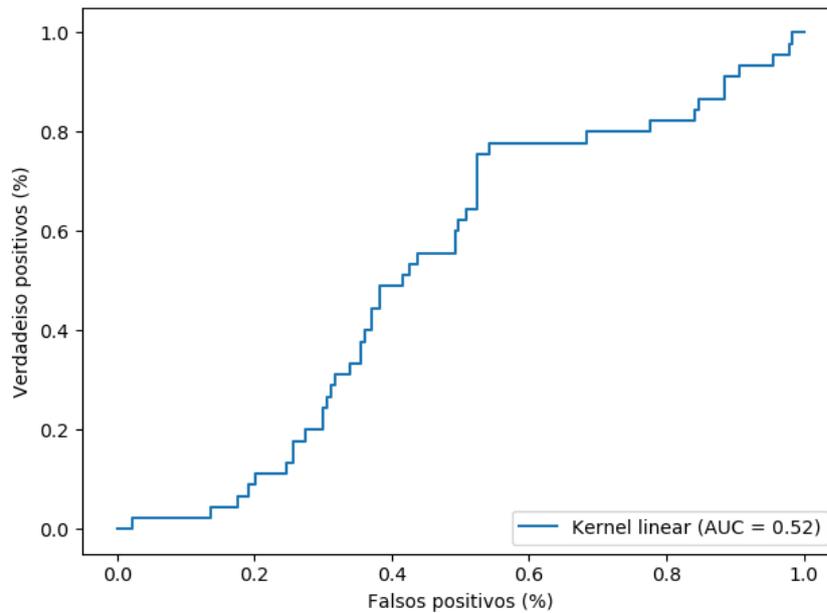


Figura 50 – Matriz de confusão do Classificador 2 com função *kernel* linear e proporção 50% teste 50% treinamento.

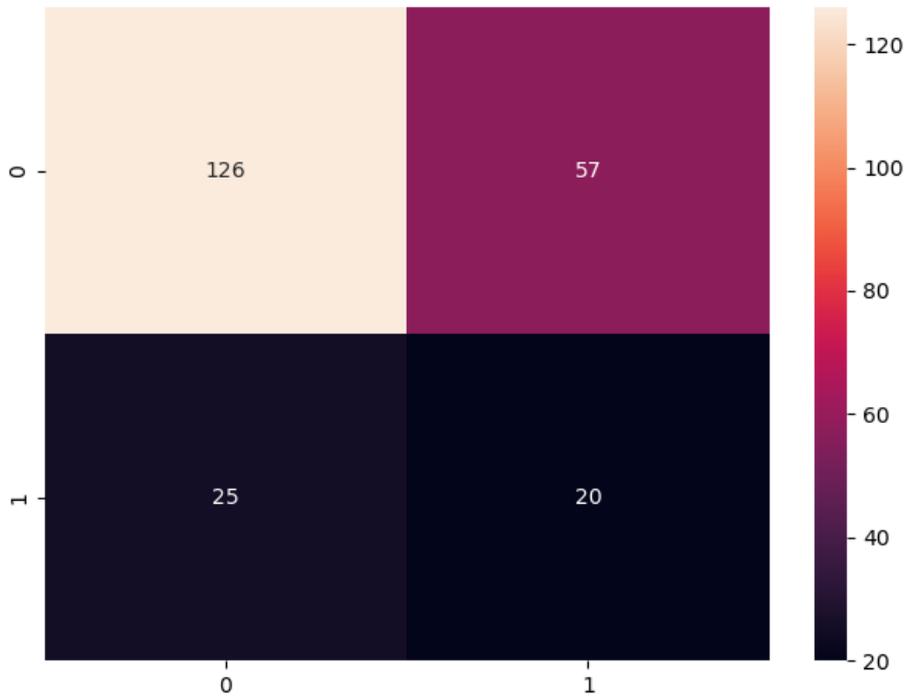


Figura 51 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 50% teste 50% treinamento.

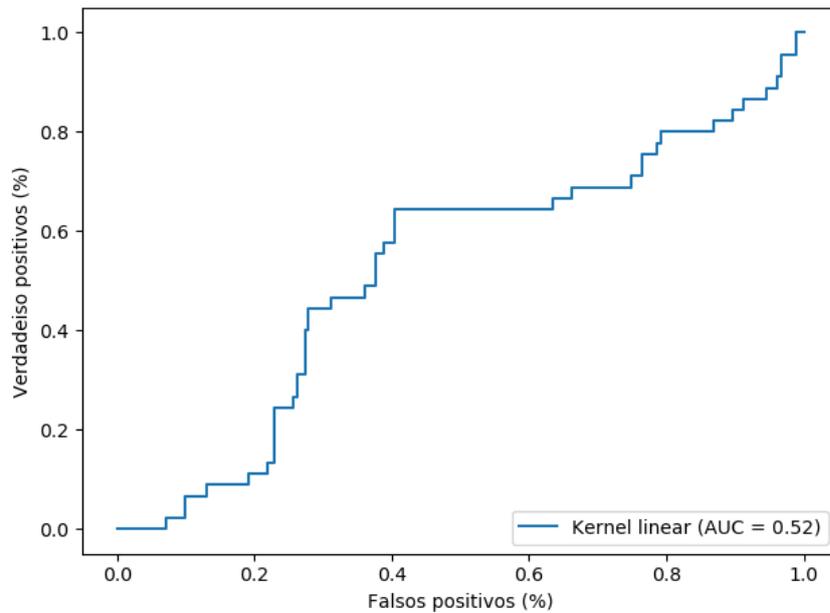


Figura 52 – Matriz de confusão do Classificador 3 com função *kernel* linear e proporção 50% teste 50% treinamento.

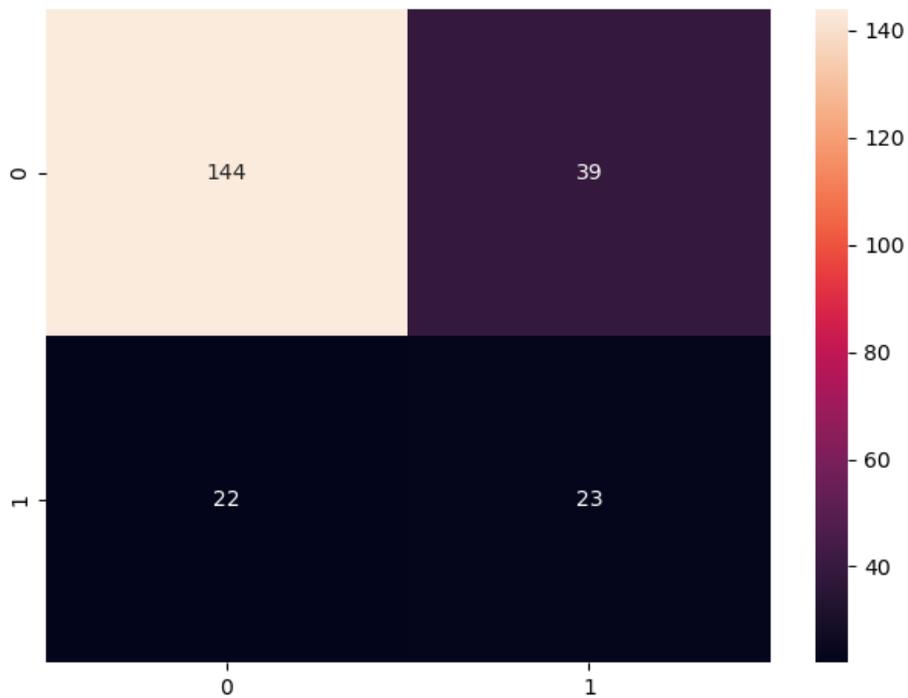


Figura 53 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 50% teste 50% treinamento.

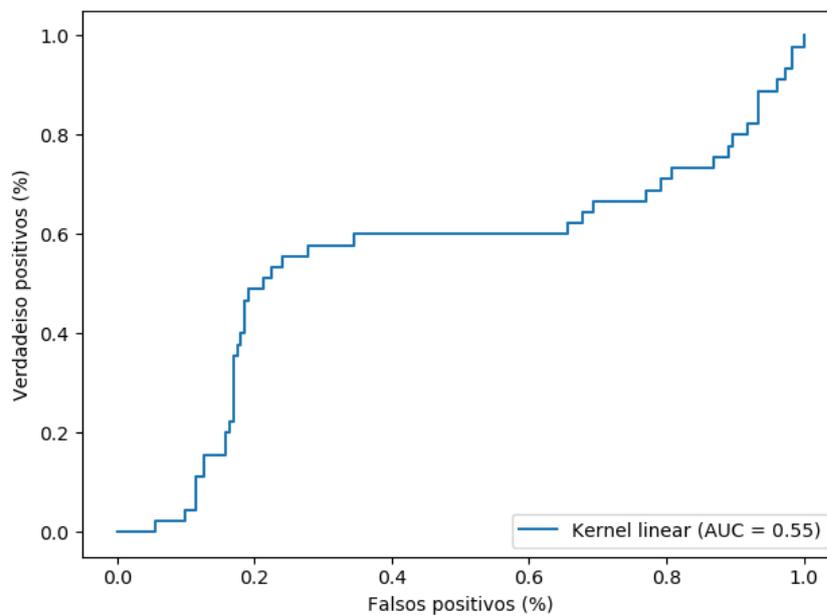


Figura 54 – Matriz de confusão do Classificador 4 com função *kernel* linear e proporção 50% teste 50% treinamento.

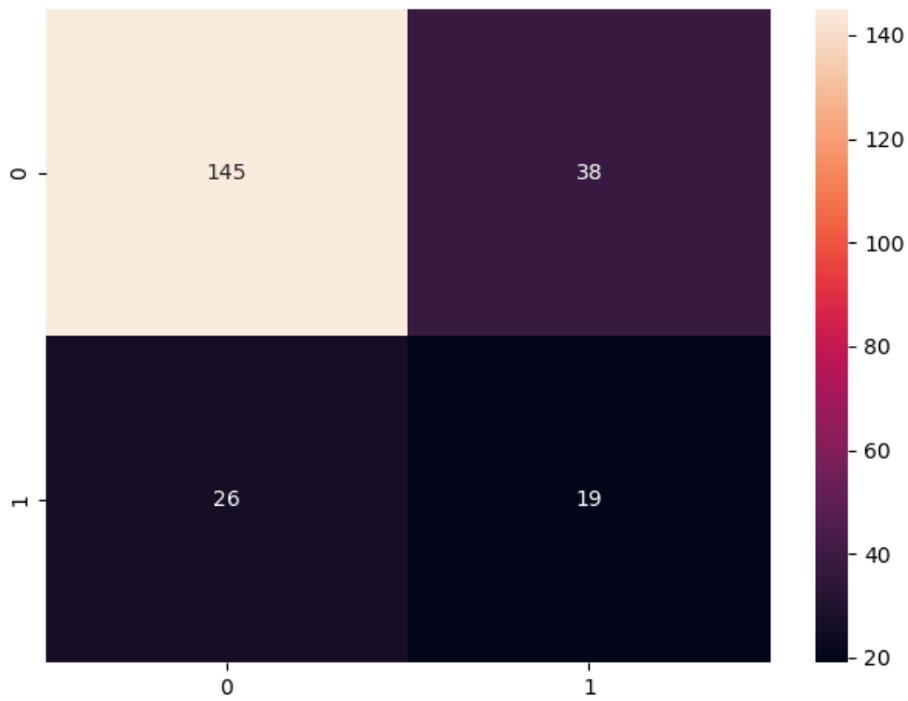


Figura 55 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 50% teste 50% treinamento.

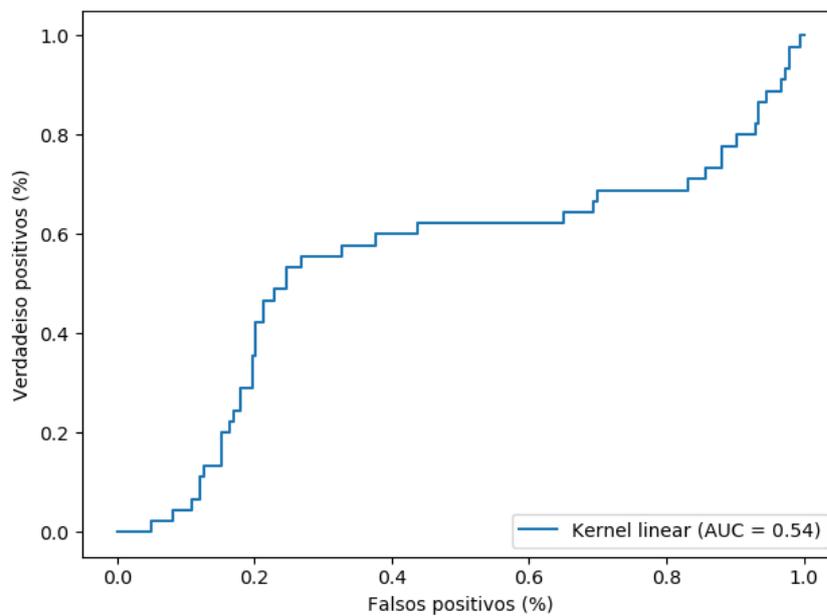


Figura 56 – Matriz de confusão do Classificador 5 com função *kernel* linear e proporção 50% teste 50% treinamento.

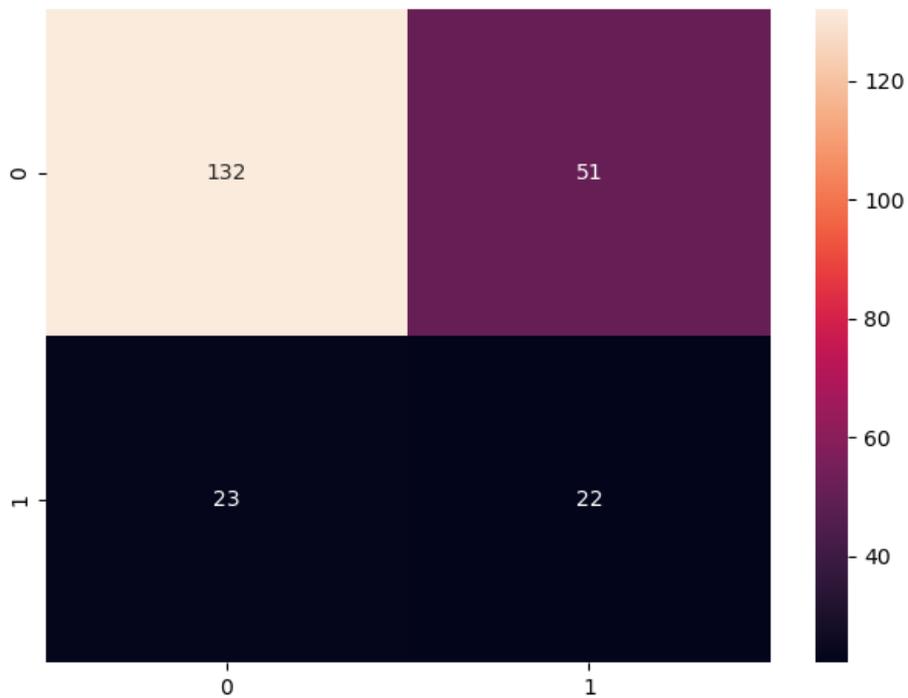
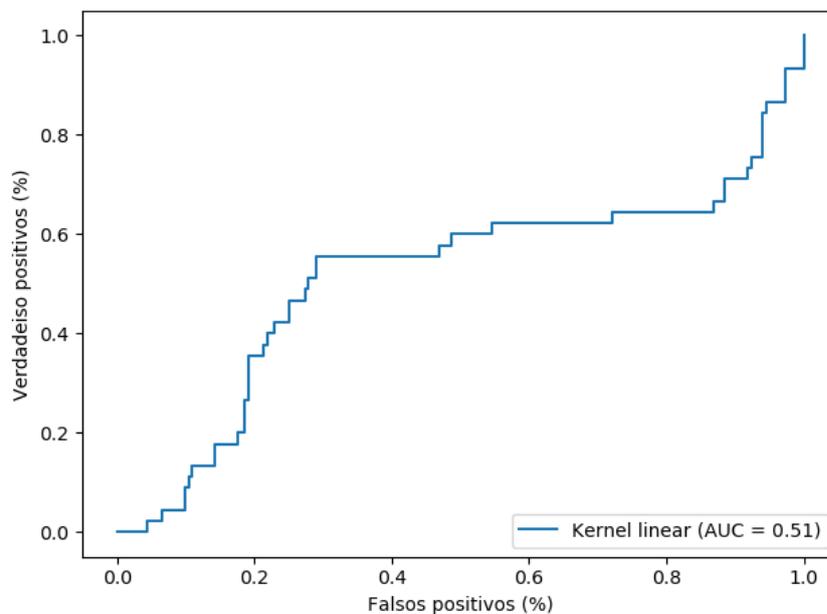


Figura 57 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 50% teste 50% treinamento.



A Tabela 13 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores com

função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 70% para treinamento e 30% teste.

Tabela 13 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 70% para treinamento e 30% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,75	0,75	0,75	106	0,61
Classificador do Estágio 2	0,73	0,68	0,71	106	0,56
Classificador do Estágio 3	0,78	0,86	0,82	106	0,71
Classificador do Estágio 4	0,79	0,80	0,80	106	0,69
Classificador do Estágio 5	0,76	0,73	0,74	106	0,61

As 58, 60, 62, 64, e 66 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 13. As Figuras 59, 61, 63, 65 e 67 ilustram as curvas ROC referentes à mesma tabela.

Figura 58 – Matriz de confusão do Classificador 1 com função *kernel* linear e proporção 70% teste 30% treinamento.

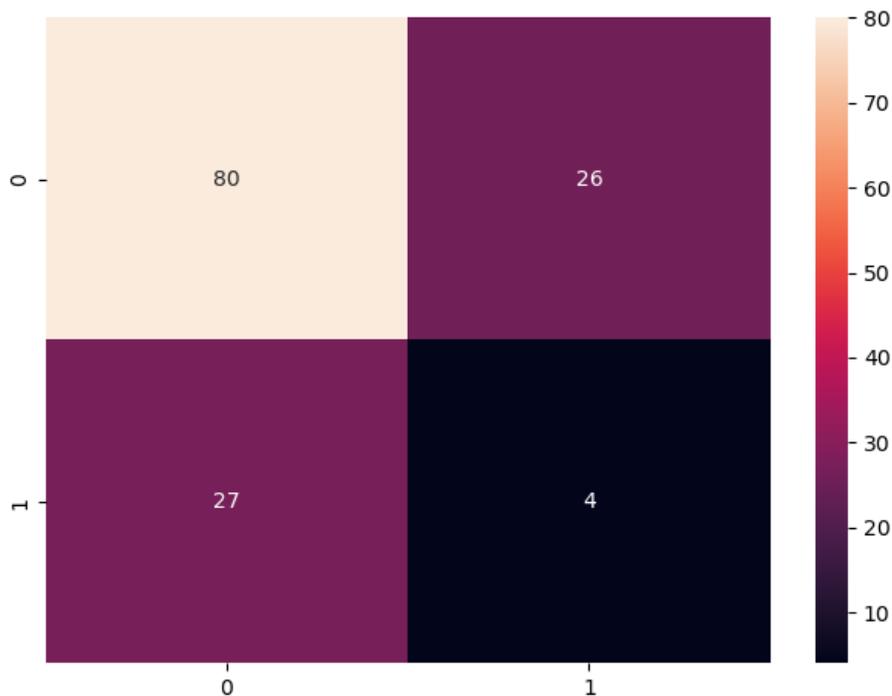


Figura 59 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 70% teste 30% treinamento.

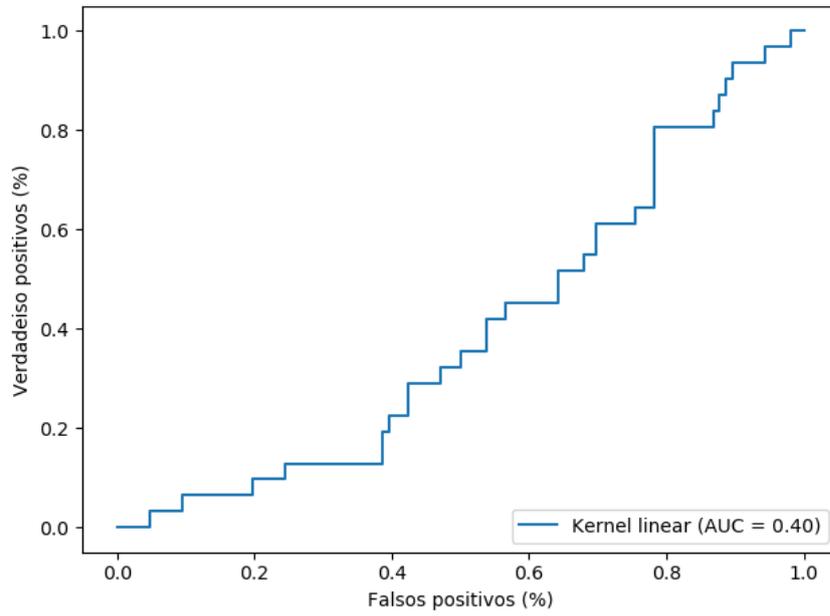


Figura 60 – Matriz de confusão do Classificador 2 com função *kernel* linear e proporção 70% teste 30% treinamento.

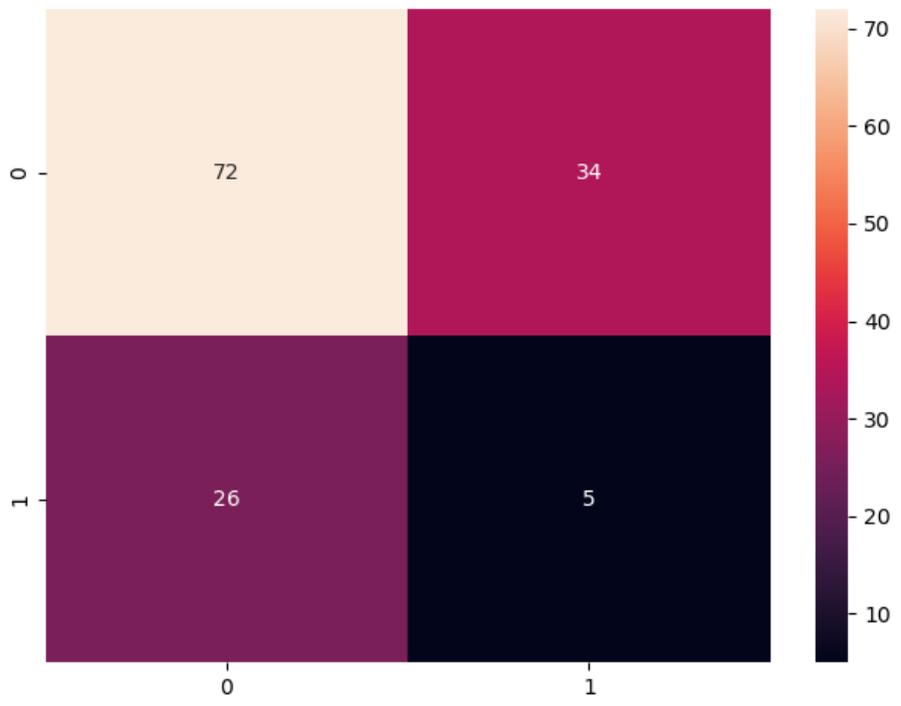


Figura 61 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 70% teste 30% treinamento.

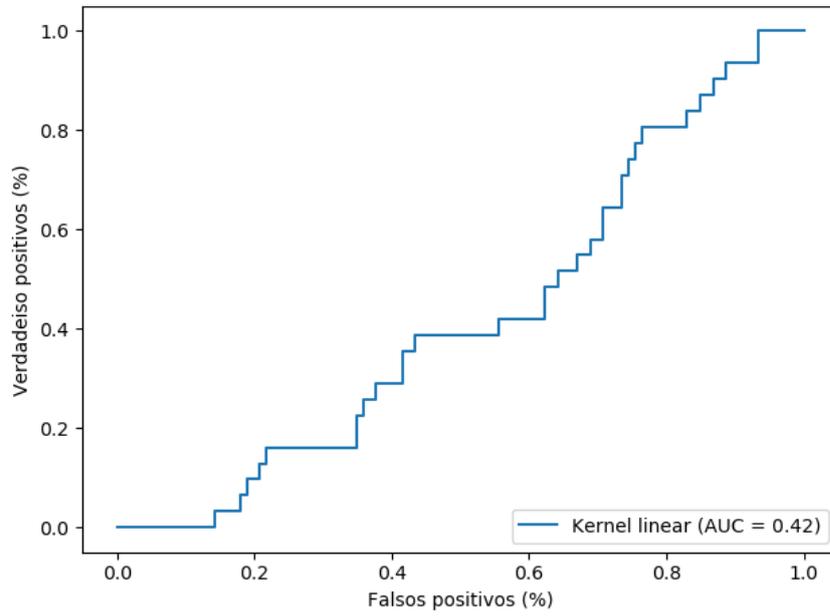


Figura 62 – Matriz de confusão do Classificador 3 com função *kernel* linear e proporção 70% teste 30% treinamento.

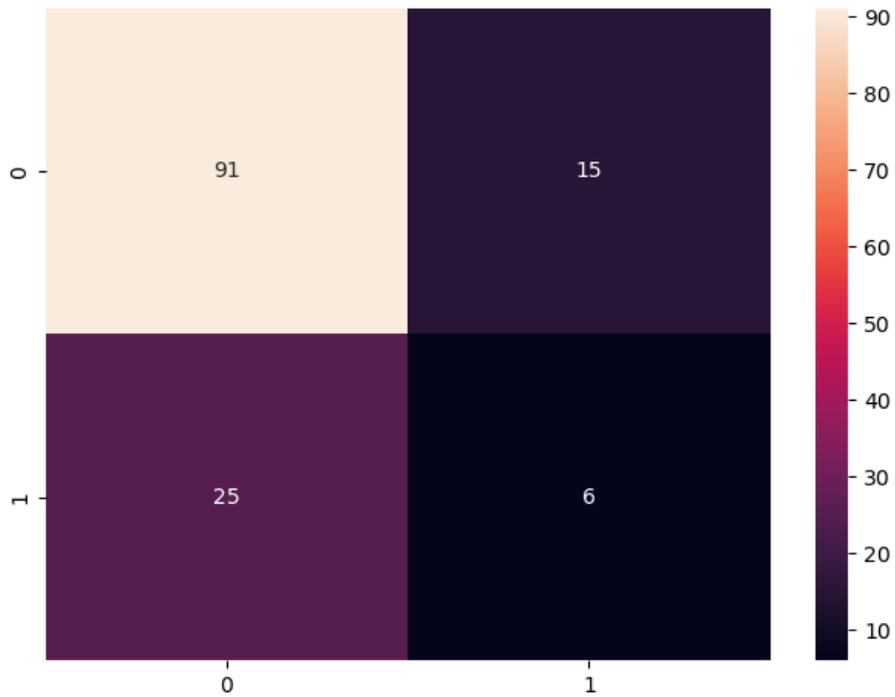


Figura 63 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 70% teste 30% treinamento.

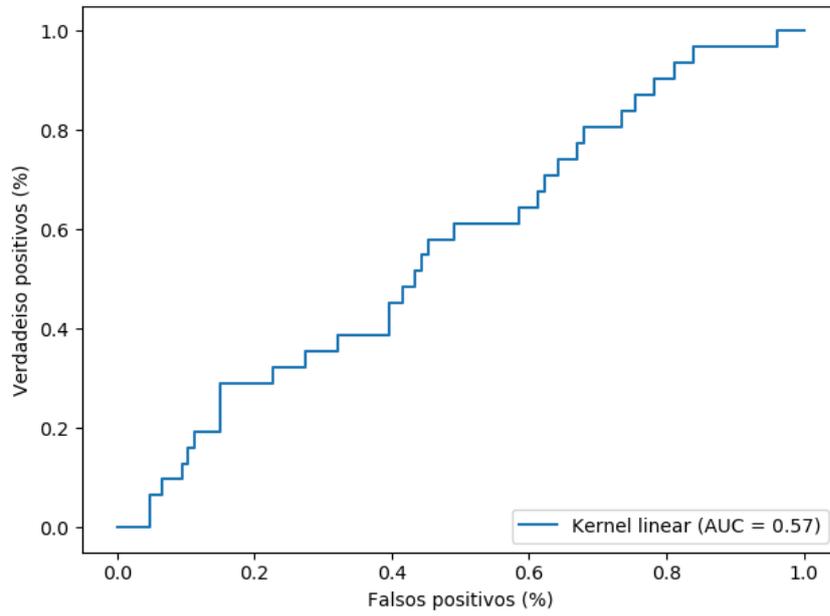


Figura 64 – Matriz de confusão do Classificador 4 com função *kernel* linear e proporção 70% teste 30% treinamento.

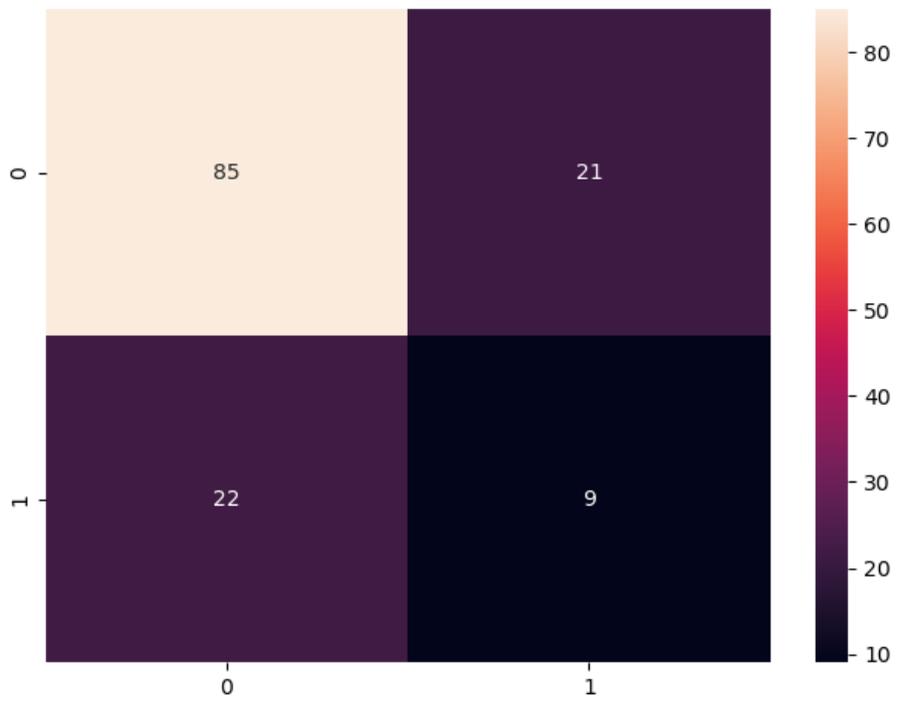


Figura 65 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 70% teste 30% treinamento.

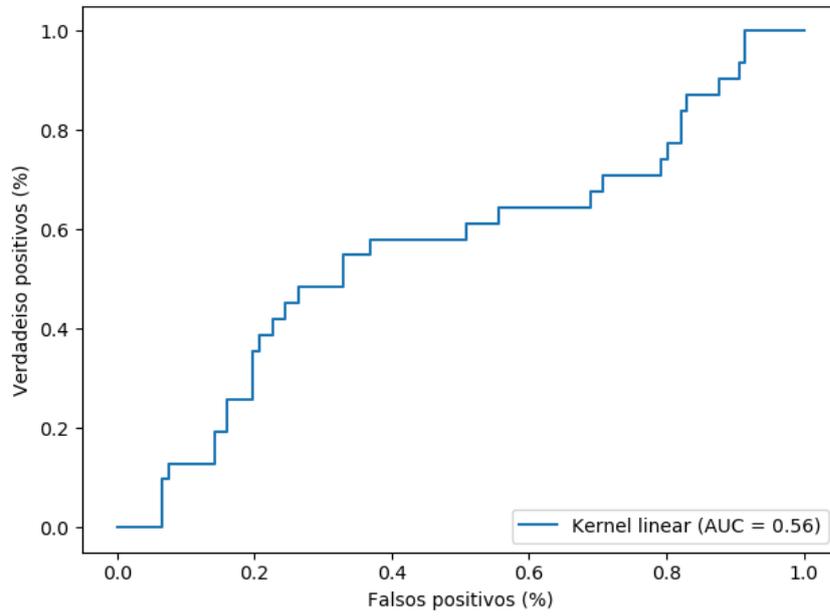


Figura 66 – Matriz de confusão do Classificador 5 com função *kernel* linear e proporção 70% teste 30% treinamento.

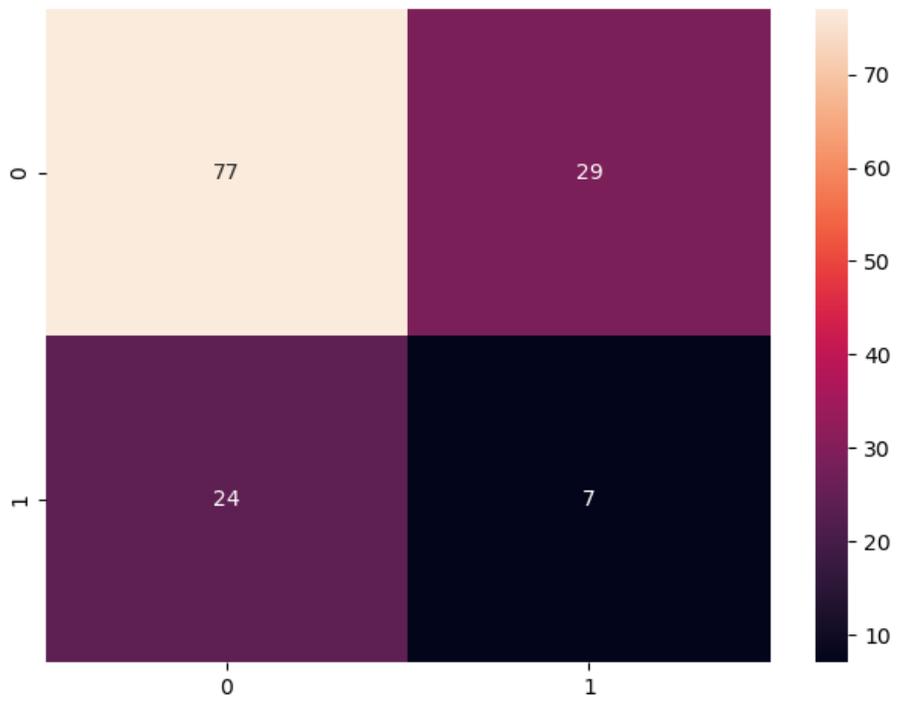
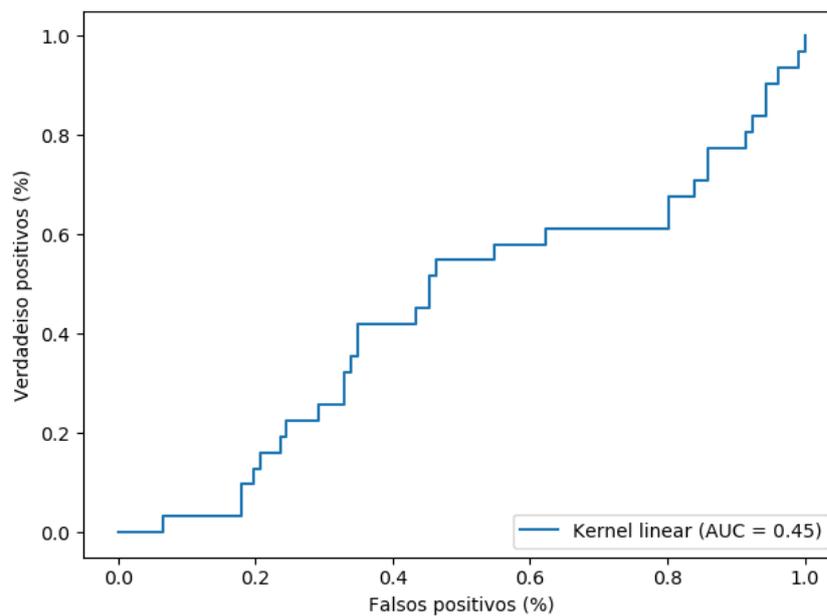


Figura 67 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 70% teste 30% treinamento.



A Tabela 14 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 70% para treinamento e 30% teste.

Tabela 14 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 70% para treinamento e 30% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,74	0,68	0,71	106	0,56
Classificador do Estágio 2	0,78	0,65	0,71	106	0,57
Classificador do Estágio 3	0,82	0,79	0,80	106	0,70
Classificador do Estágio 4	0,81	0,78	0,80	106	0,69
Classificador do Estágio 5	0,86	0,75	0,80	106	0,71

As 68, 70, 72, 74, e 76 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 14. As Figuras 69, 71, 73, 75 e 77 ilustram as curvas ROC referentes à mesma tabela.

Figura 68 – Matriz de confusão do Classificador 1 com função *kernel* linear e proporção 70% teste 30% treinamento.

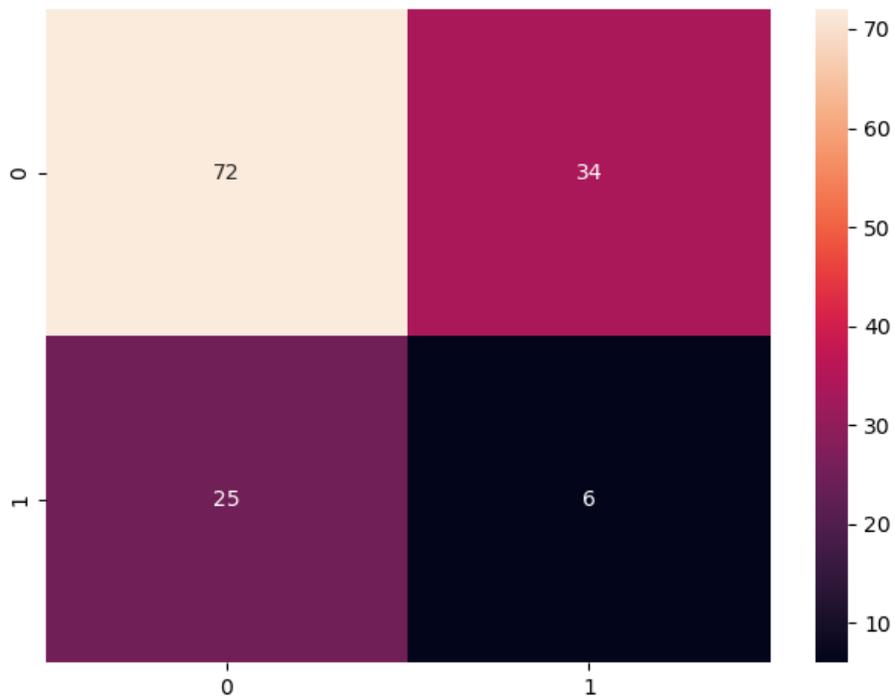


Figura 69 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 70% teste 30% treinamento.

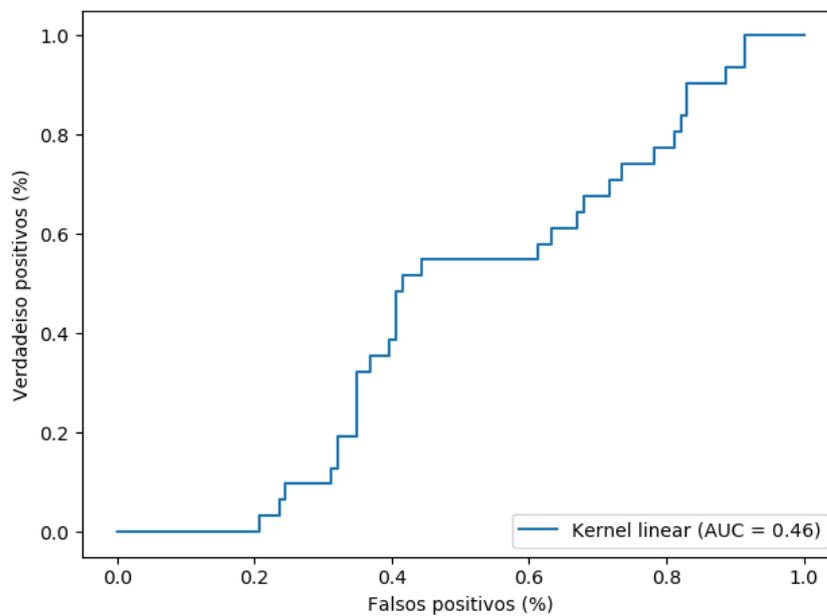


Figura 70 – Matriz de confusão do Classificador 2 com função *kernel* linear e proporção 70% teste 30% treinamento.

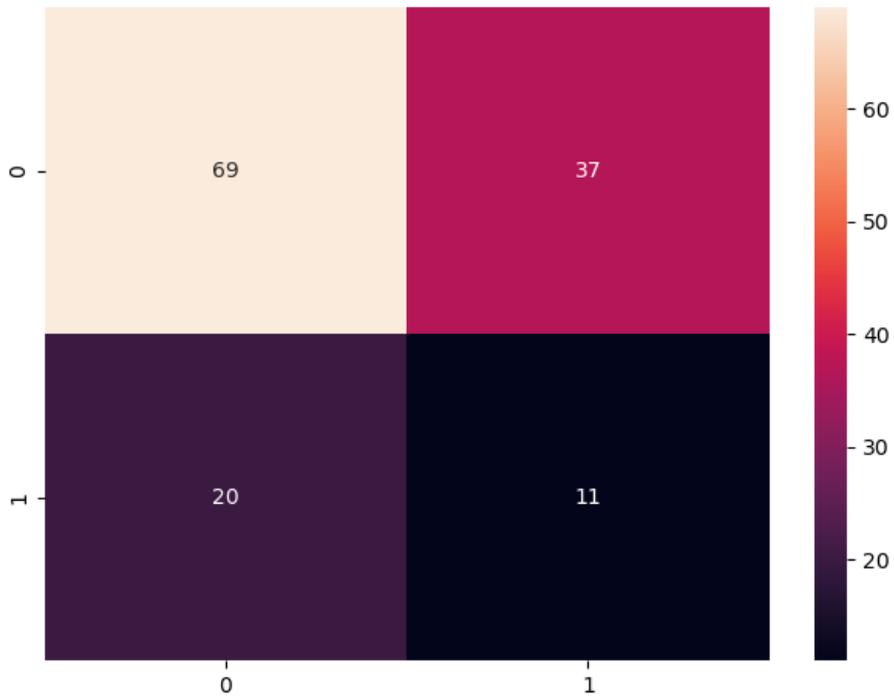


Figura 71 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 70% teste 30% treinamento.

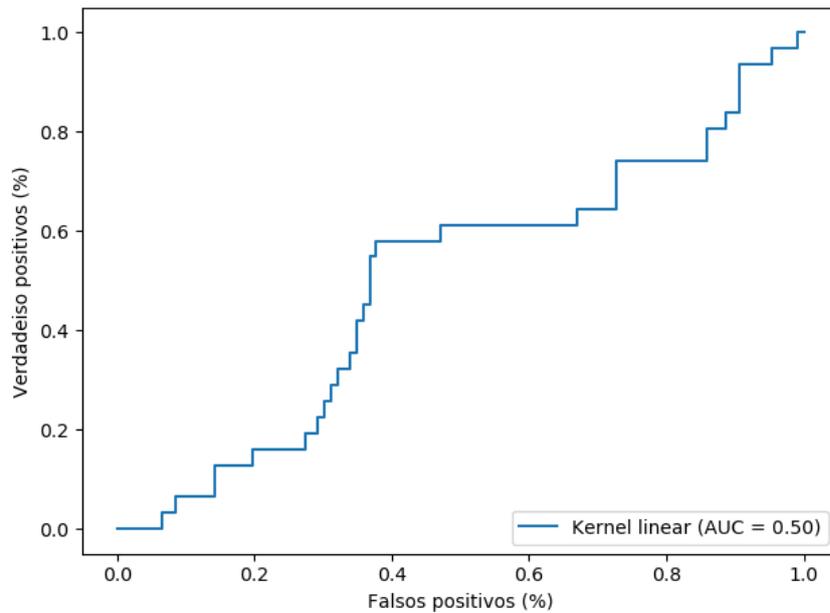


Figura 72 – Matriz de confusão do Classificador 3 com função *kernel* linear e proporção 70% teste 30% treinamento.

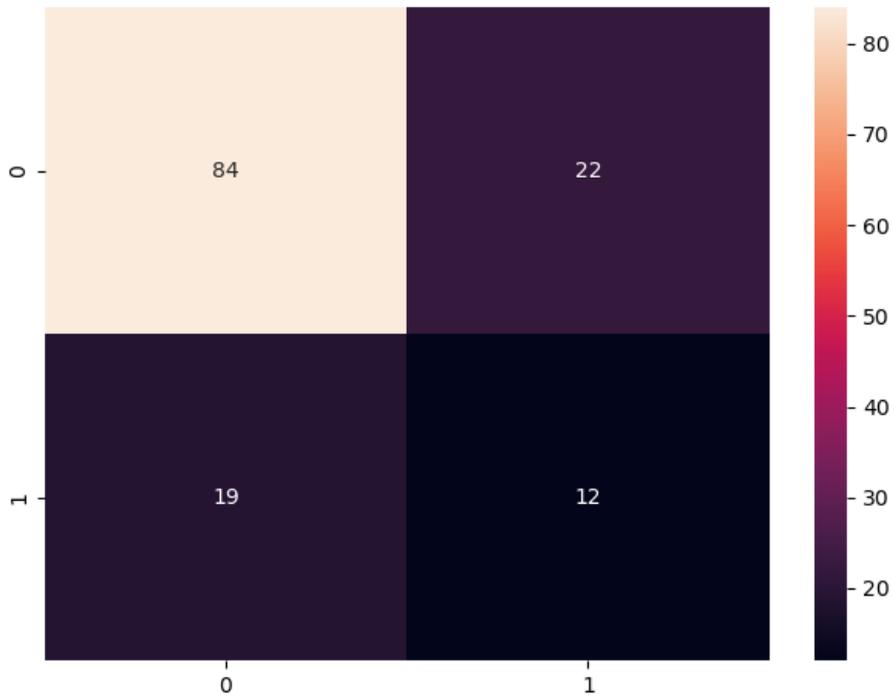


Figura 73 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 70% teste 30% treinamento.

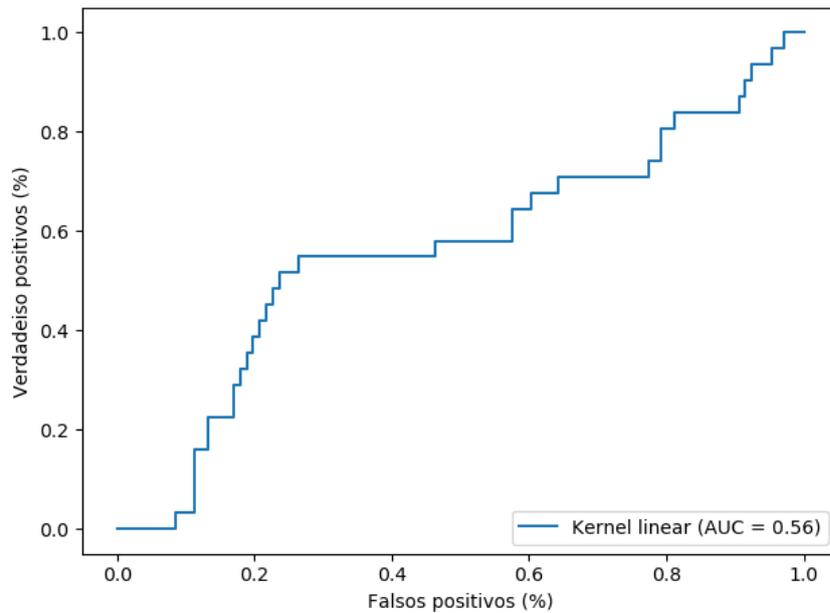


Figura 74 – Matriz de confusão do Classificador 4 com função *kernel* linear e proporção 70% teste 30% treinamento.

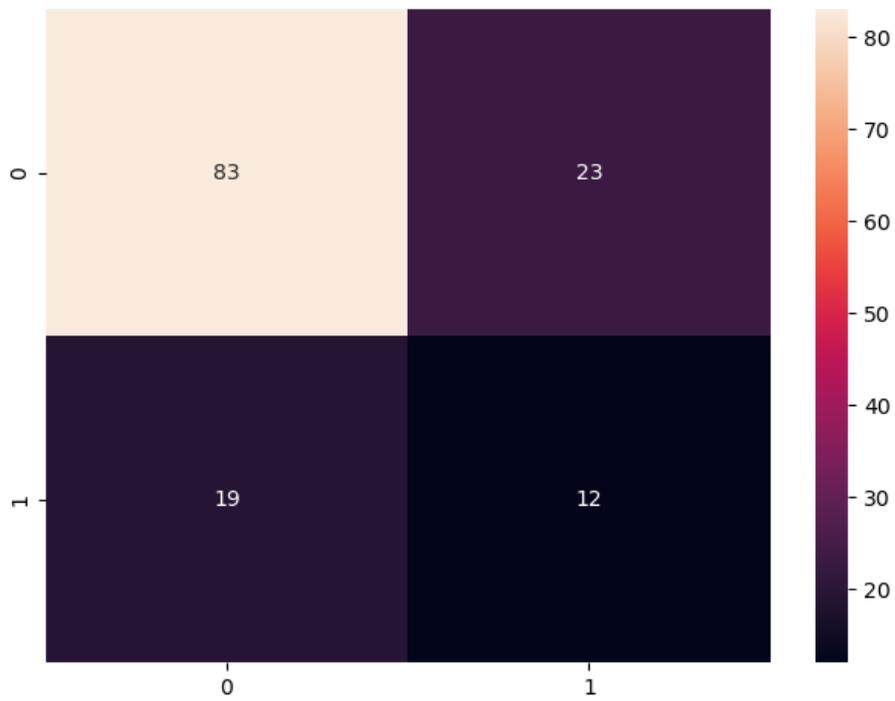


Figura 75 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 70% teste 30% treinamento.

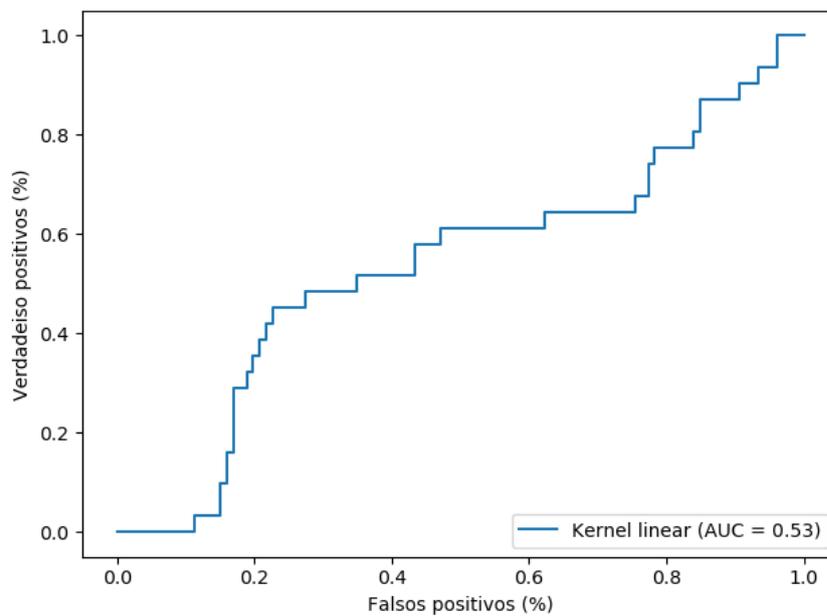


Figura 76 – Matriz de confusão do Classificador 5 com função *kernel* linear e proporção 70% teste 30% treinamento.

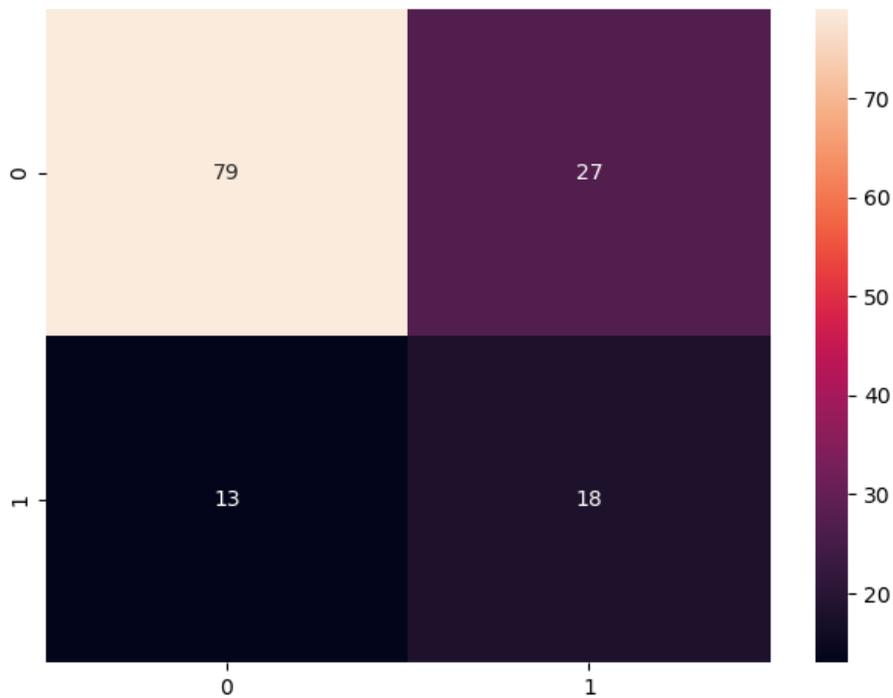
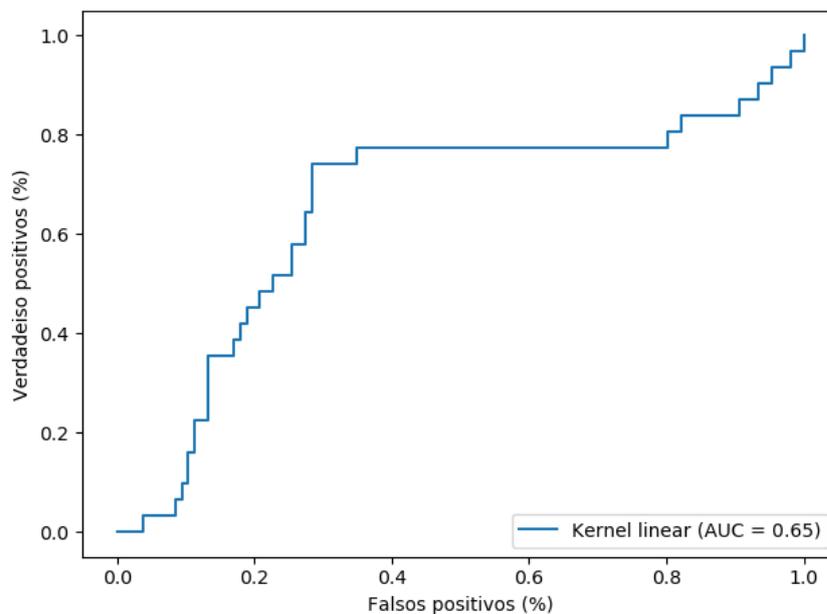


Figura 77 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 70% teste 30% treinamento.



A Tabela 15 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores com

função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 80% para treinamento e 20% teste.

Tabela 15 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 80% para treinamento e 20% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,75	0,78	0,76	72	0,62
Classificador do Estágio 2	0,76	0,74	0,75	72	0,61
Classificador do Estágio 3	0,85	0,85	0,85	72	0,76
Classificador do Estágio 4	0,80	0,83	0,82	72	0,71
Classificador do Estágio 5	0,75	0,72	0,74	72	0,60

As 78, 80, 82, 84, e 86 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 15. As Figuras 79, 81, 83, 85 e 87 ilustram as curvas ROC referentes à mesma tabela.

Figura 78 – Matriz de confusão do Classificador 1 com função *kernel* linear e proporção 80% teste 20% treinamento.

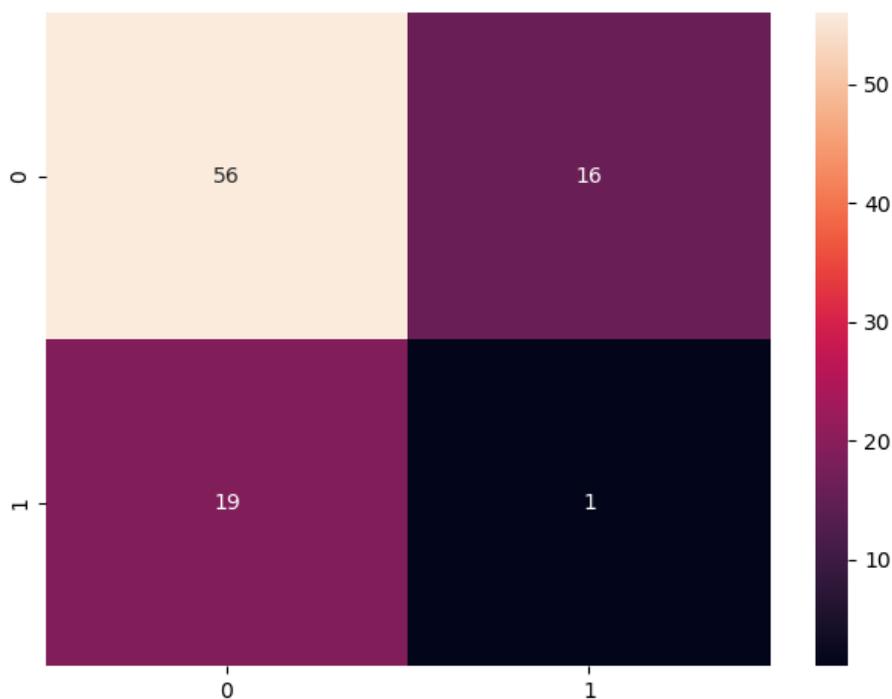


Figura 79 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 70% teste 30% treinamento.

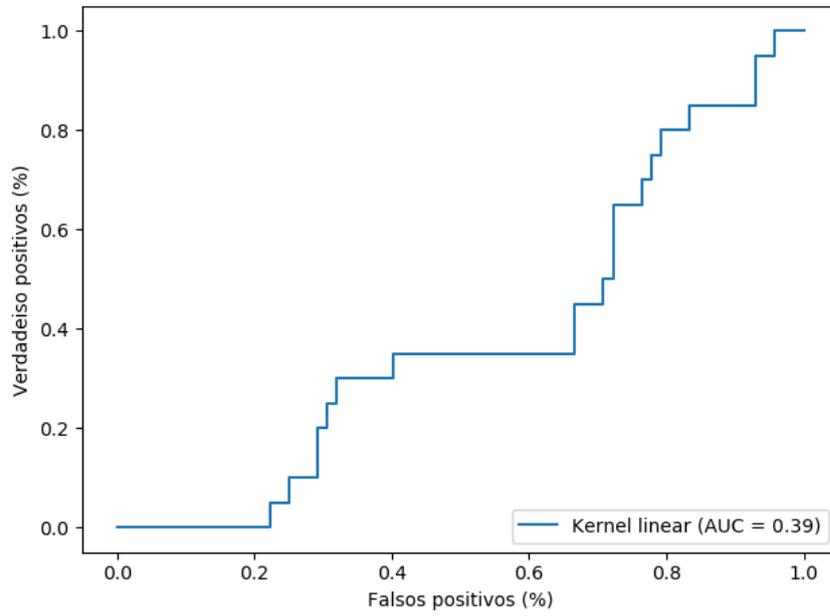


Figura 80 – Matriz de confusão do Classificador 2 com função *kernel* linear e proporção 80% teste 20% treinamento.

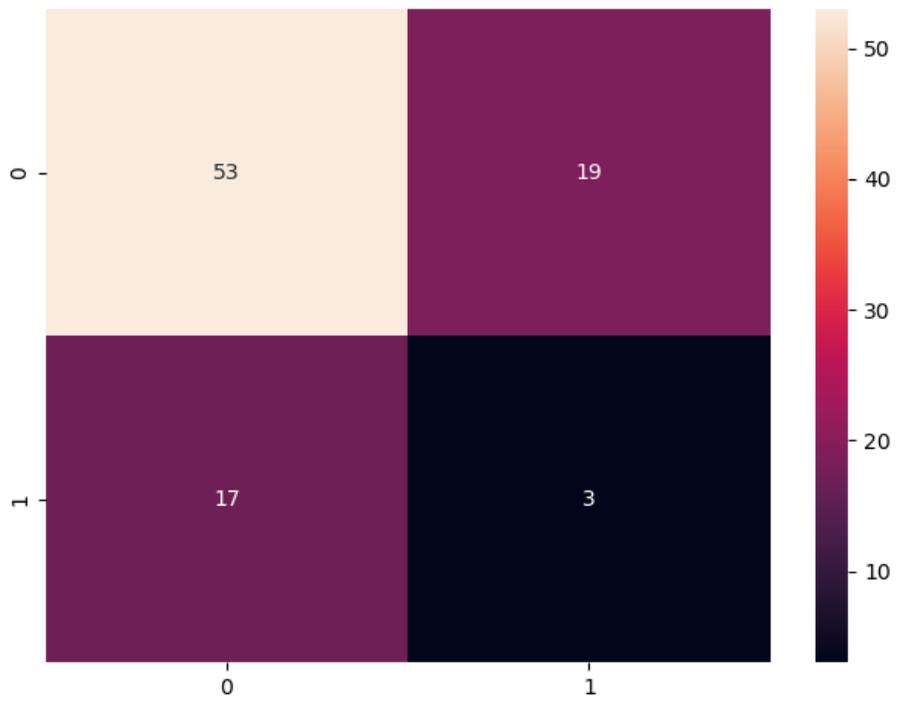


Figura 81 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 80% teste 20% treinamento.

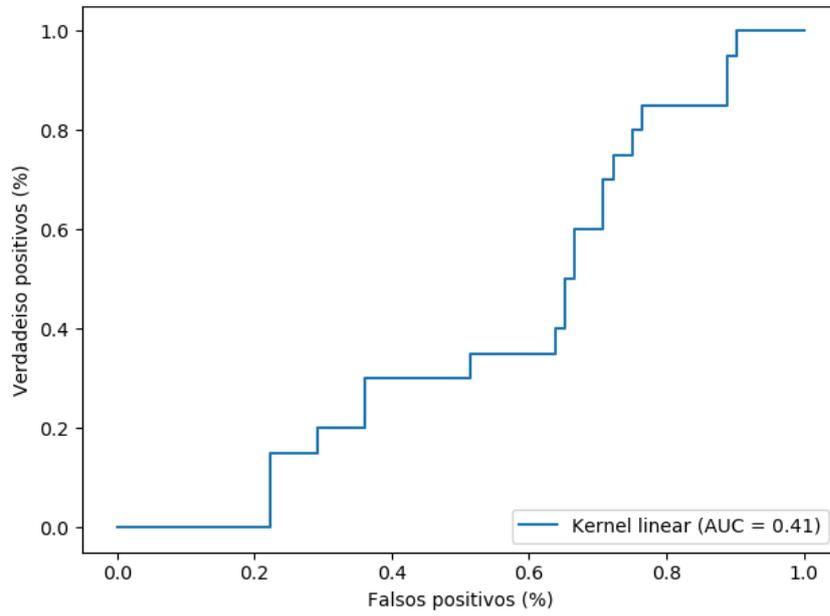


Figura 82 – Matriz de confusão do Classificador 3 com função *kernel* linear e proporção 80% teste 20% treinamento.

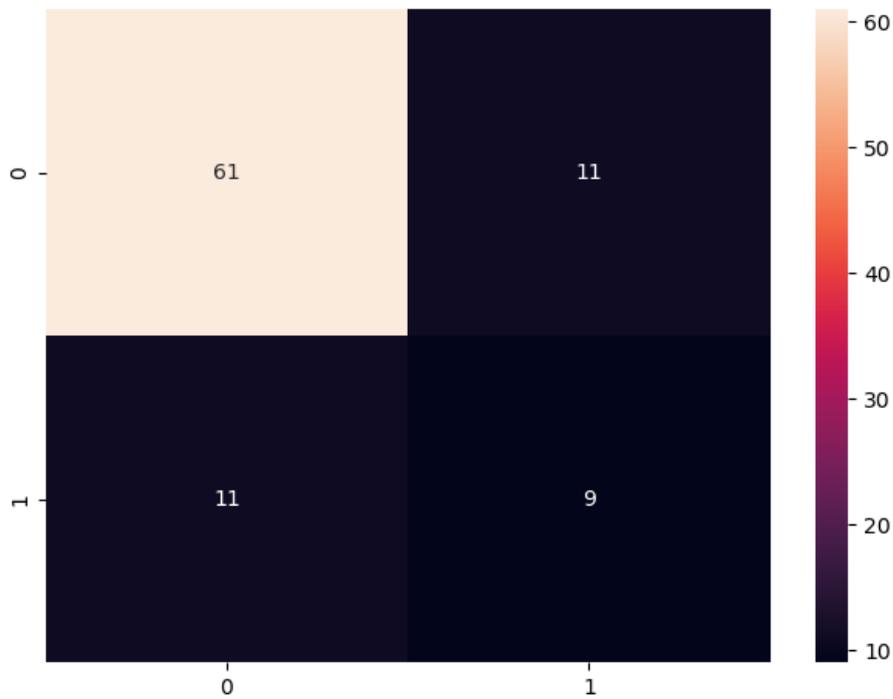


Figura 83 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 80% teste 20% treinamento.

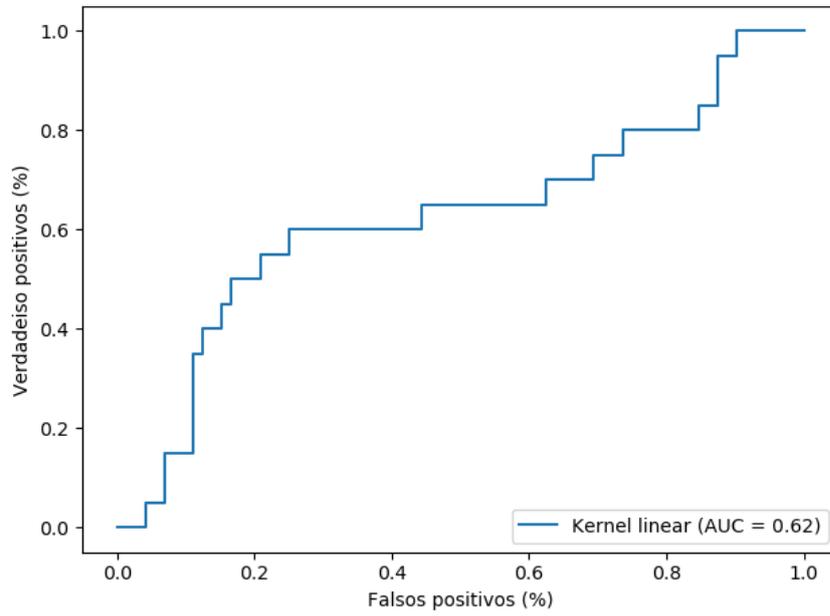


Figura 84 – Matriz de confusão do Classificador 4 com função *kernel* linear e proporção 80% teste 20% treinamento.

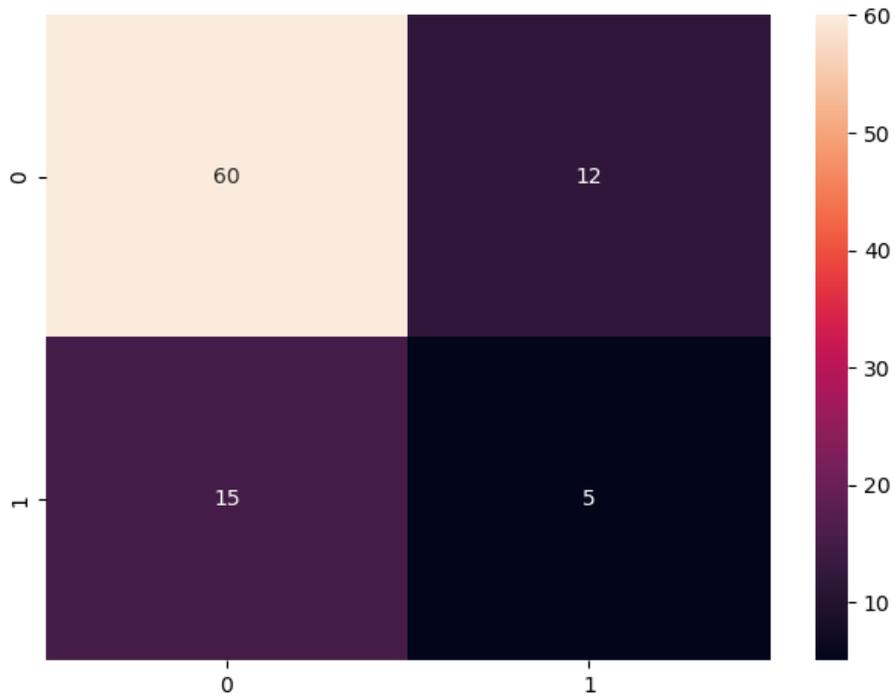


Figura 85 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 80% teste 20% treinamento.

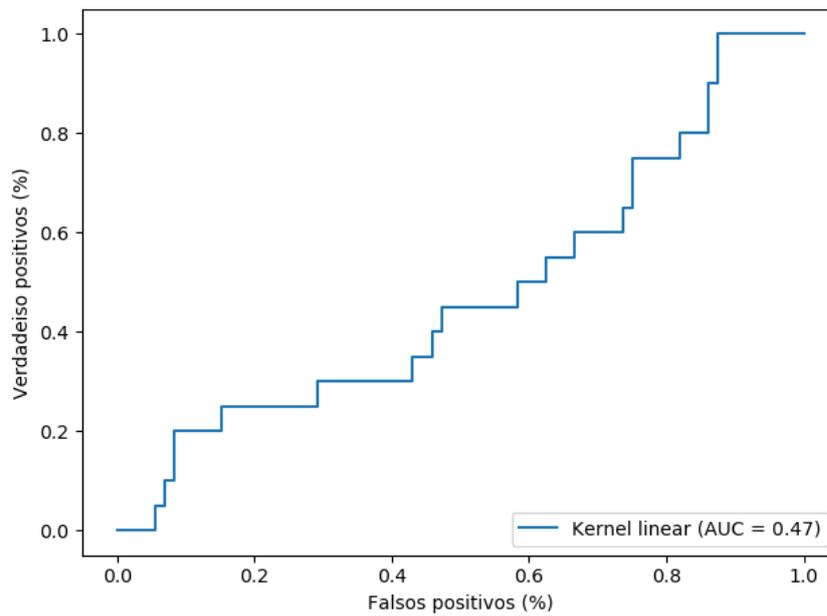


Figura 86 – Matriz de confusão do Classificador 5 com função *kernel* linear e proporção 80% teste 20% treinamento.

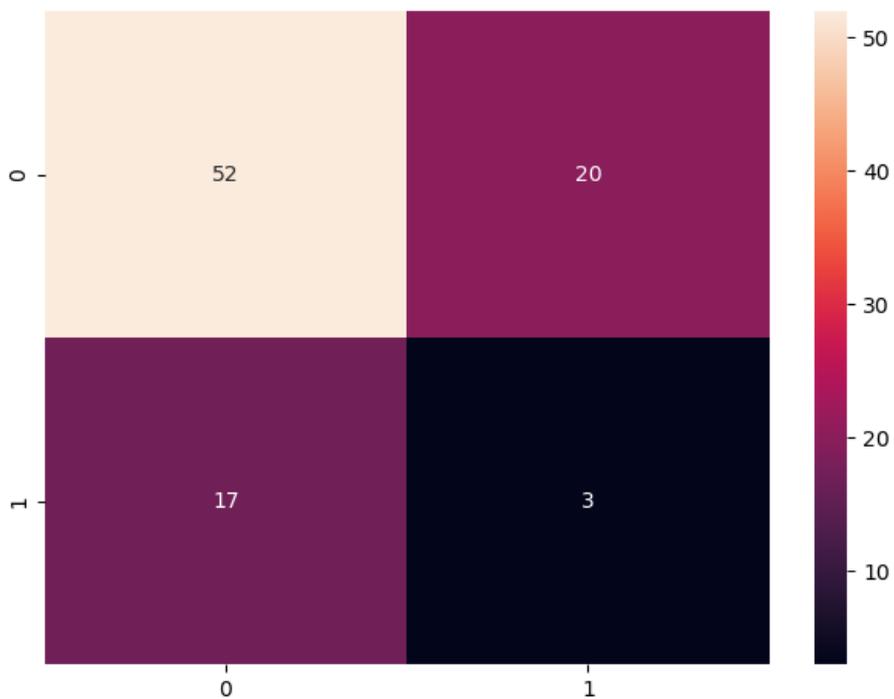
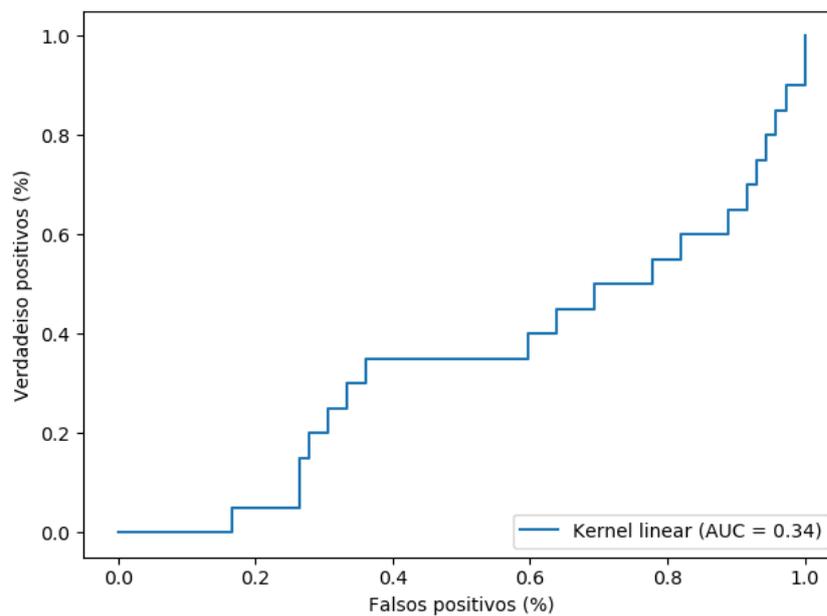


Figura 87 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 80% teste 20% treinamento.



A Tabela 16 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* linear de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 80% para treinamento e 20% teste.

Tabela 16 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função *kernel* linear utilizando a proporção de 80% para treinamento e 20% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,71	0,61	0,66	72	0,50
Classificador do Estágio 2	0,72	0,65	0,69	72	0,53
Classificador do Estágio 3	0,79	0,81	0,80	72	0,68
Classificador do Estágio 4	0,81	0,78	0,79	72	0,68
Classificador do Estágio 5	0,82	0,75	0,78	72	0,67

As 88, 90, 92, 94, e 96 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 16. As Figuras 99, 91, 93, 95 e 97 ilustram as curvas ROC referentes à mesma tabela.

Figura 88 – Matriz de confusão do Classificador 1 proporção 80% teste 20% treinamento.

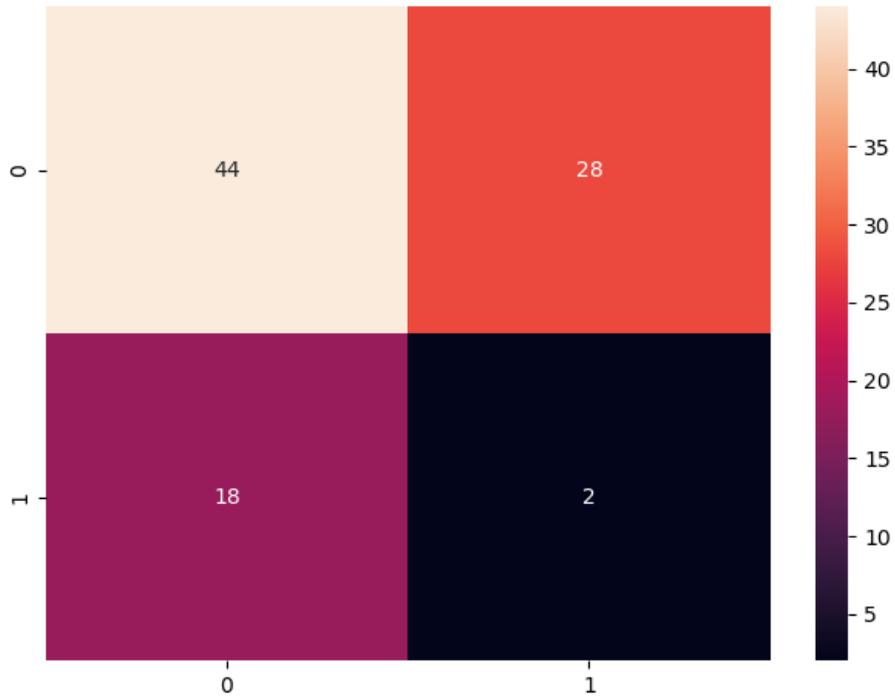


Figura 89 – Curva ROC do Classificador 1 com função *kernel* linear e proporção 80% teste 20% treinamento.

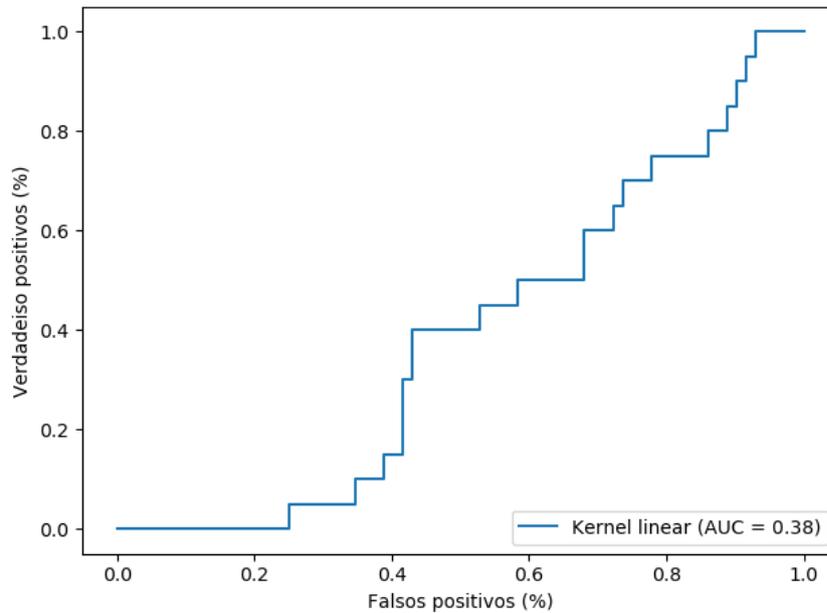


Figura 90 – Matriz de confusão do Classificador 2 proporção 80% teste 20% treinamento.

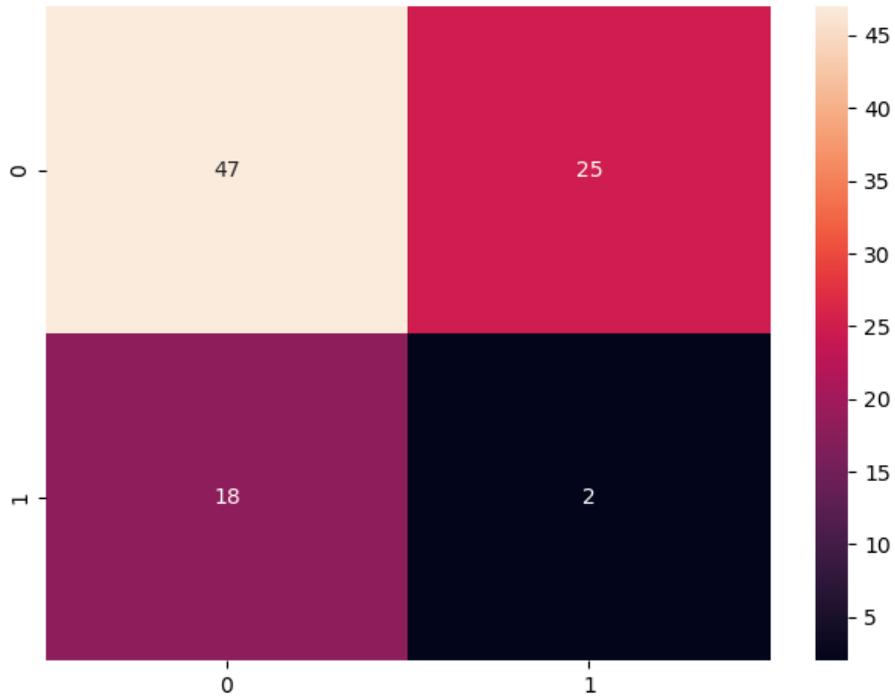


Figura 91 – Curva ROC do Classificador 2 com função *kernel* linear e proporção 80% teste 20% treinamento.

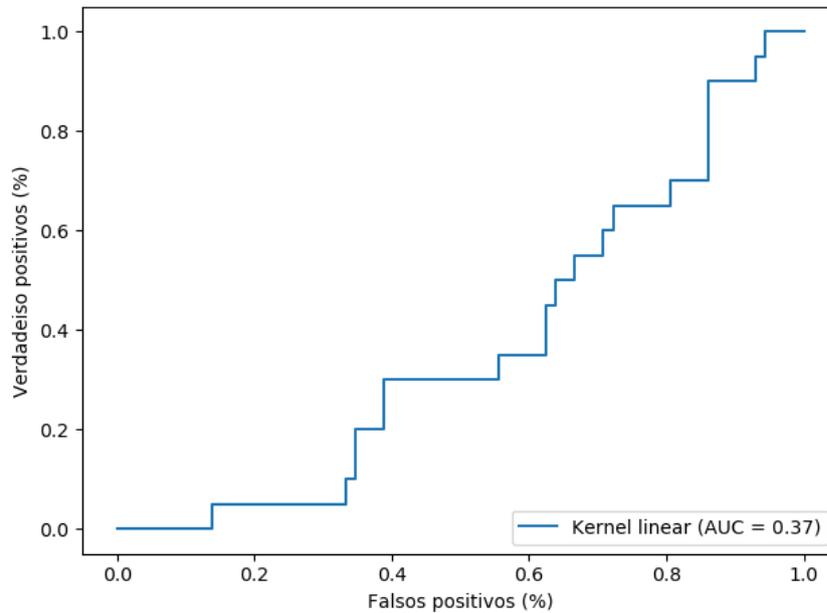


Figura 92 – Matriz de confusão do Classificador 3 proporção 80% teste 20% treinamento.

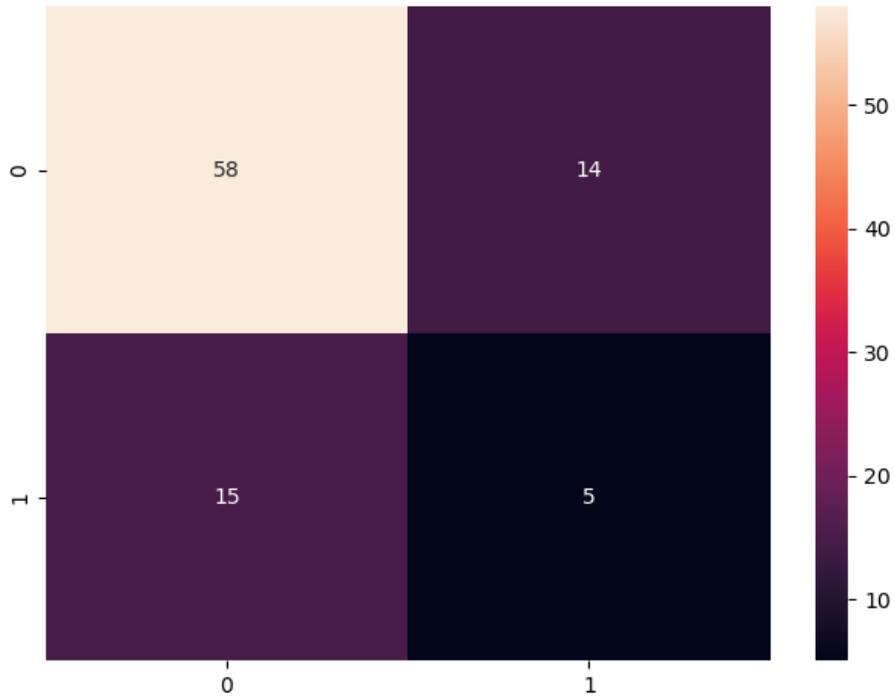


Figura 93 – Curva ROC do Classificador 3 com função *kernel* linear e proporção 80% teste 20% treinamento.

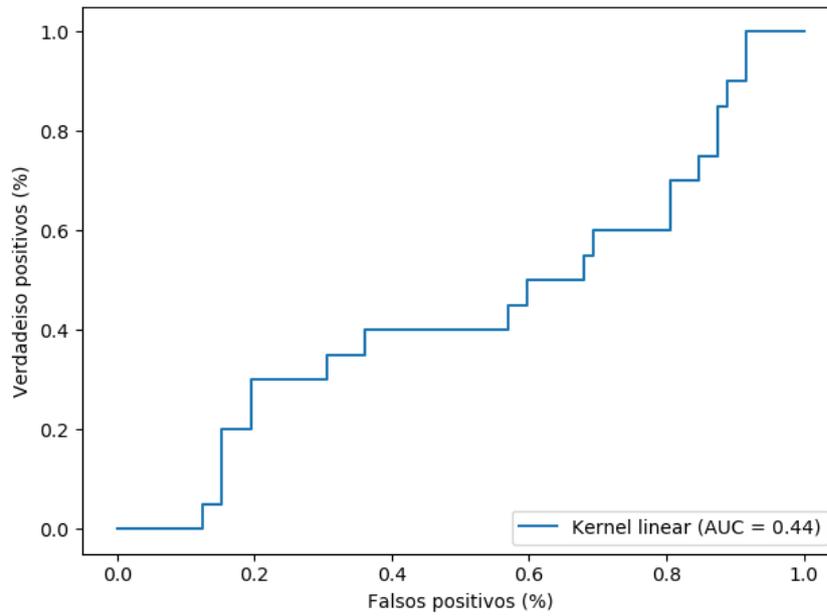


Figura 94 – Matriz de confusão do Classificador 4 proporção 80% teste 20% treinamento.

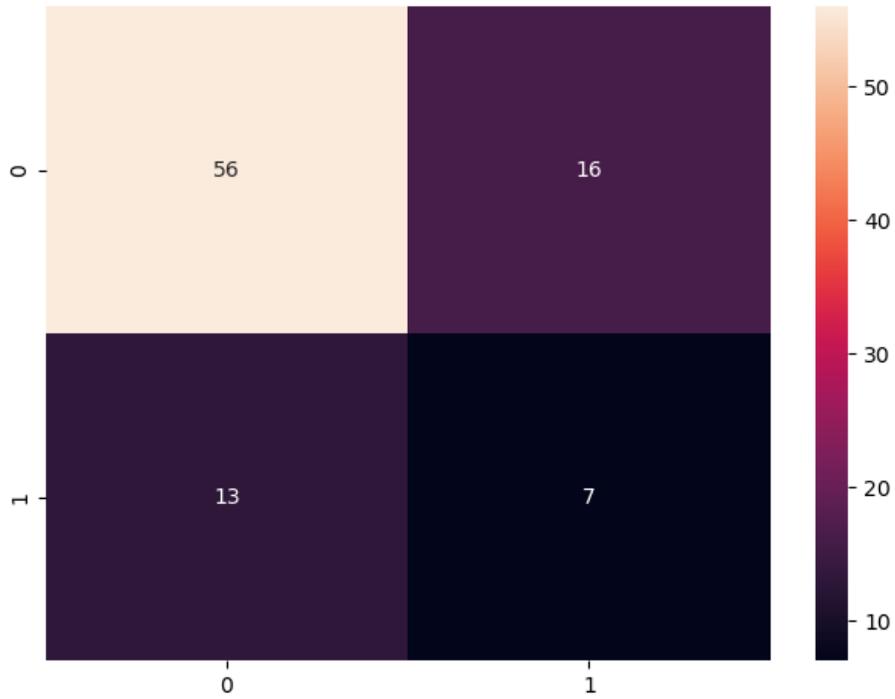


Figura 95 – Curva ROC do Classificador 4 com função *kernel* linear e proporção 80% teste 20% treinamento.

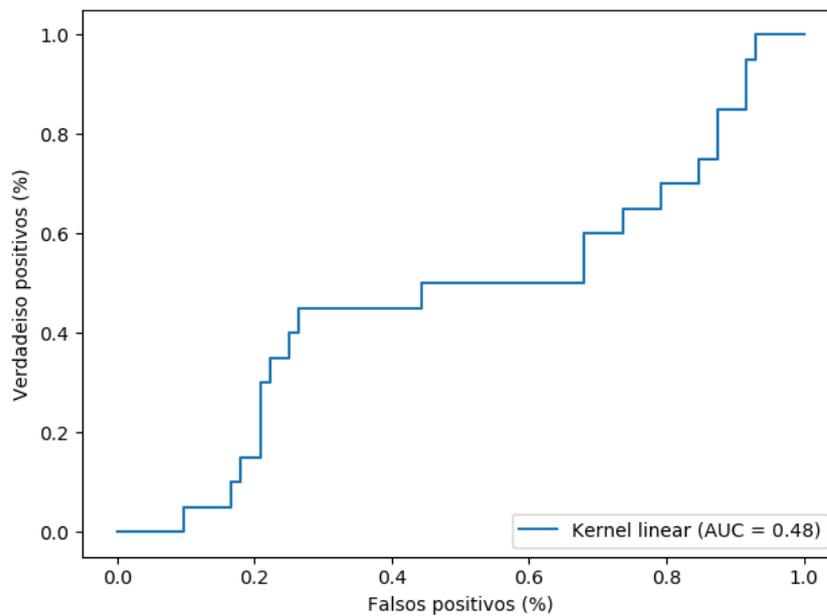


Figura 96 – Matriz de confusão do Classificador 5 proporção 80% teste 20% treinamento.

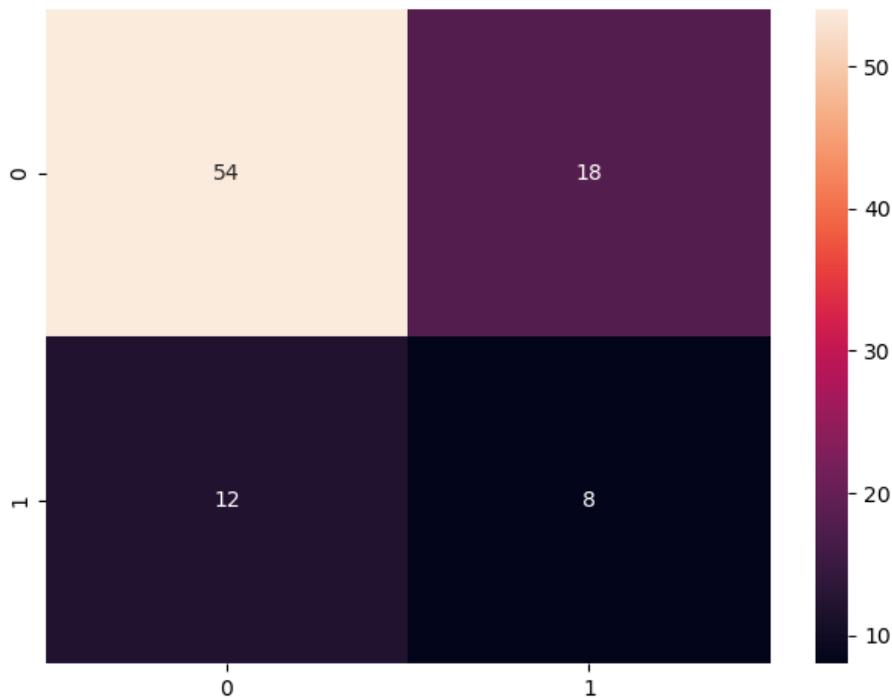
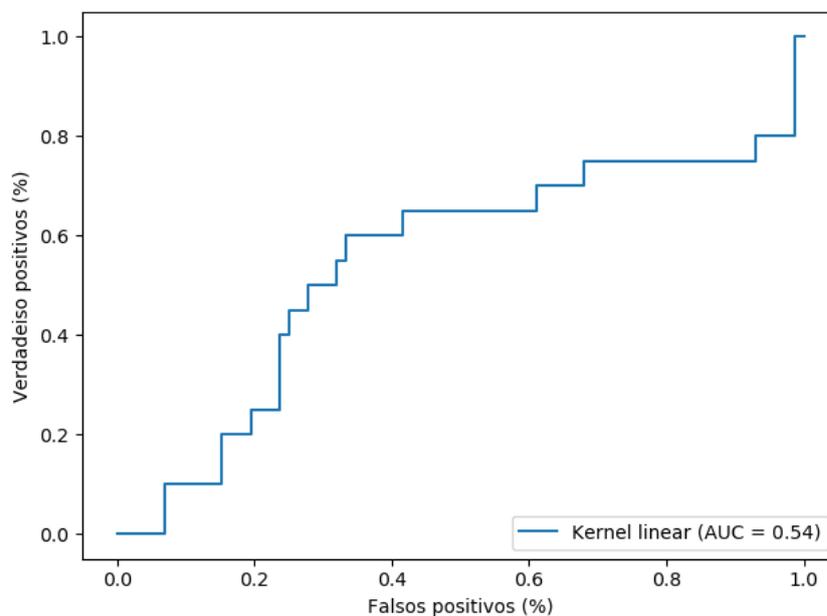


Figura 97 – Curva ROC do Classificador 5 com função *kernel* linear e proporção 80% teste 20% treinamento.



A Tabela 17 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 50% para treinamento e teste.

Tabela 17 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 50% para treinamento e 50% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,83	0,91	0,87	72	0,78
Classificador do Estágio 2	0,80	0,91	0,85	72	0,75
Classificador do Estágio 3	0,81	1,0	0,89	72	0,81
Classificador do Estágio 4	0,84	0,95	0,89	72	0,81
Classificador do Estágio 5	0,81	0,97	0,88	72	0,79

As 98, 100, 102, 104, e 106 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 17. As Figuras 99, 91, 93, 95 e 97 ilustram as curvas ROC referentes à mesma tabela.

Figura 98 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

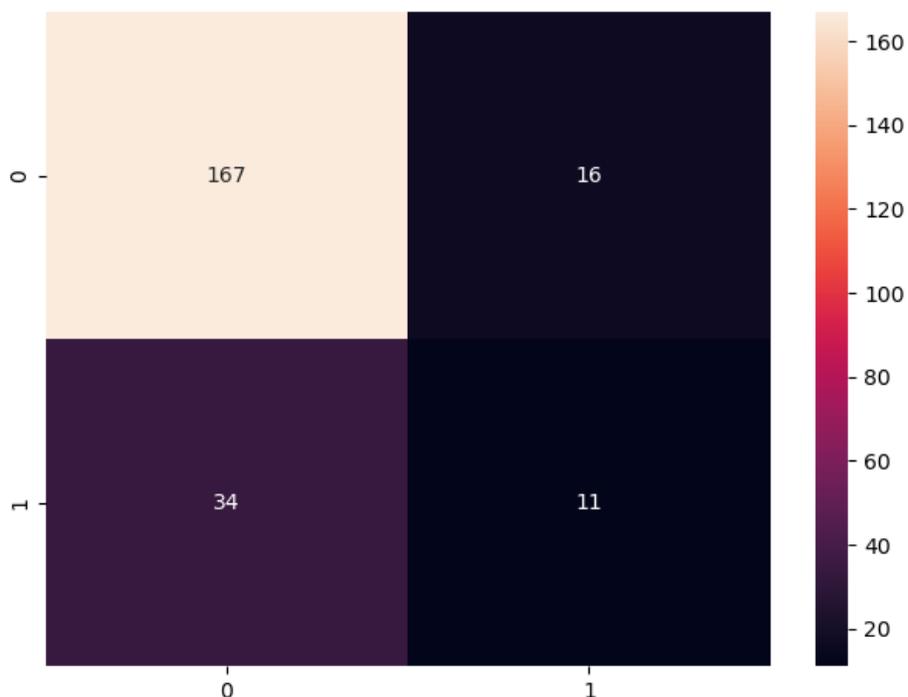


Figura 99 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

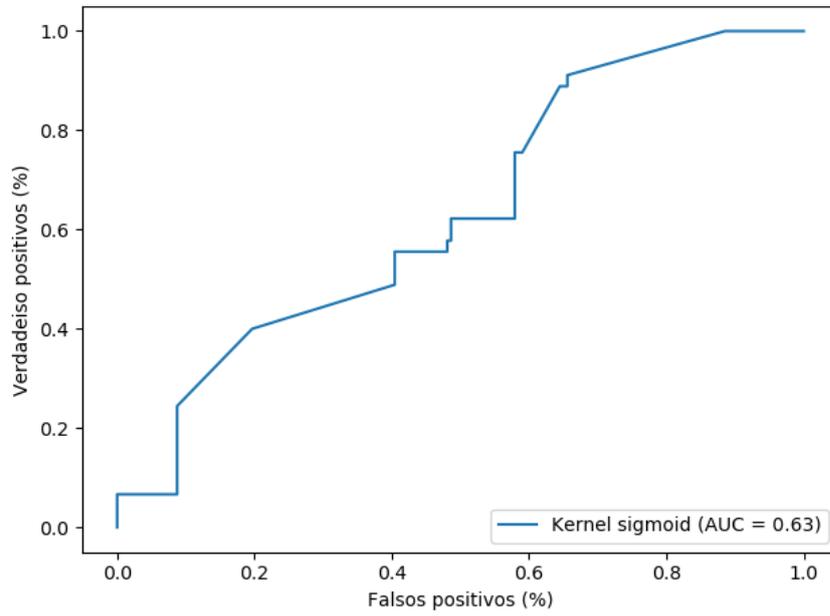


Figura 100 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

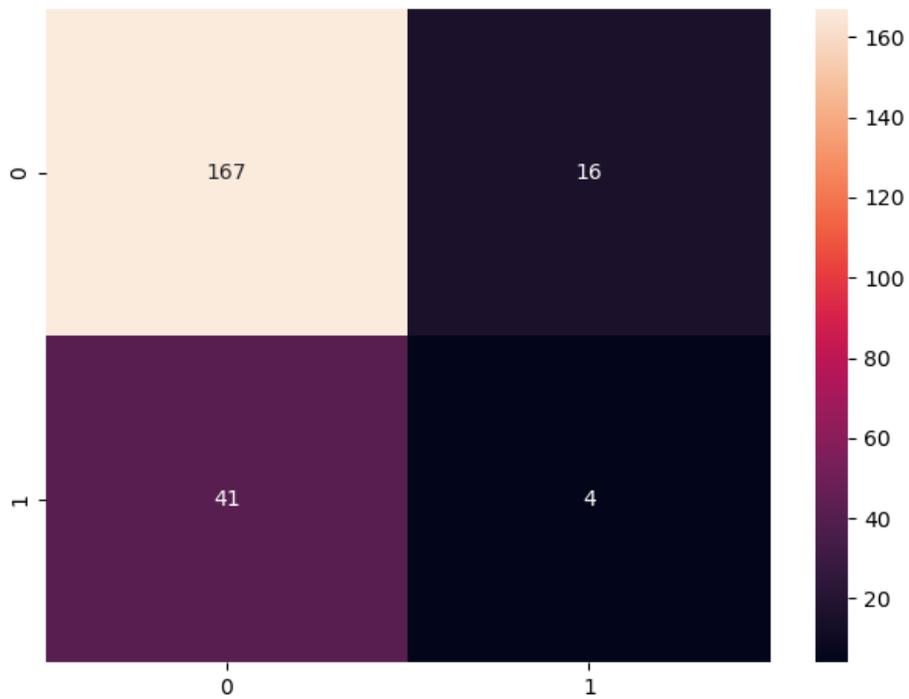


Figura 101 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

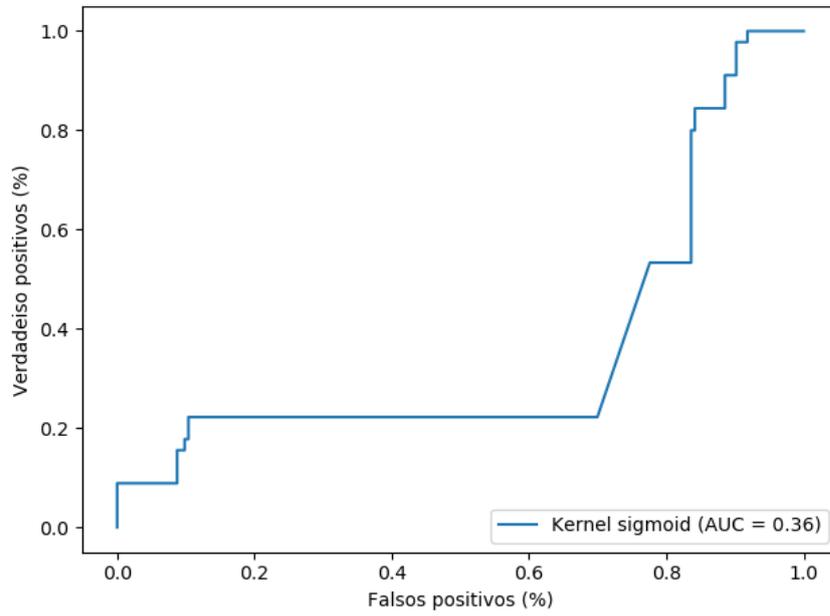


Figura 102 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

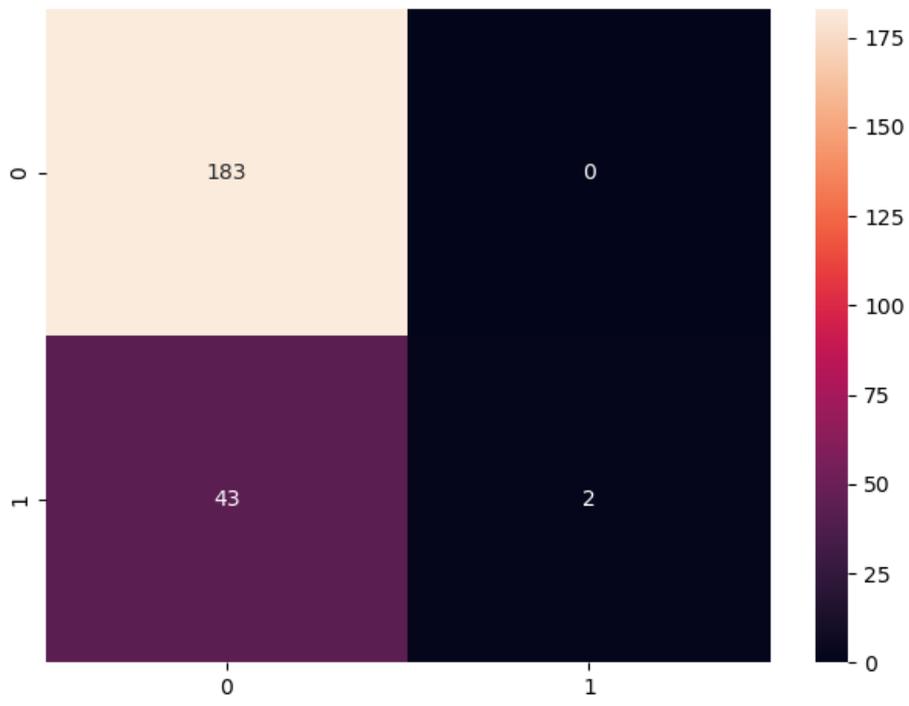


Figura 103 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

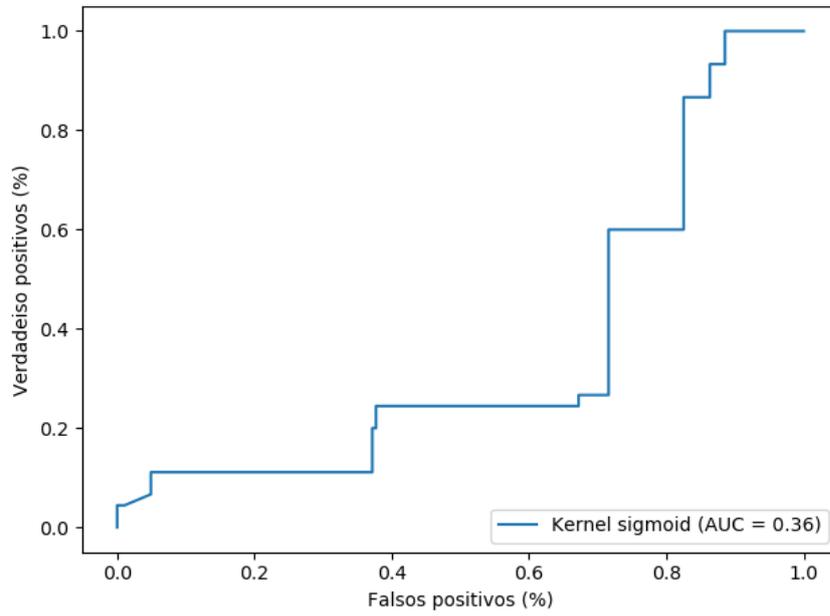


Figura 104 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

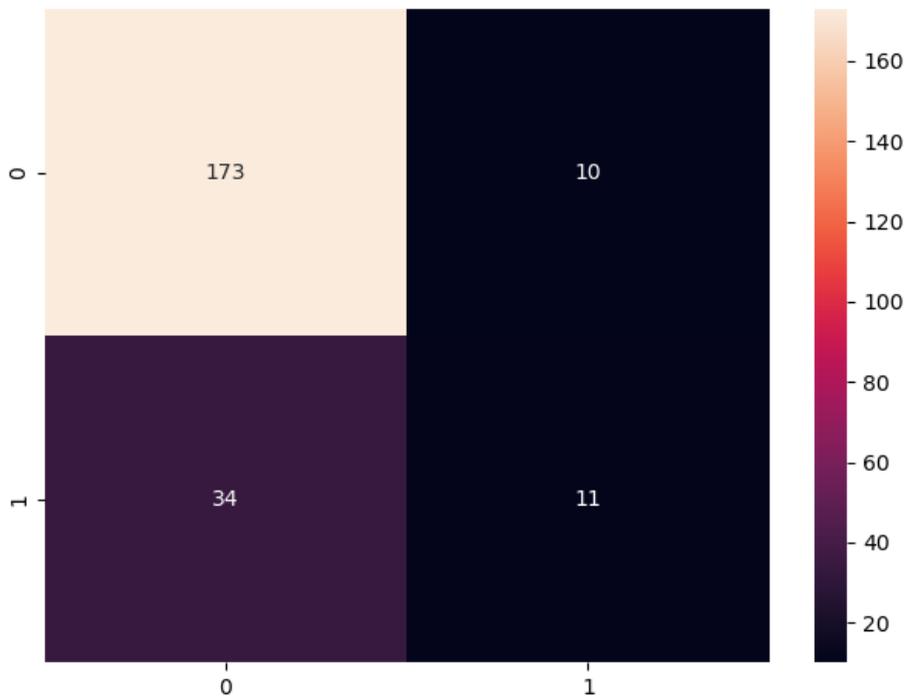


Figura 105 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

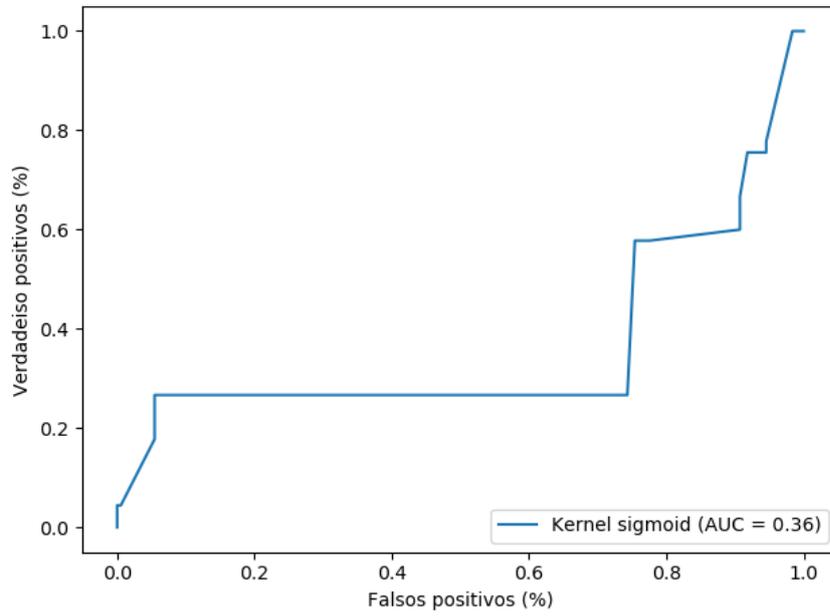


Figura 106 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

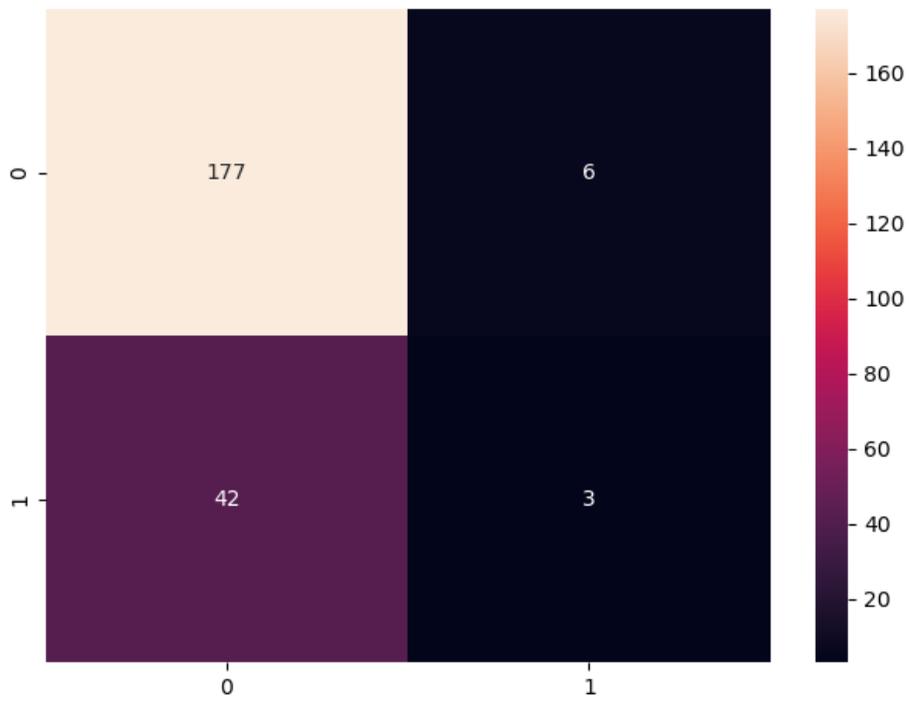
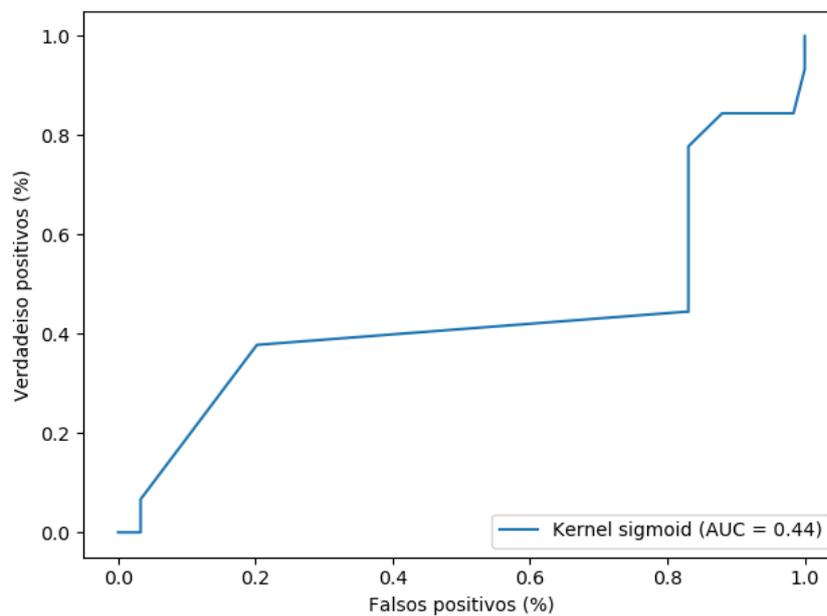


Figura 107 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.



A Tabela 18 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 50% para treinamento e teste.

Tabela 18 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 50% para treinamento e 50% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,80	0,94	0,86	183	0,76
Classificador do Estágio 2	0,80	0,97	0,88	183	0,78
Classificador do Estágio 3	0,80	1,0	0,89	183	0,80
Classificador do Estágio 4	0,81	1,0	0,89	183	0,81
Classificador do Estágio 5	0,80	0,93	0,86	183	0,75

As 108, 110, 112, 114, e 116 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 18. As Figuras 109, 111, 113, 115 e 117 ilustram as curvas ROC referentes à

mesma tabela.

Figura 108 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

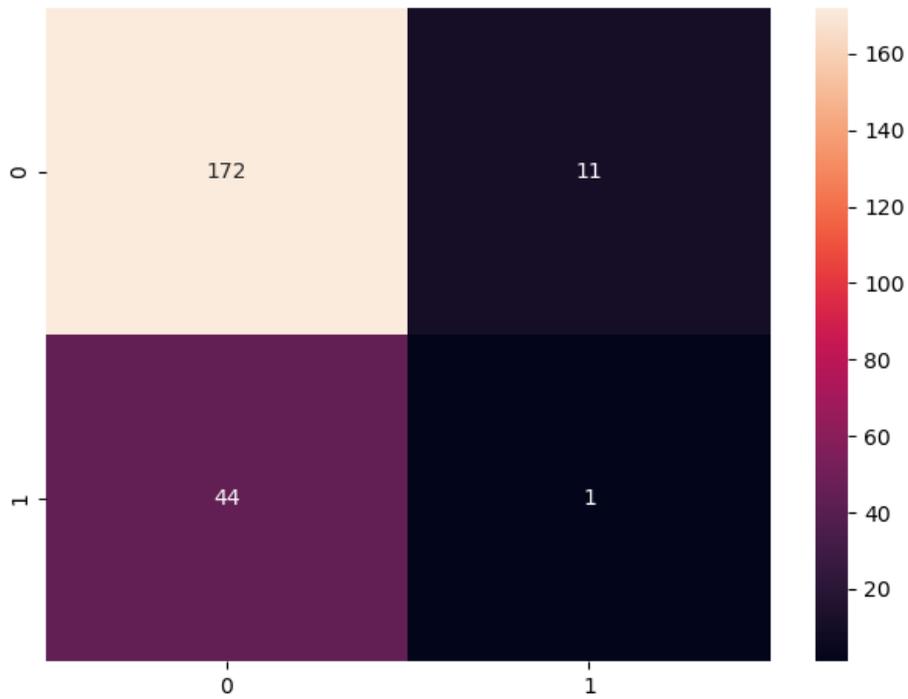


Figura 109 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

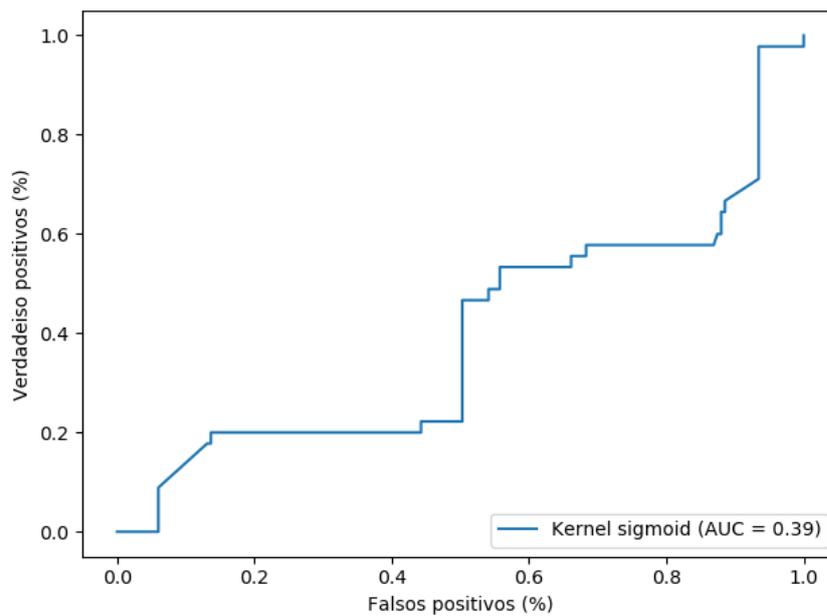


Figura 110 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

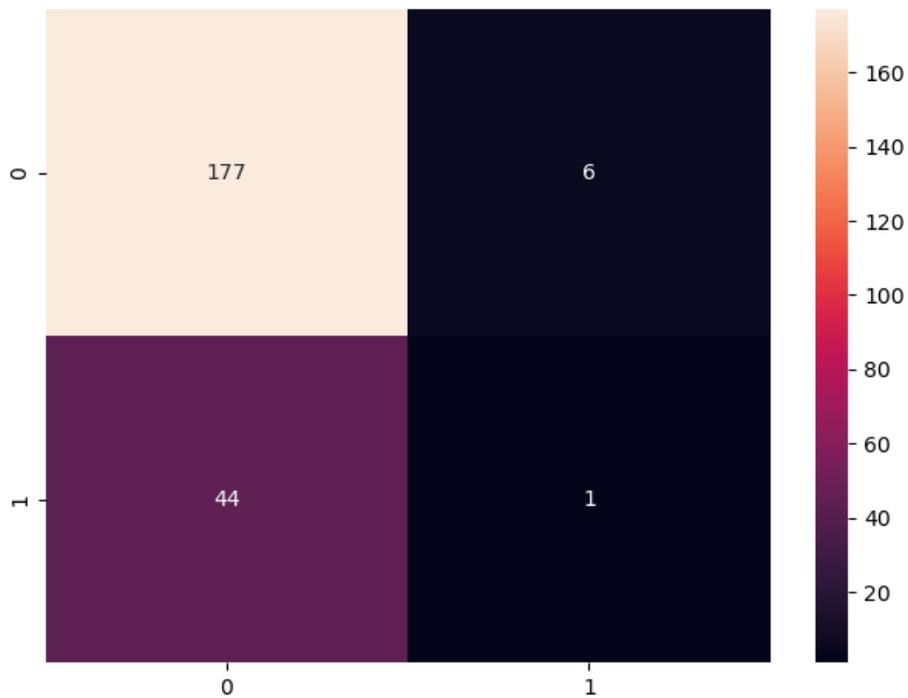


Figura 111 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

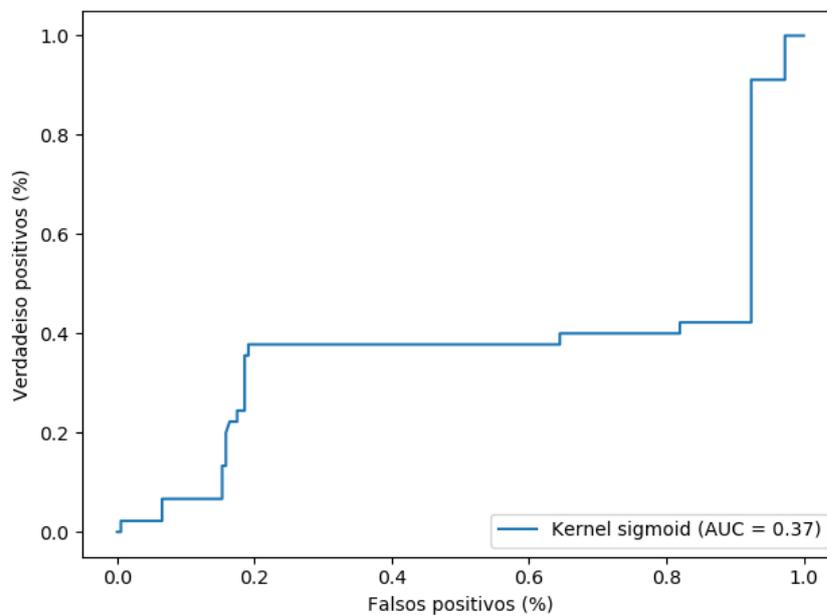


Figura 112 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

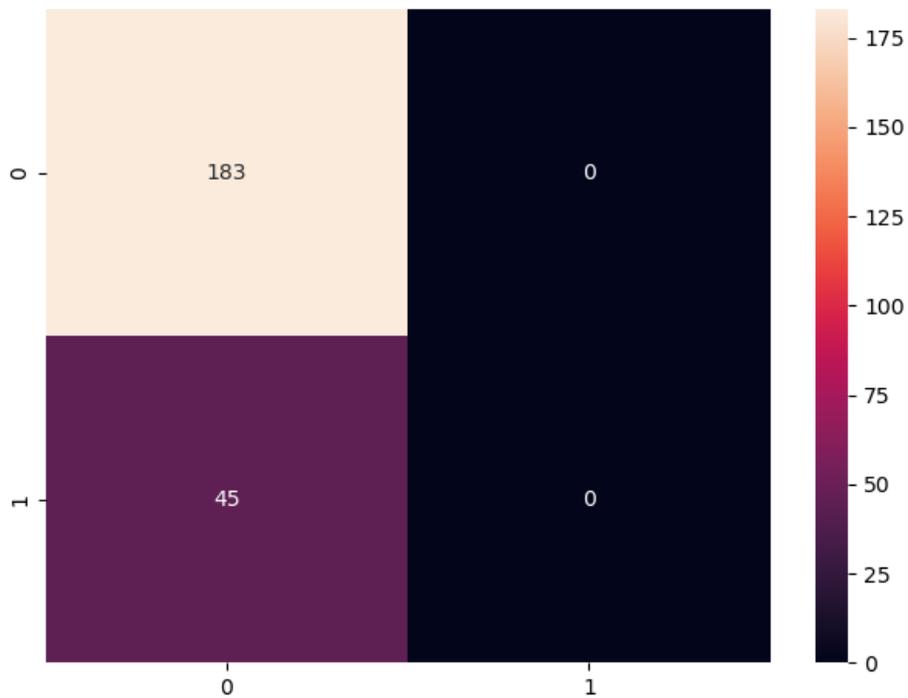


Figura 113 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

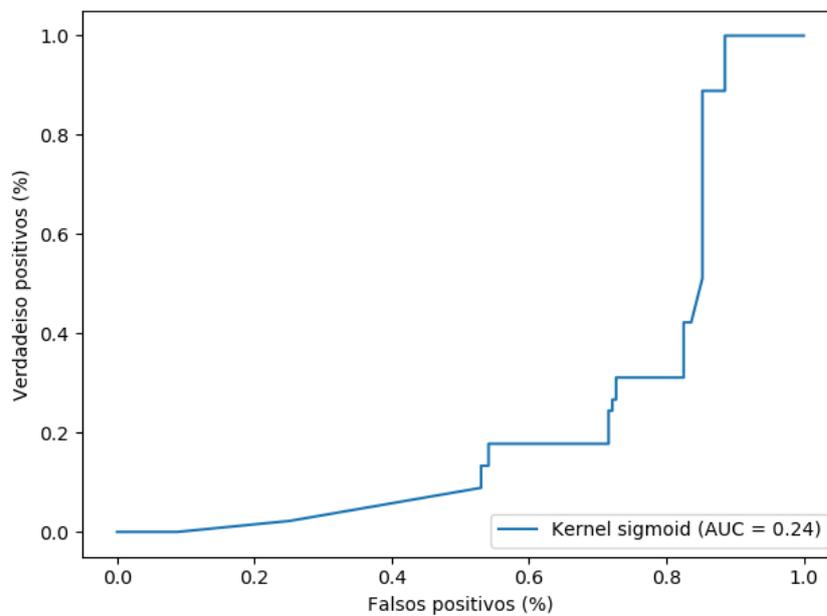


Figura 114 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

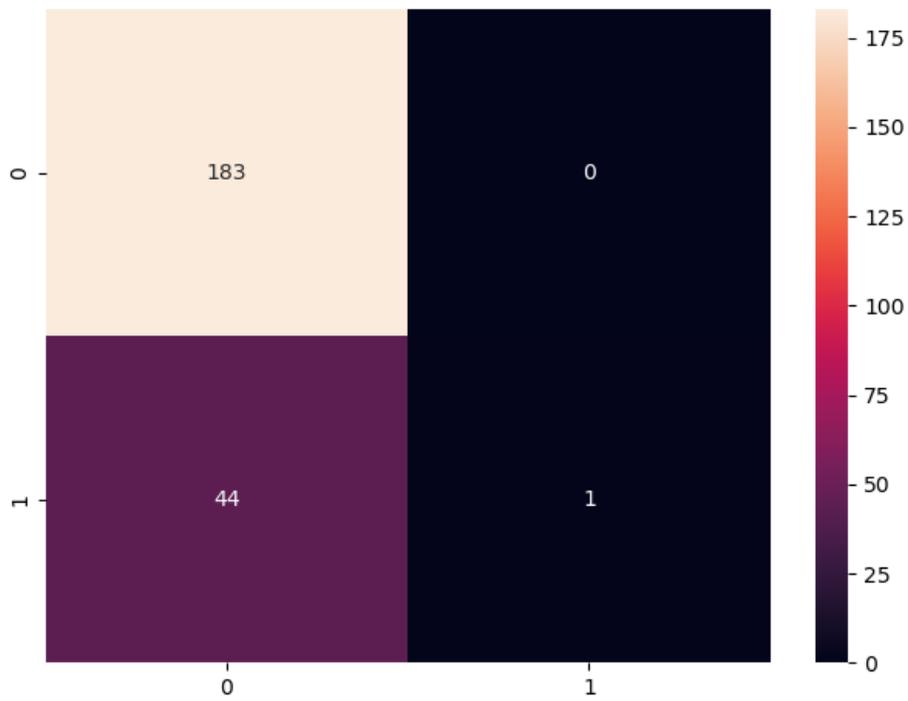


Figura 115 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

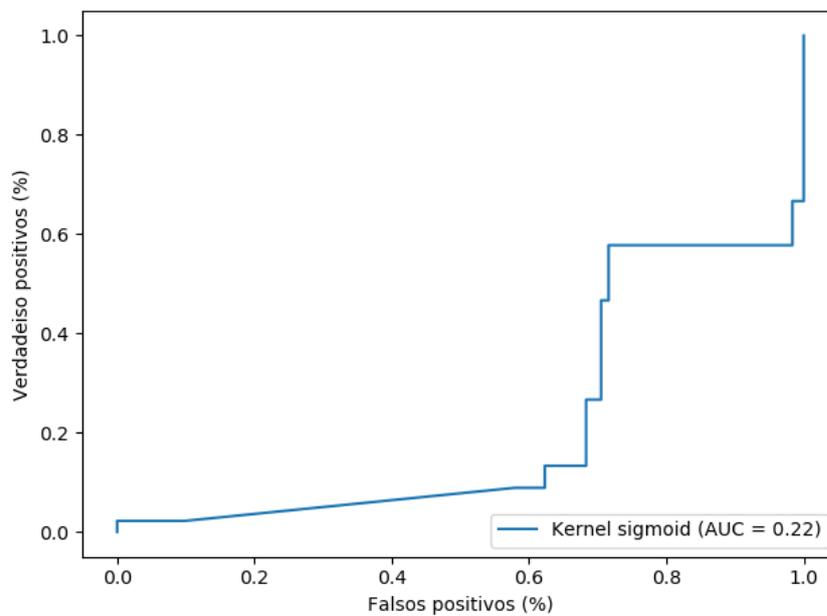


Figura 116 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.

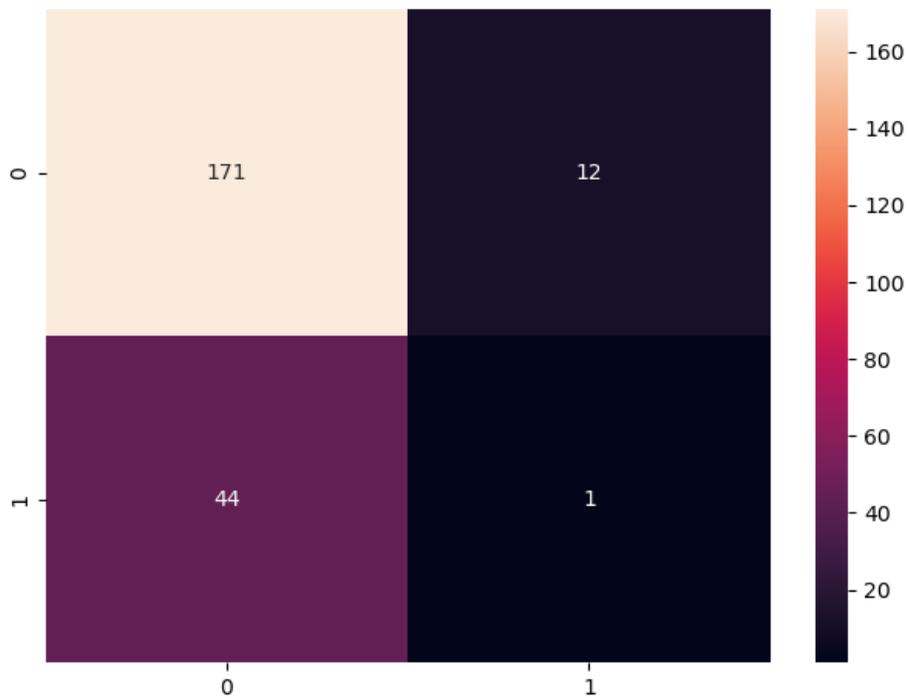
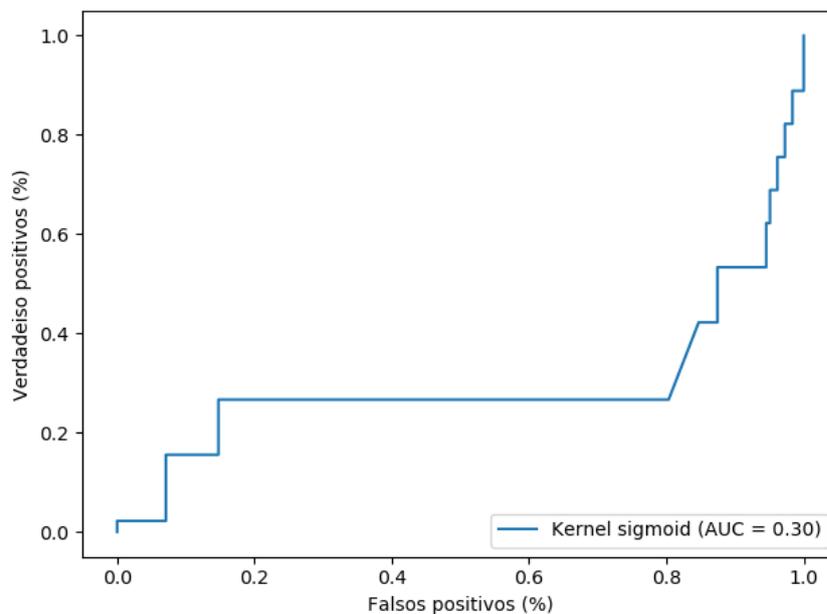


Figura 117 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 50% teste 50% treinamento.



A Tabela 19 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores

com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 70% para treinamento e 30% para teste.

Tabela 19 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 70% para treinamento e 30% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,81	0,90	0,85	106	0,75
Classificador do Estágio 2	0,80	0,99	0,89	106	0,80
Classificador do Estágio 3	0,78	1,0	0,88	106	0,78
Classificador do Estágio 4	0,81	1,00	0,89	106	0,82
Classificador do Estágio 5	0,78	0,96	0,86	106	0,77

As 118, 120, 122, 124, e 126 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 19. As Figuras 129, 131, 133, 137 e 127 ilustram as curvas ROC referentes à mesma tabela.

Figura 118 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

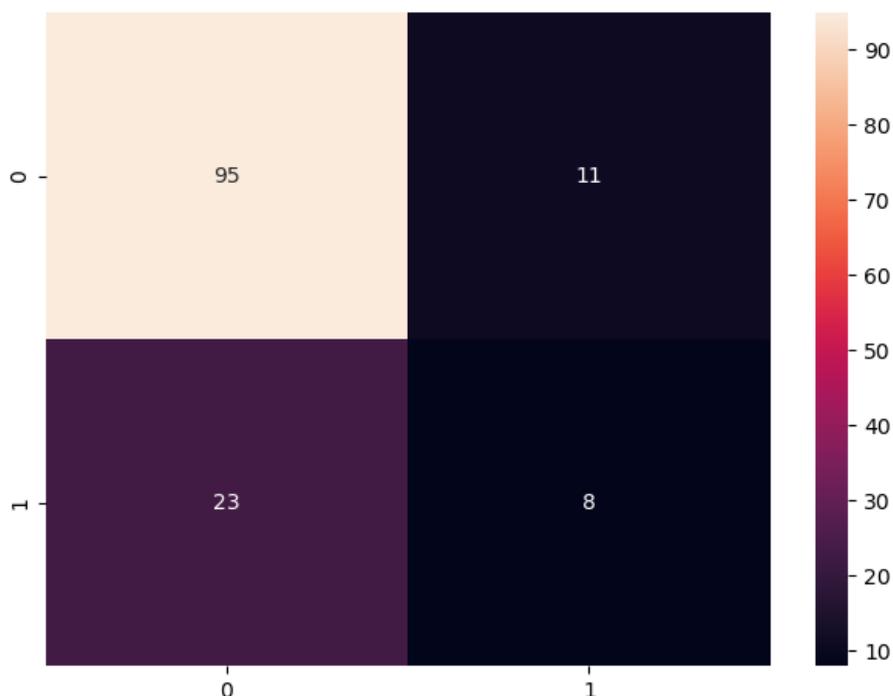


Figura 119 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

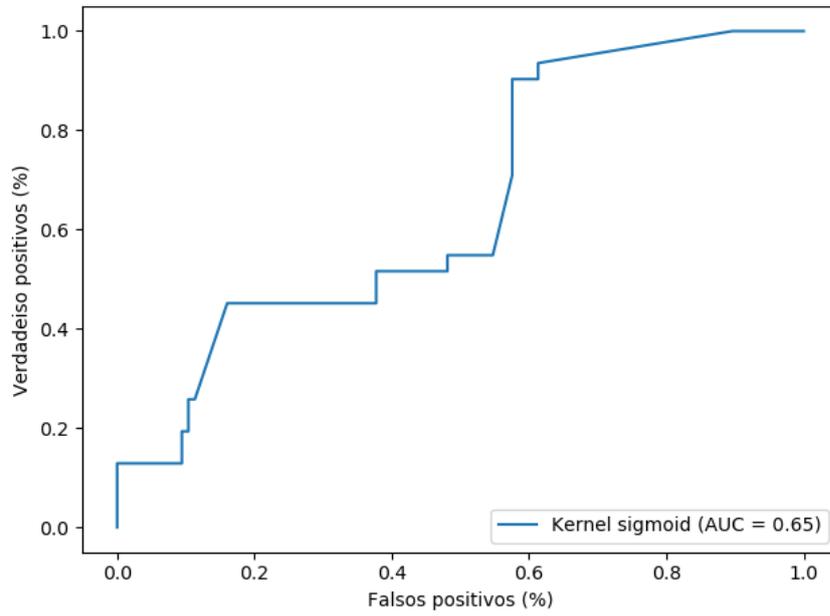


Figura 120 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

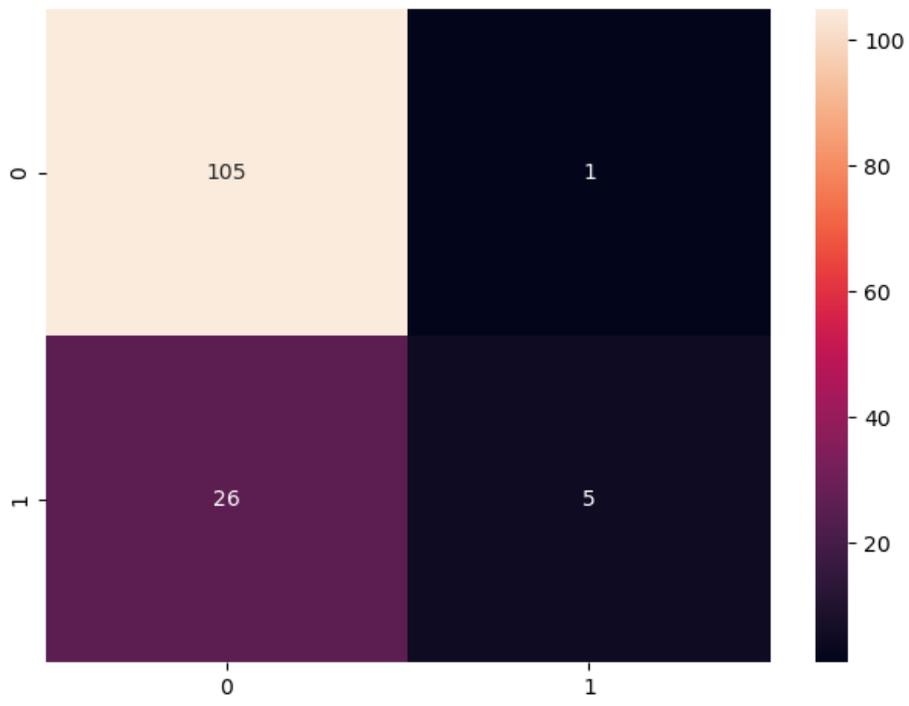


Figura 121 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

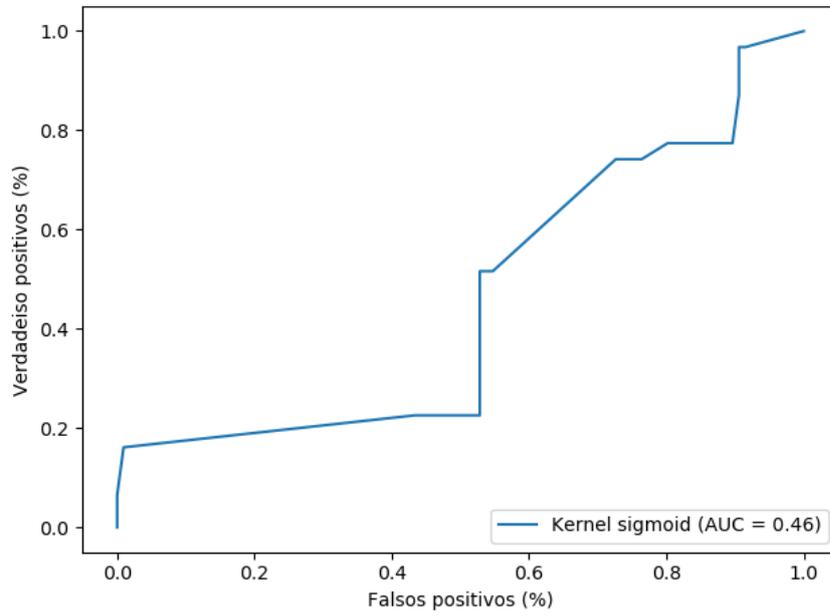


Figura 122 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

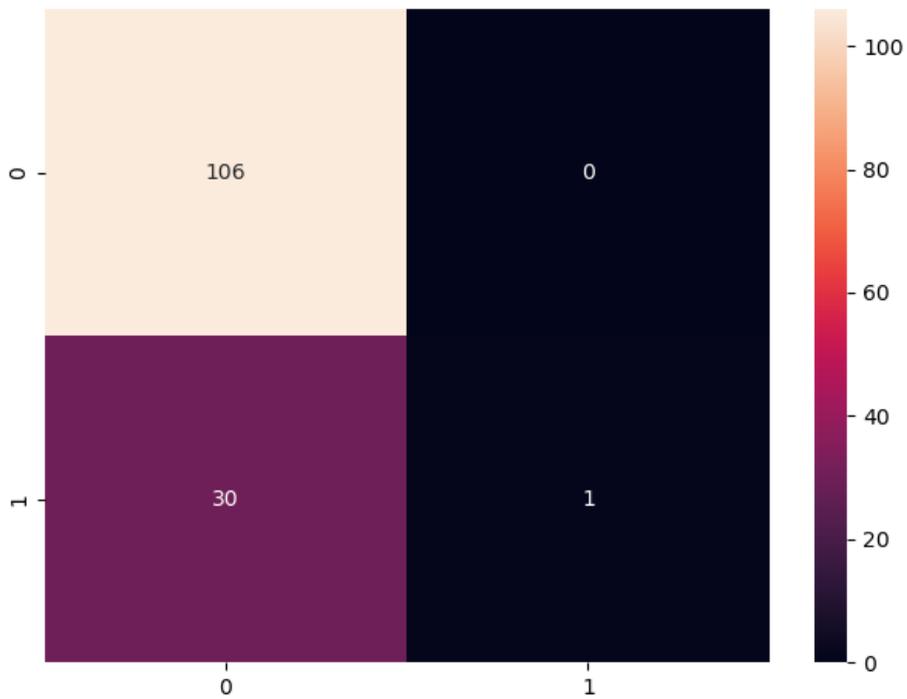


Figura 123 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

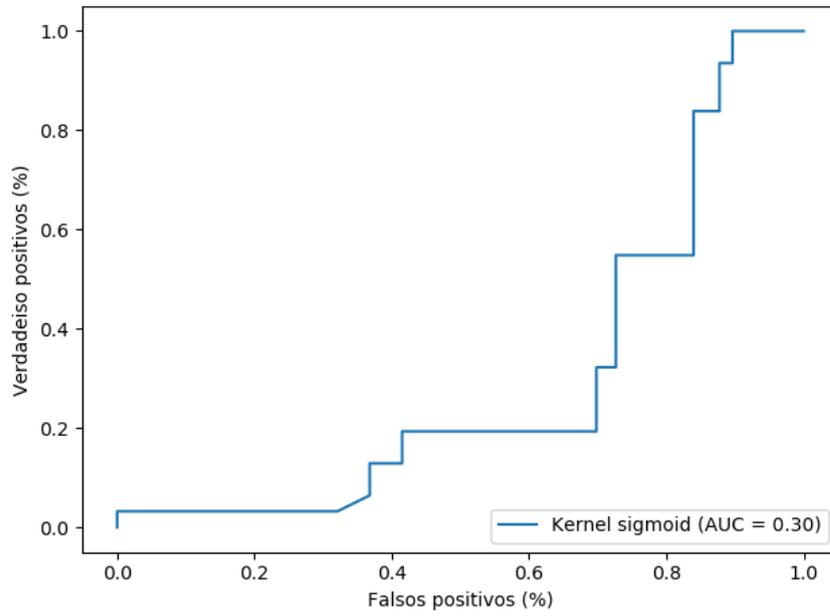


Figura 124 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

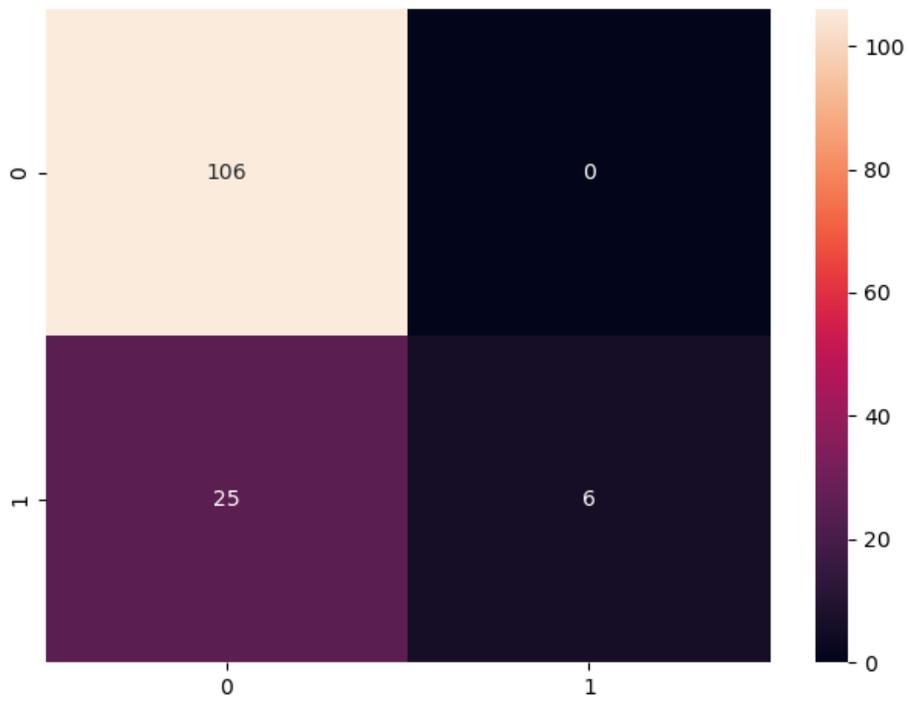


Figura 125 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

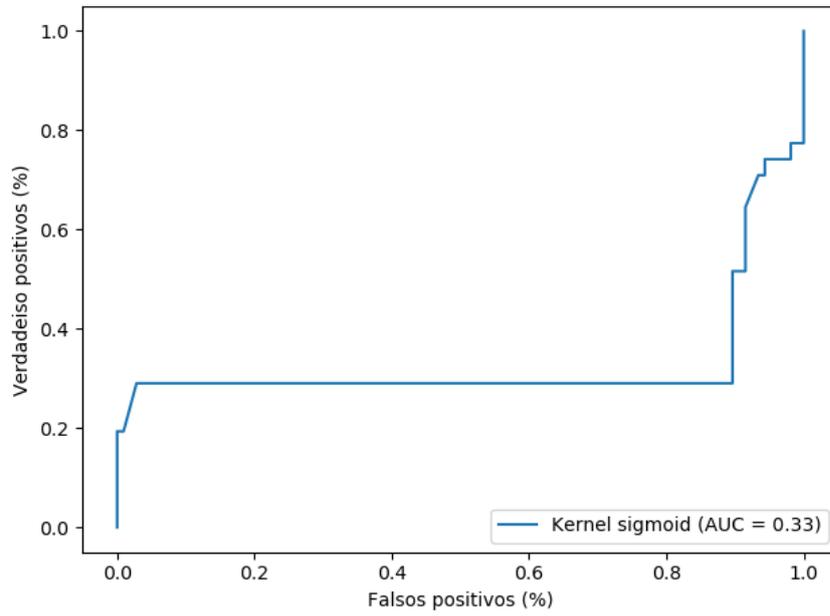


Figura 126 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

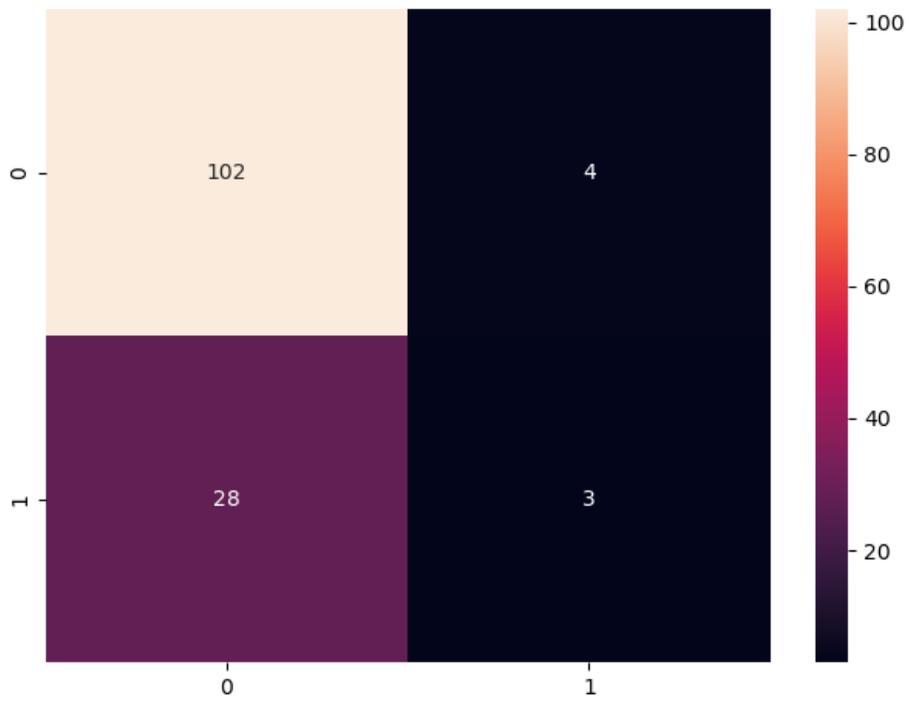
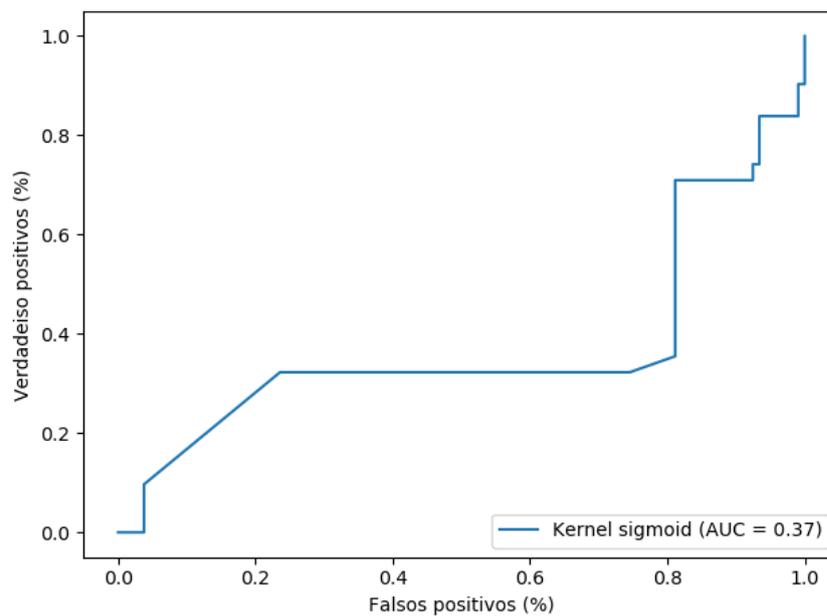


Figura 127 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.



A Tabela 20 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 70% para treinamento e 30% para teste.

Tabela 20 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 70% para treinamento e 30% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,78	0,94	0,85	106	0,74
Classificador do Estágio 2	0,77	0,95	0,85	106	0,74
Classificador do Estágio 3	0,77	1,00	0,87	106	0,77
Classificador do Estágio 4	0,78	1,00	0,88	106	0,78
Classificador do Estágio 5	0,76	0,92	0,83	106	0,72

As 128, 130, 132, 134, e 136 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 20. As Figuras 129, 131, 133, 137 e 127 ilustram as curvas ROC referentes à

mesma tabela.

Figura 128 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

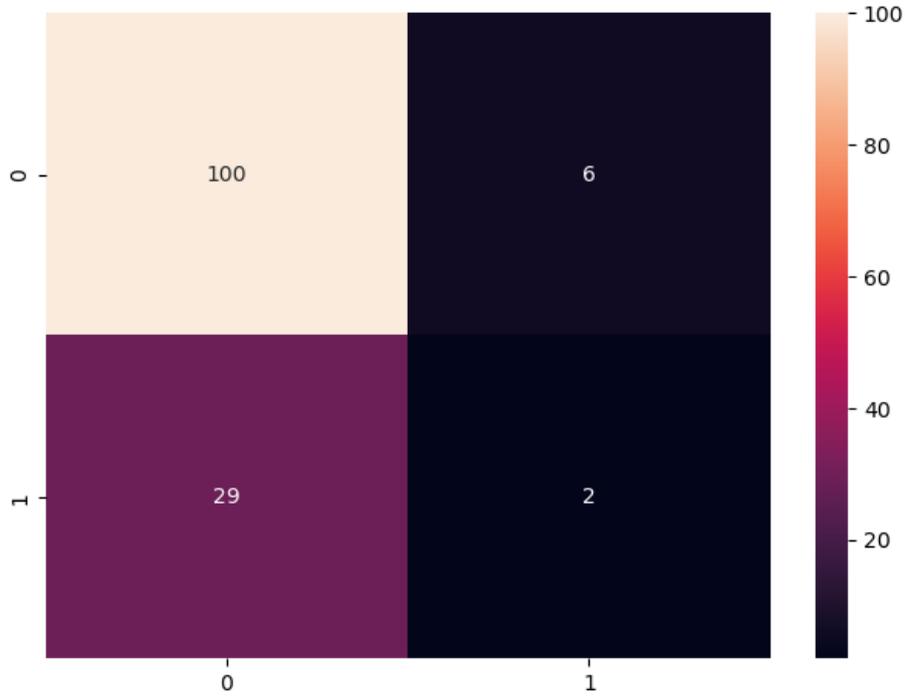


Figura 129 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

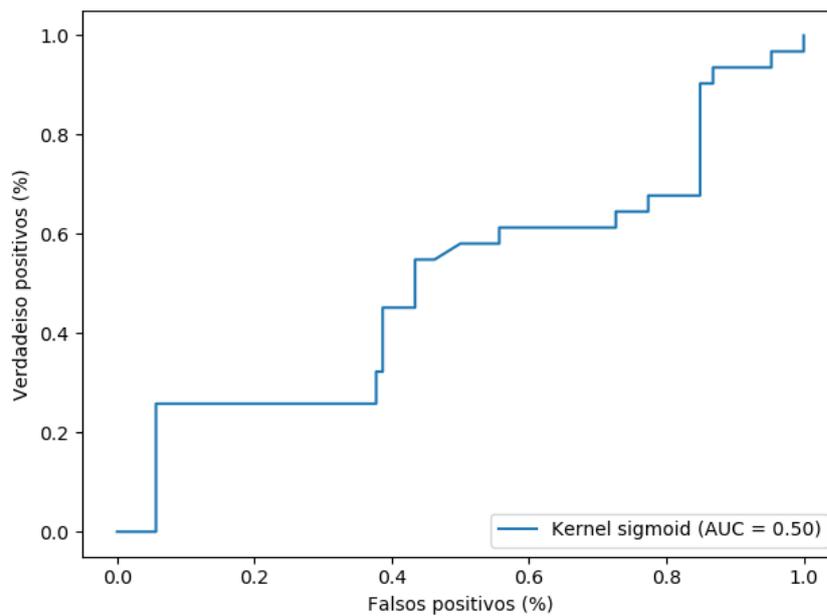


Figura 130 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

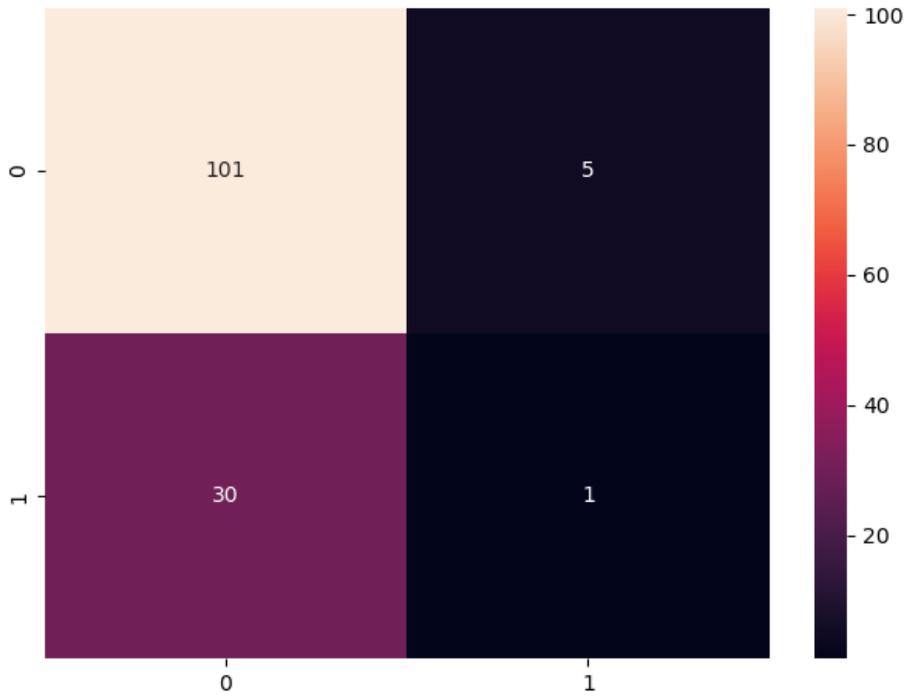


Figura 131 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

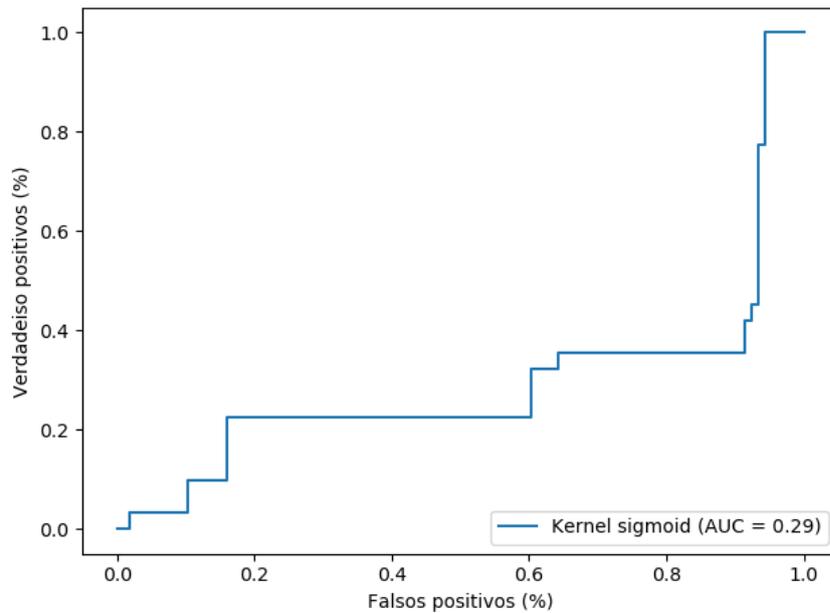


Figura 132 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

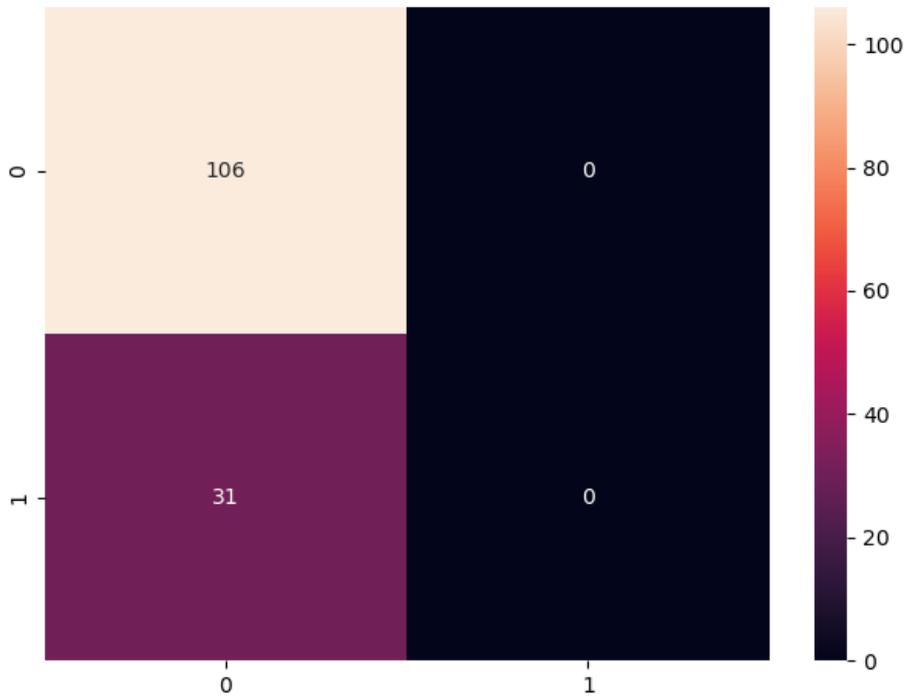


Figura 133 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

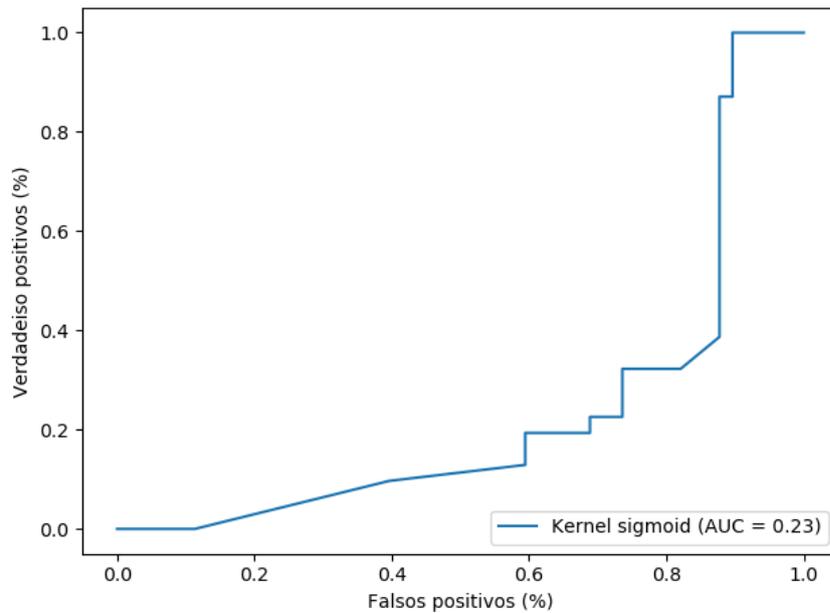


Figura 134 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

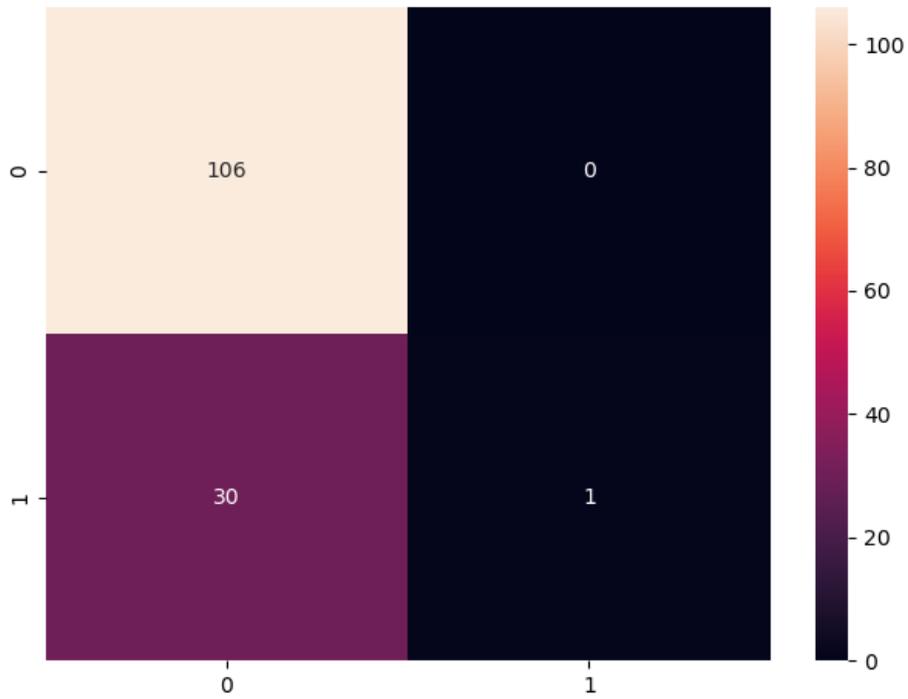


Figura 135 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

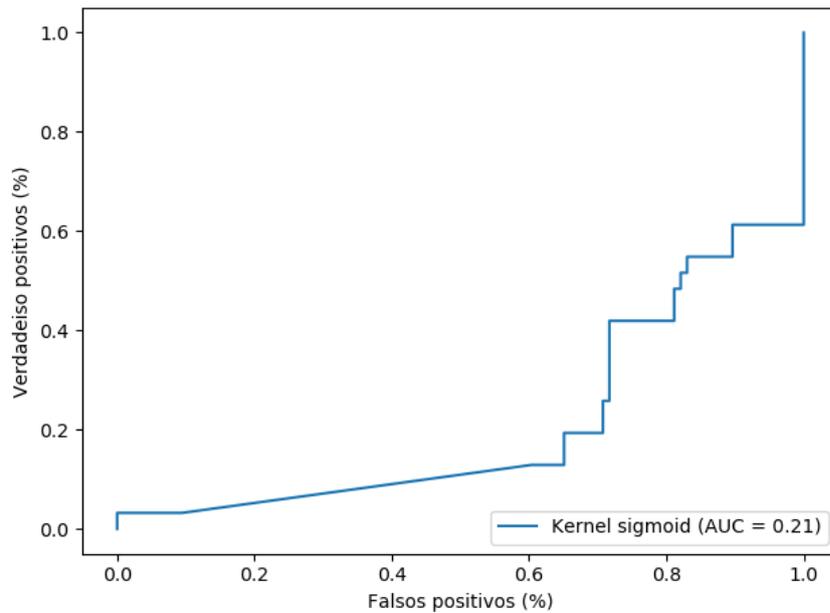


Figura 136 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.

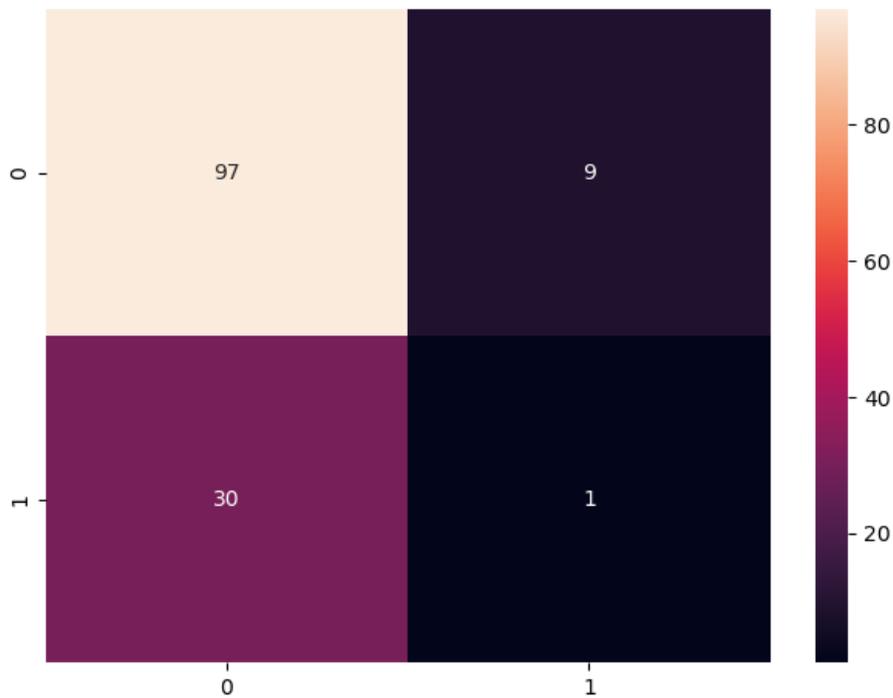
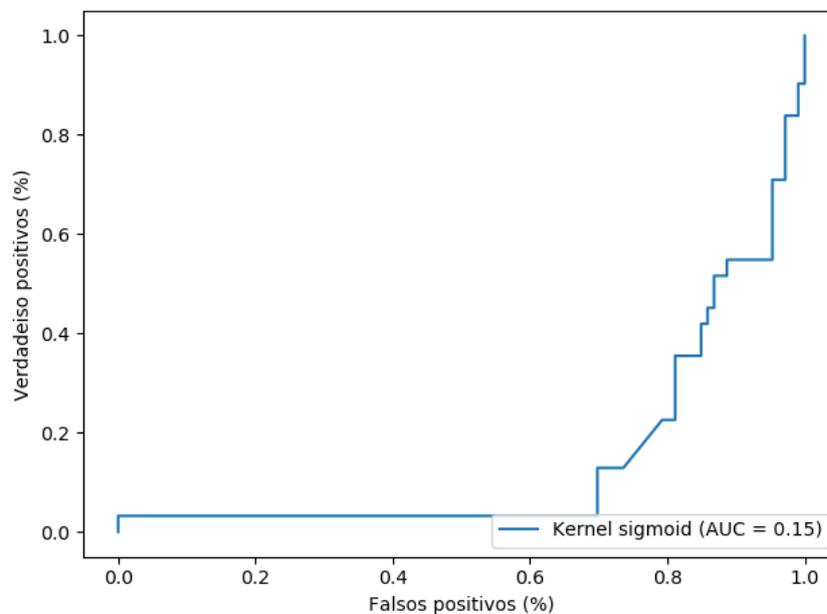


Figura 137 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 70% teste 30% treinamento.



A Tabela 21 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro Gaussiano, com classificadores

com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 80% para treinamento e 20% para teste.

Tabela 21 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro Gaussiano para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 80% para treinamento e 20% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,82	0,89	0,85	72	0,76
Classificador do Estágio 2	0,79	1,00	0,88	72	0,79
Classificador do Estágio 3	0,79	1,00	0,88	72	0,79
Classificador do Estágio 4	0,83	0,97	0,90	72	0,83
Classificador do Estágio 5	0,79	0,96	0,87	72	0,77

As 138, 140, 142, 144, e 146 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 21. As Figuras 139, 141, 143, 145 e 147 ilustram as curvas ROC referentes à mesma tabela.

Figura 138 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

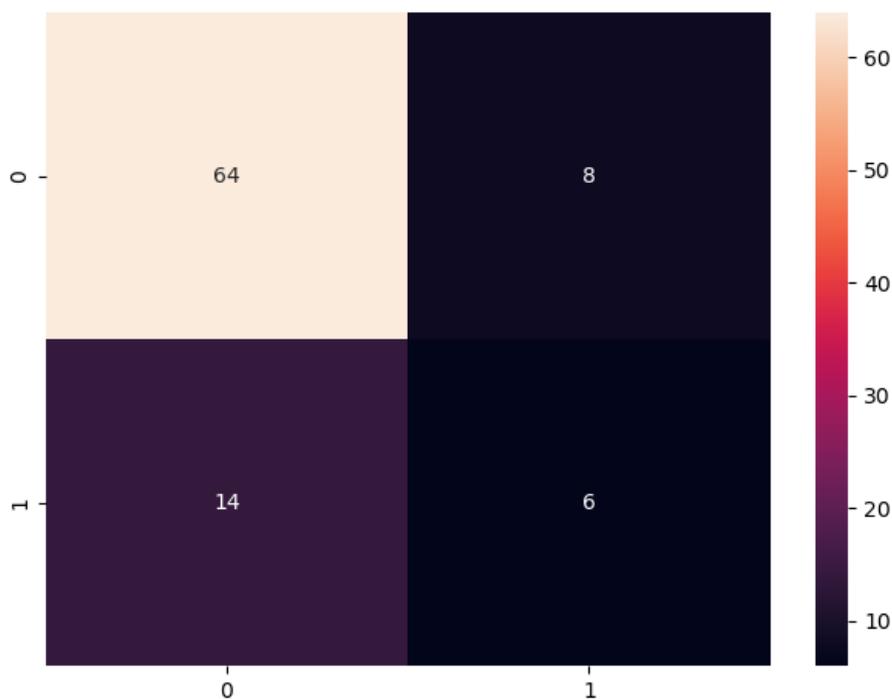


Figura 139 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

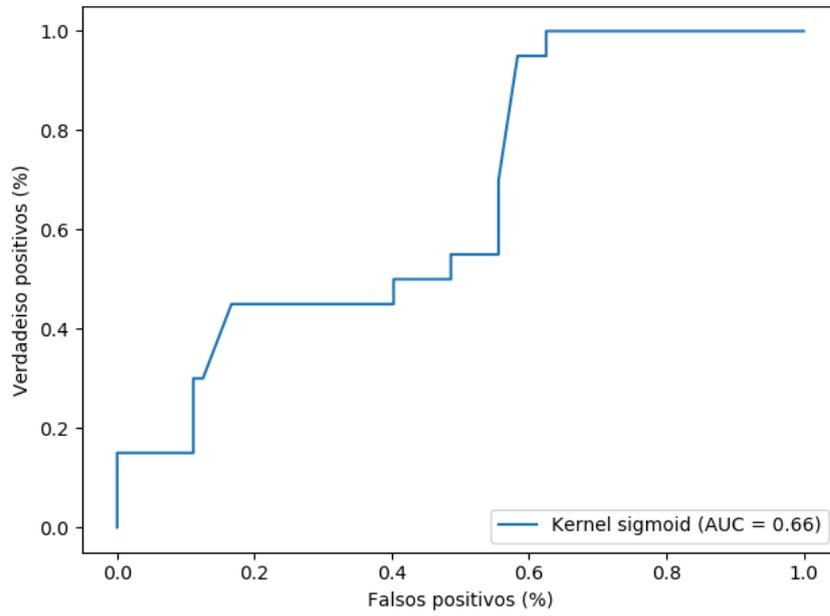


Figura 140 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

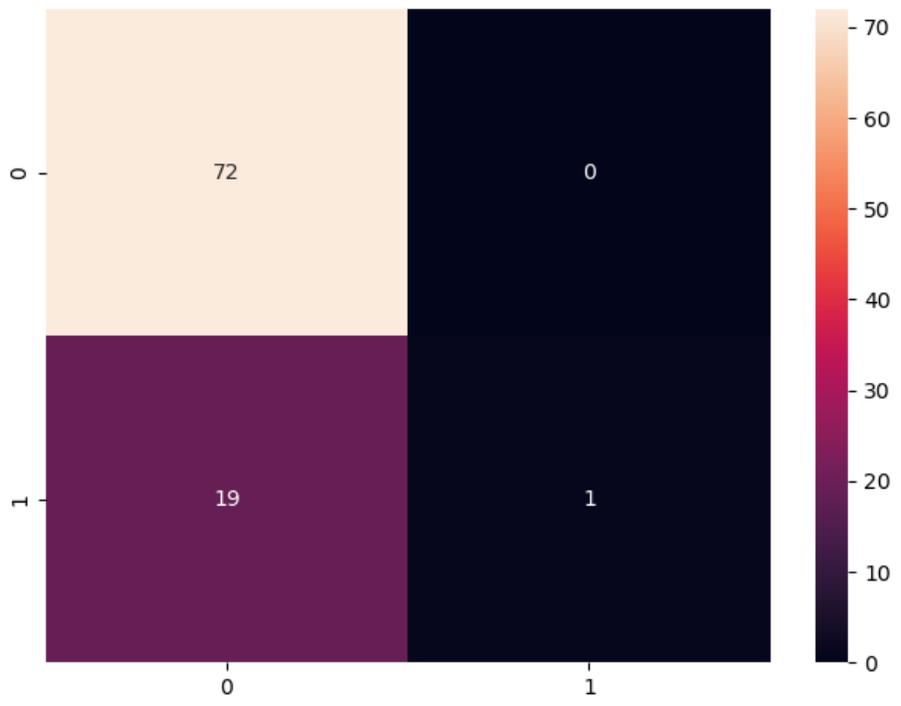


Figura 141 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

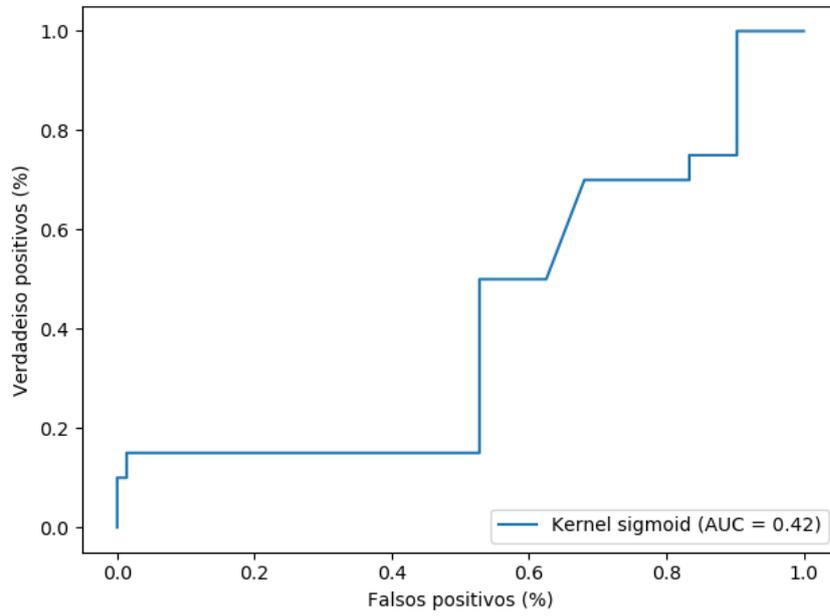


Figura 142 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

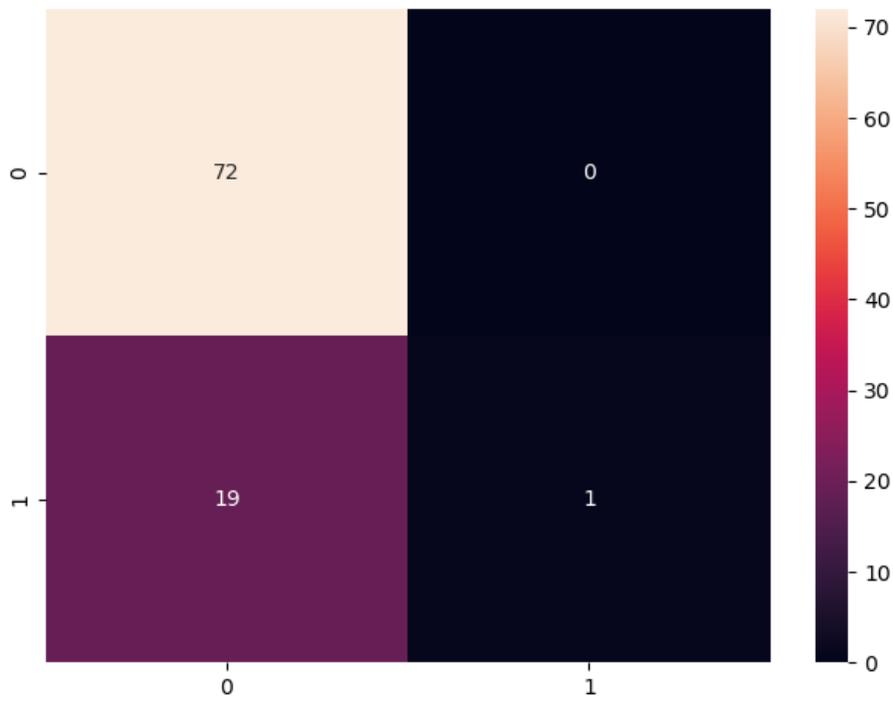


Figura 143 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

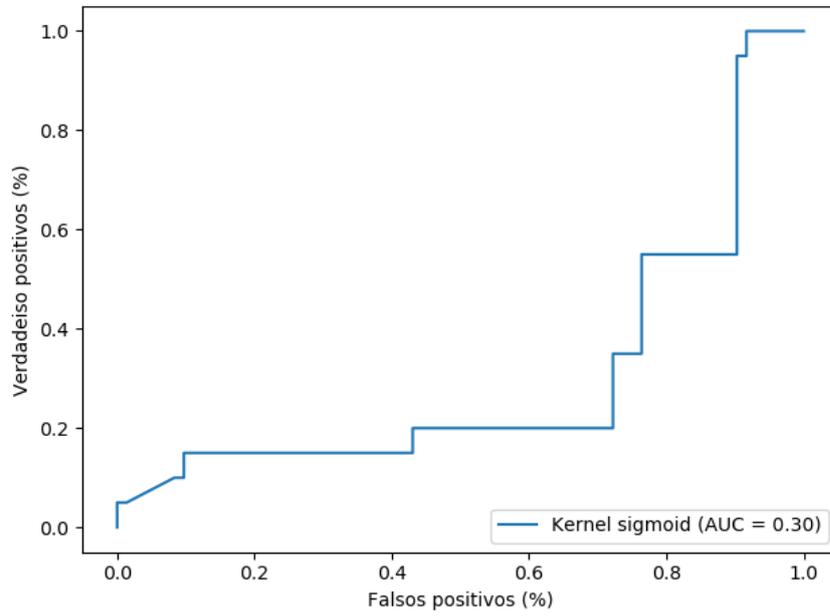


Figura 144 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

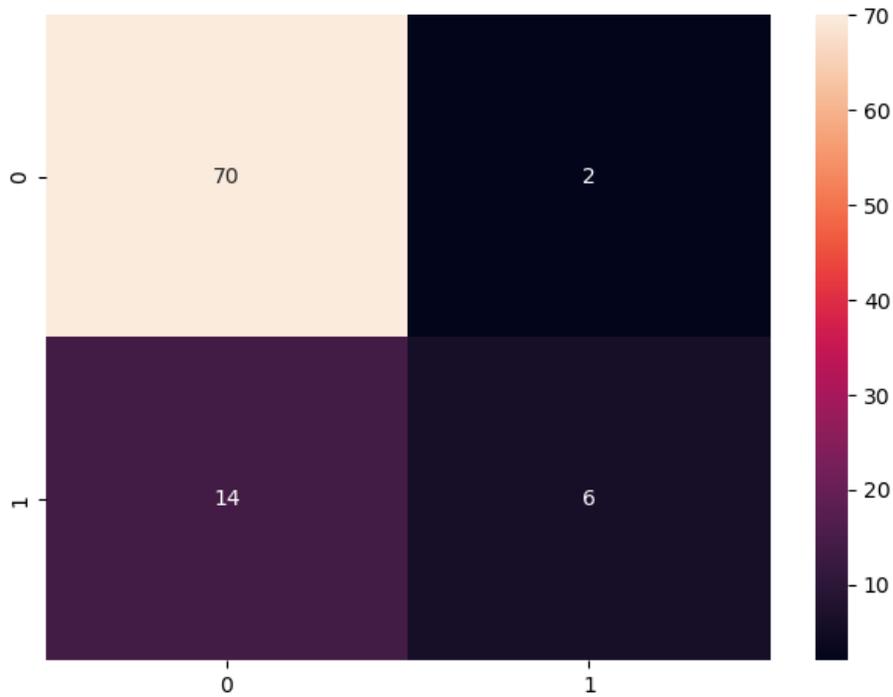


Figura 145 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

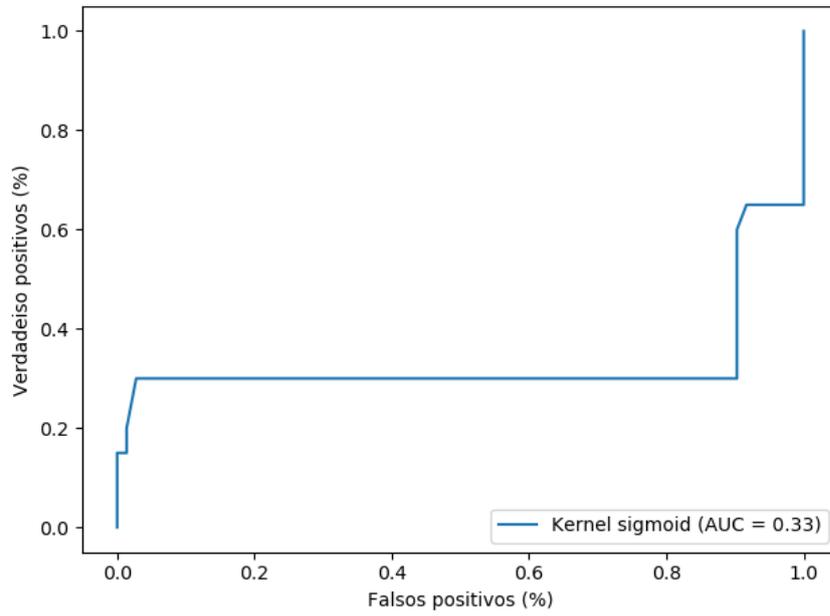


Figura 146 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

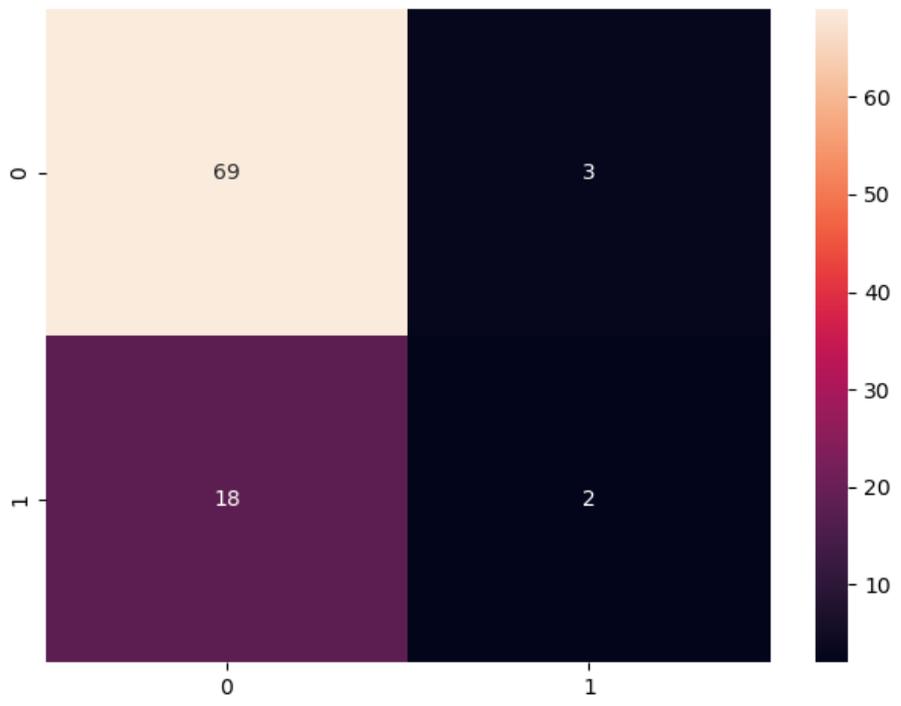
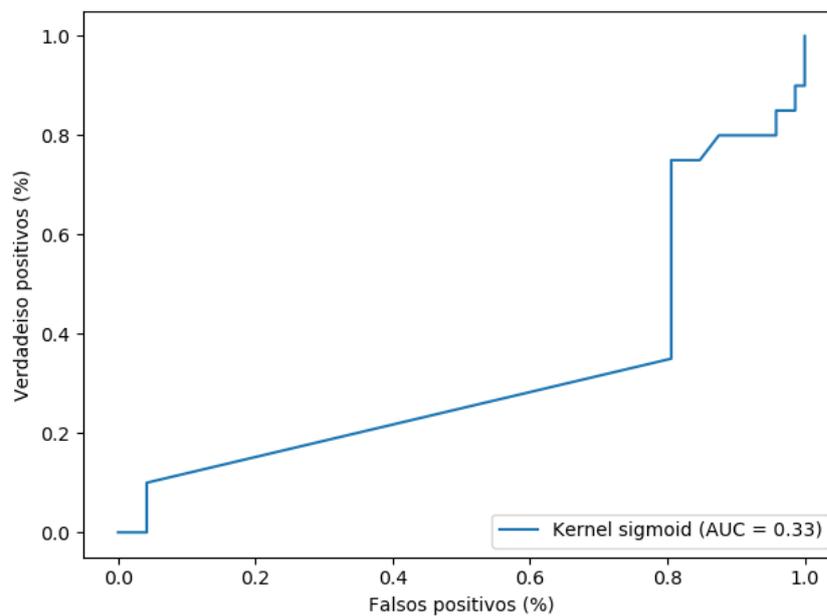


Figura 147 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.



A Tabela 22 apresenta os resultados obtidos a partir de ensaios realizados para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) em seus diferentes estágios de desenvolvimento, a partir de imagens filtradas com o filtro *Non-local Means*, com classificadores com função *Kernel* sigmoide de modo a compor o conjunto de classificadores SVM proposto no método. Nessa etapa foi utilizada a proporção de 80% para treinamento e 20% para teste.

Tabela 22 – Resultados referentes aos ensaios realizados em imagens filtradas pelo filtro *Non-local Means* para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) por meio do classificador SVM com função sigmoide utilizando a proporção de 80% para treinamento e 20% para teste.

Classificador	Precisão	Revocação	<i>F1-Score</i>	Vetores de Suporte	Acurácia (%)
Classificador do Estágio 1	0,76	0,88	0,81	72	0,68
Classificador do Estágio 2	0,80	0,97	0,88	72	0,78
Classificador do Estágio 3	0,78	1,00	0,88	72	0,78
Classificador do Estágio 4	0,76	0,90	0,83	72	0,71
Classificador do Estágio 5	0,76	0,88	0,81	72	0,68

As 148, 150, 152, 154, e 156 ilustram as matrizes de confusão referente aos resultados exposta na Tabela 22. As Figuras 149, 151, 153, 155 e 157 ilustram as curvas ROC referentes à

mesma tabela.

Figura 148 – Matriz de confusão do Classificador 1 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

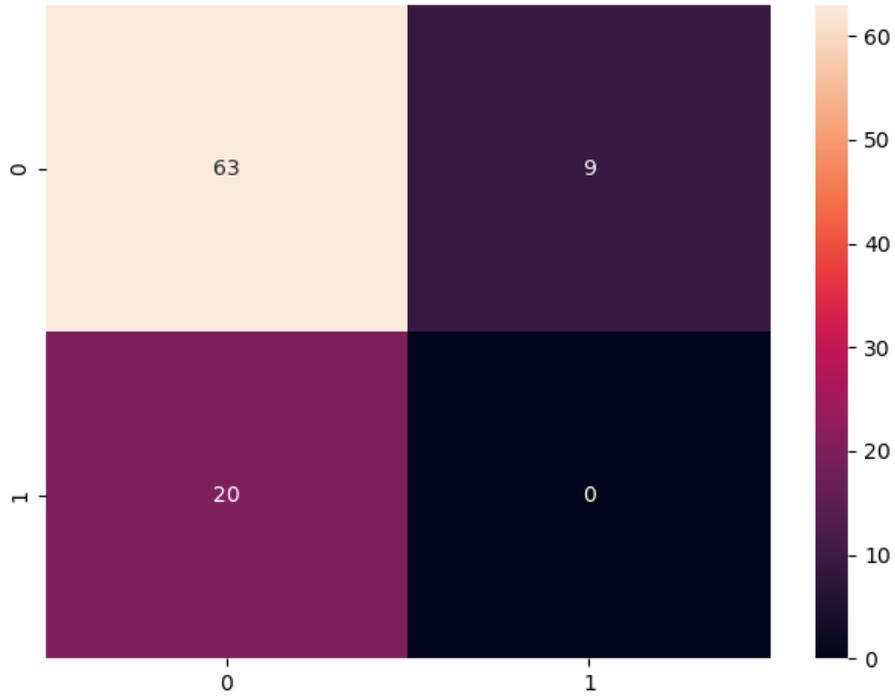


Figura 149 – Curva ROC do Classificador 1 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

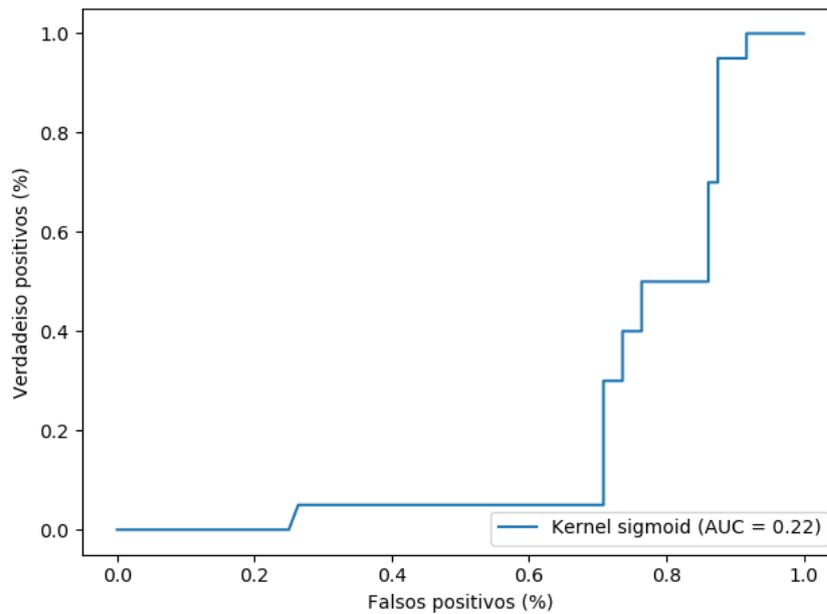


Figura 150 – Matriz de confusão do Classificador 2 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

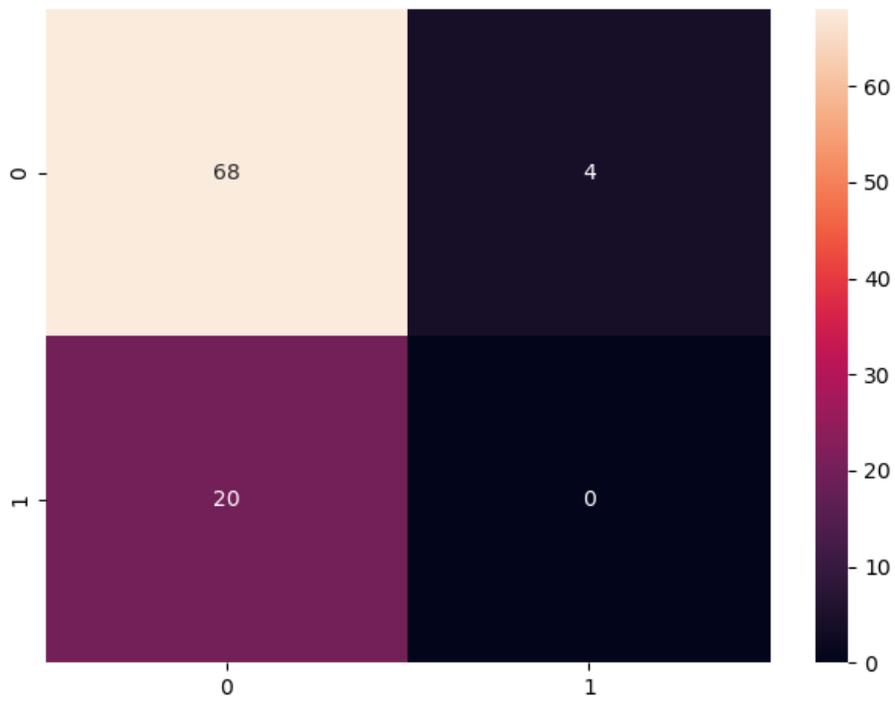


Figura 151 – Curva ROC do Classificador 2 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

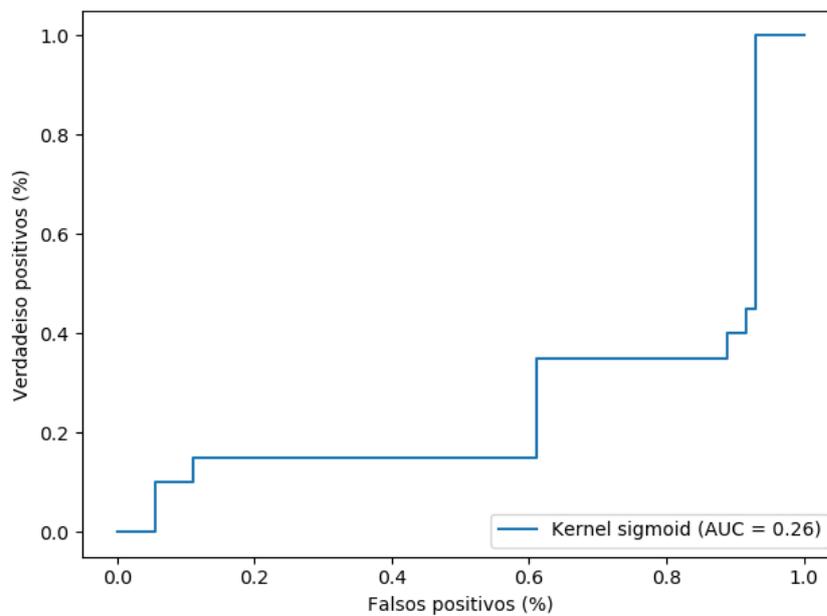


Figura 152 – Matriz de confusão do Classificador 3 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

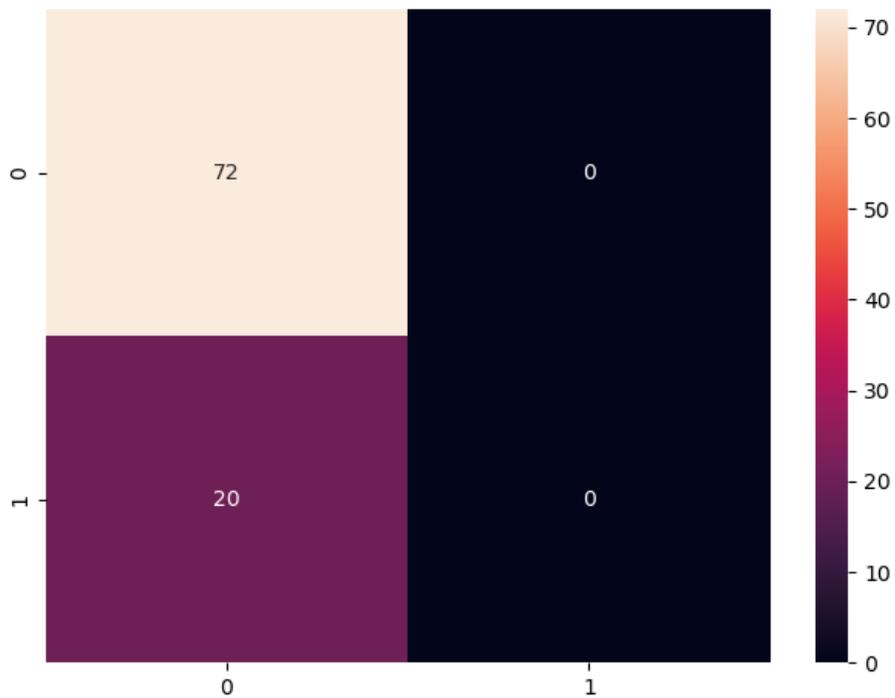


Figura 153 – Curva ROC do Classificador 3 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

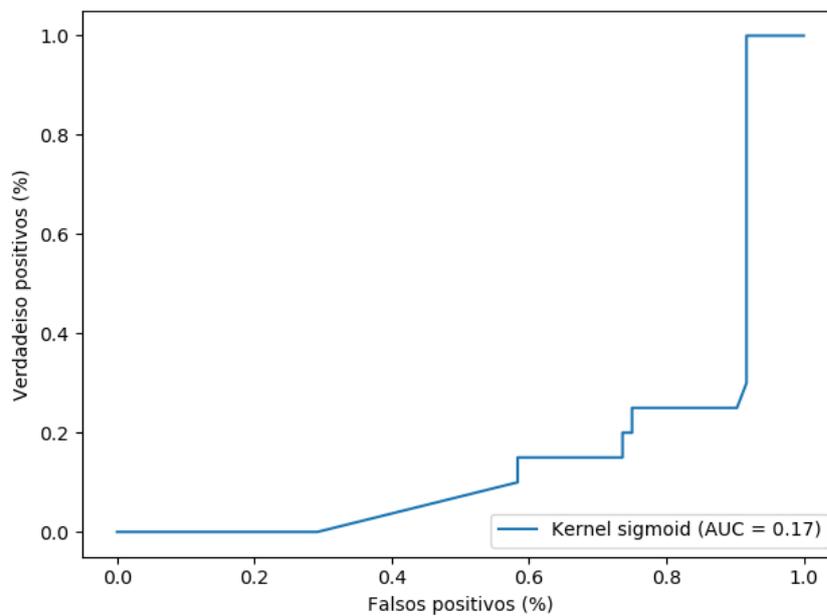


Figura 154 – Matriz de confusão do Classificador 4 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

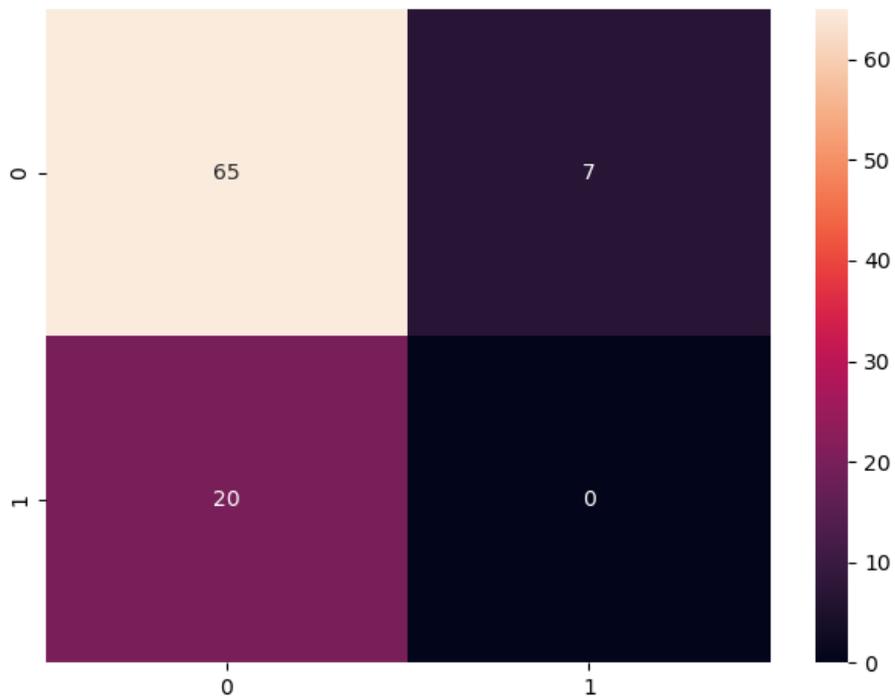


Figura 155 – Curva ROC do Classificador 4 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

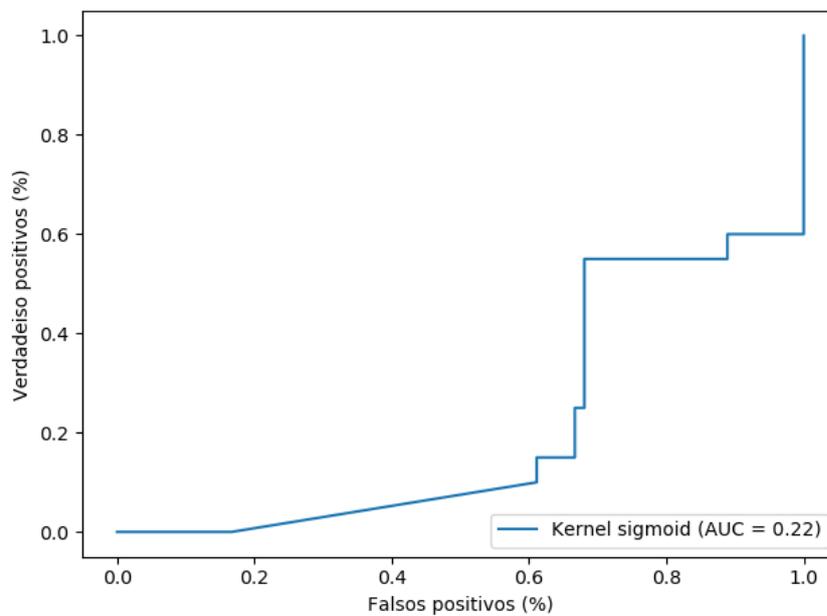


Figura 156 – Matriz de confusão do Classificador 5 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.

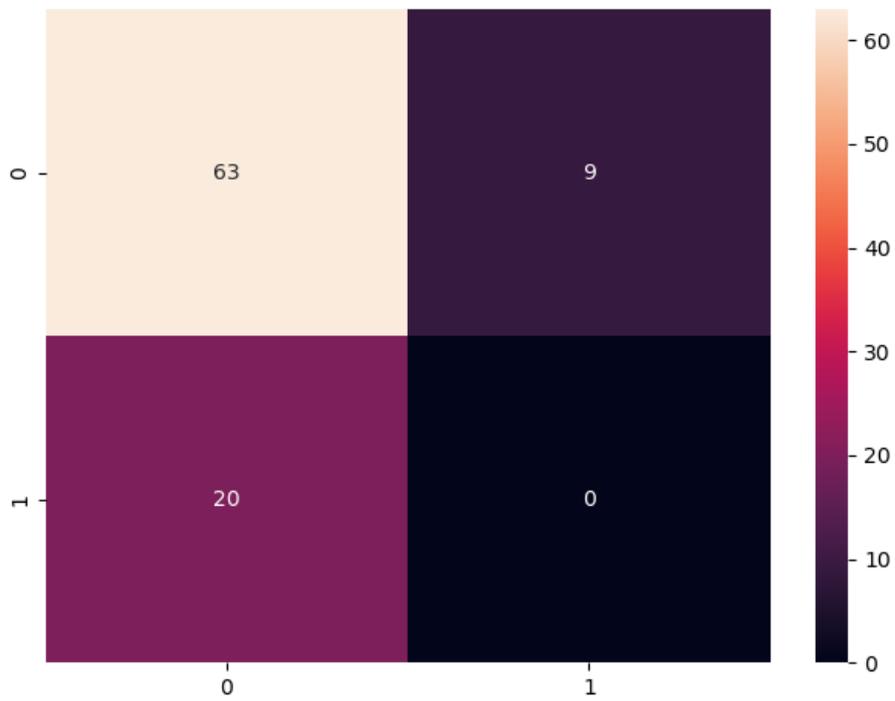
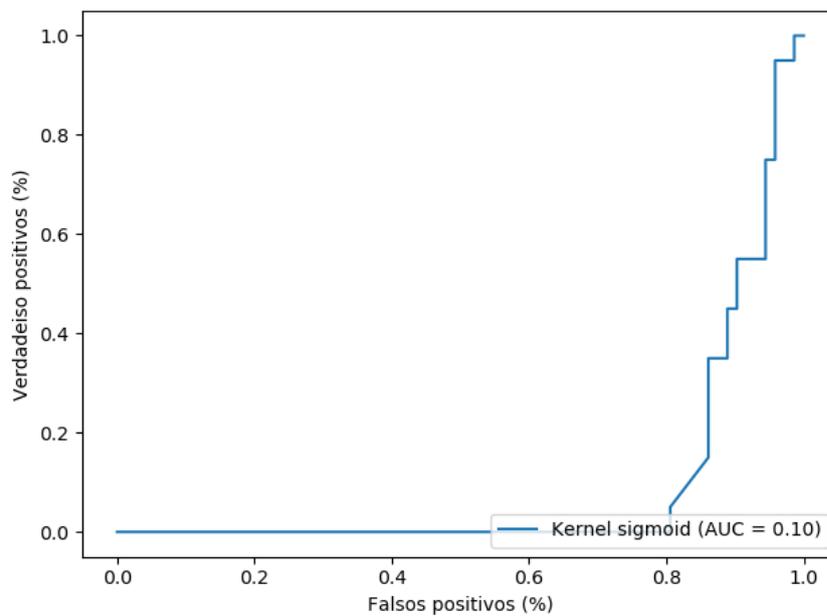


Figura 157 – Curva ROC do Classificador 5 com função *kernel* sigmoide e proporção 80% teste 20% treinamento.



3.7.1 Avaliação da etapa de treinamento do conjunto de classificadores SVM

A partir dos resultados obtidos com os treinamentos e testes de cada classificador SVM foi possível avaliar que a taxa de precisão se manteve em um valor considerado satisfatório para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*), ou seja, acima de 70%, assim como, ao analisar a taxa de acurácia dos classificadores.

Para a proporção de 50% utilizada para teste e treinamento, observou-se que o desempenho dos classificadores SVM com função *kernel* sigmoide, no que se refere a taxa de acurácia, obtiveram melhores resultados para a classificação de padrões de todos os estágios, com acurácia de 78% para os estágios 1 e 2, 81% para os estágios 3 e 4 e 79% para o estágio 5. Ao avaliar os classificadores pela taxa de precisão, observou-se também a melhor eficácia dos classificadores com função *kernel* sigmoide.

Para a proporção de 70% e 30% referentes a treinamento e teste, respectivamente, observou-se, assim como para a proporção 50% x 50%, que o desempenho dos classificadores SVM com função *kernel* sigmoide, obtiveram os melhores resultados para a classificação de todos os estágios de desenvolvimento da lagarta, com taxa de acurácia de 74%, 80%, 78%, 82% e 77% para os estágios 1, 2, 3, 4 e 5, respectivamente. Quando analisada a taxa de precisão os classificadores com função *kernel* sigmoide também obtiveram melhores resultados.

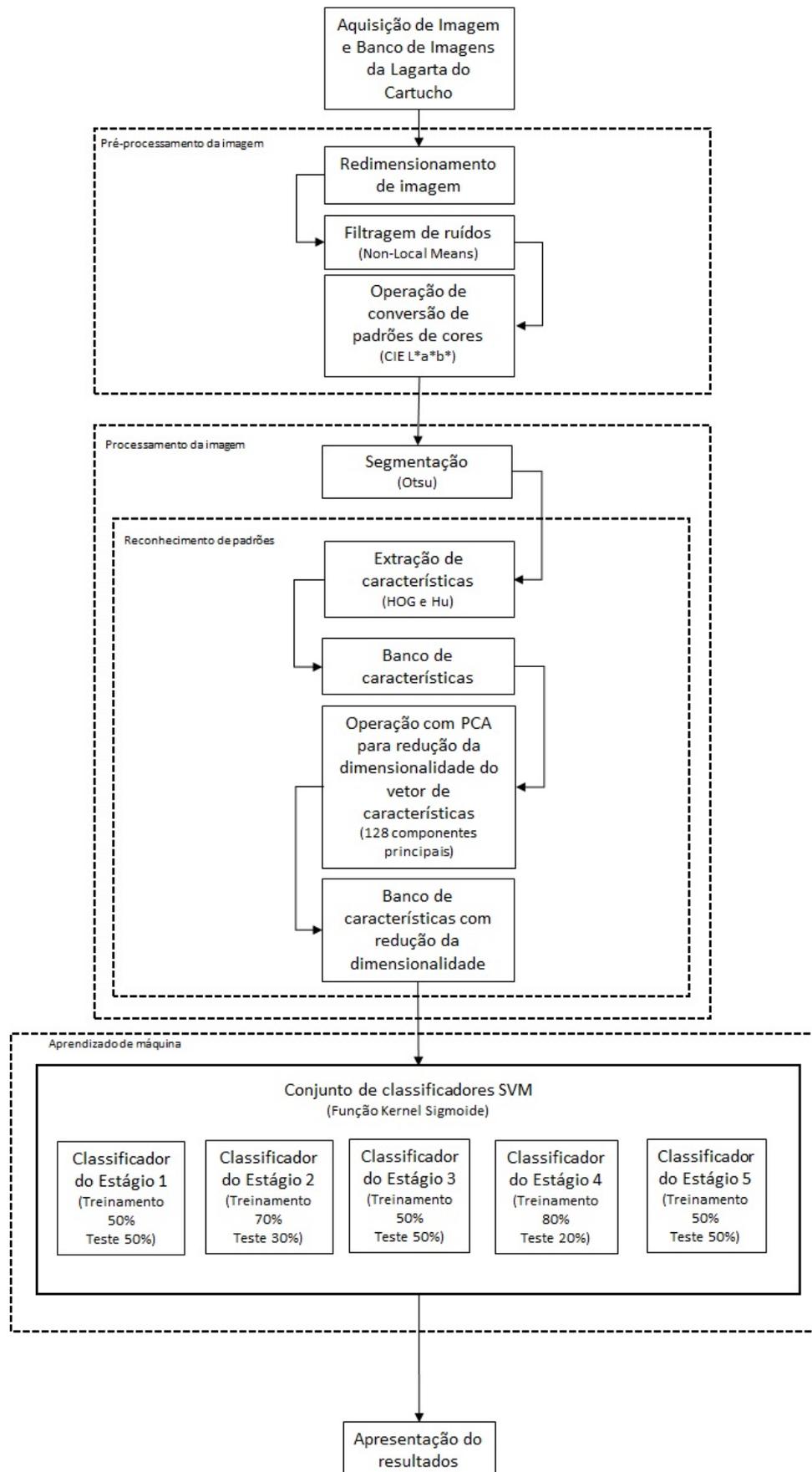
Para a proporção de 80% e 20% referentes a treinamento e teste, respectivamente, o desempenho dos classificadores SVM com função *kernel* sigmoide também obtiveram os melhores resultados para todos os estágios de desenvolvimento da lagarta, com taxa de acurácia de 74%, 79%, 78%, 80% e 77%, para os estágios 1, 2, 3, 4 e 5, respectivamente. Porém, ao avaliar a taxa de precisão, os classificadores SVM *kernel* linear obtiveram melhores resultados, com taxa de precisão de 85% e 82% para os estágios 1 e 5, respectivamente.

A parametrização do conjunto de classificadores SVM levou em consideração, primeiramente, a taxa de precisão para a classificação do estágio alvo da lagarta e em seguida a taxa de acurácia, conforme exibido na Tabela 23. A Figura 158 ilustra o método estabelecido para classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho.

Tabela 23 – Conjunto de classificadores SVM para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*).

Classificador	Função <i>Kernel</i>	Parâmetro C	Parâmetro Delta (δ)
Classificador do Estágio 1	Sigmoide	10	1,0
Classificador do Estágio 2	Sigmoide	100	1,0
Classificador do Estágio 3	Sigmoide	0,1	0,1
Classificador do Estágio 4	Sigmoide	0,1	10
Classificador do Estágio 5	Sigmoide	1	1

Figura 158 – Diagrama do método para classificação de padrões de da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho.



O código desenvolvido em linguagem de programação *Python* pode ser visualizado no *link* <https://github.com/alexbertolla/classificacao_padroes_lagarta_cartucho_milho.git>

Capítulo 4

CONCLUSÕES

Este trabalho apresentou como principal contribuição o estabelecimento de um método para classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho. Essa praga uma vez reconhecida auxilia na organização de prognósticos, o que é essencial para a diminuição de perdas na cultura.

4.1 Principais conclusões

Neste contexto, uma série de estudos foram realizados no âmbito da área de processamento de imagens e sinais, de forma a se buscar a estrutura do método para a classificação dessa praga em seus principais estágios de desenvolvimento.

A validação do método proposto foi estabelecida a partir da aquisição de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) em banco de imagens *online*, *Insect Image*, e também a partir da técnica *data augmentation*. Essa técnica permitiu o aumento significativo do banco de imagens, representando os principais estágios de desenvolvimento da lagarta e também, possíveis locais onde a praga seria encontrada na planta, como por exemplo, em folhas e espigas de milho.

Com os resultados obtidos na etapa de pré-processamento das imagens, foi possível concluir que, no cenário proposto, o filtro *Non-local Means* foi mais eficiente que o filtro Gaussiano para suavização de imagens com ruídos do tipo Gaussiano. Tal conclusão pôde ser verificada a partir da análise das métricas MSE e PSNR, utilizadas para validação do processo de filtragem de ruídos.

A técnica de limiarização e *pixels* sementes a partir do componente H do padrão de cor HSV, se mostrou eficaz para a segmentação de imagens da Lagarta do Cartucho (*Spodoptera frugiperda*) presente em folhas, entretanto, o mesmo processo não obteve os mesmos resultados para a segmentação da lagarta em espigas de milho. Apesar da eficiência dessa técnica de segmentação em imagens da lagarta em folhas, o processo de escolha manual dos *pixels* sementes pode se tornar um processo demorado e ineficiente dependendo da complexidade das imagens,

assim como, da quantidade de imagens que se deseja processar.

Entretanto, a automatização da etapa de segmentação de imagens por meio do método de Otsu e do padrão de cor CIE $L^*a^*b^*$, se mostrou eficiente e eficaz tanto para a segmentação de imagens da lagarta presente em folhas como em espigas de milho. Os resultados mostraram ainda, que para a segmentação de imagens onde a lagarta se encontra sobre folhas, a divisão da segmentação em três etapas, segmentação do componente a^* , segmentação do componente b^* e, por fim, a segmentação de forma a interseccionar os resultados da segmentação de a^* com o resultado da segmentação de b^* , resultou em uma imagem onde todas as características da lagarta pôde ser preservada, de modo a permitir que os dados de entrada confiáveis e significativos para as etapas de extração de características e classificação. Vale ressaltar, que mesmo não sendo escopo deste projeto de pesquisa, as técnicas utilizadas no processo de segmentação se mostraram eficazes também para segmentar anomalias presentes em folhas da planta do milho.

No que se refere às técnicas de extração de características, foi possível concluir que a utilização integrada do descritor de textura HOG, assim como, o descritor de momentos invariantes de Hu, no qual se permite extrair informações geométricas, se mostrou eficaz e inovador. O uso da técnica de redução da dimensionalidade do vetor de características por meio de PCA, permitiu ainda a eliminação de informações redundantes e de pouca representatividade para a etapa de classificação dos estágios da Lagarta do Cartucho (*Spodoptera frugiperda*), proporcionando maior eficiência no desempenho dos classificadores.

Quanto aos resultados obtidos na classificação dos estágios de desenvolvimento da Lagarta do Cartucho (*Spodoptera frugiperda*) no cenário proposto para o desenvolvimento do método, foi possível concluir que para a classificação do estágio 1, o classificador SVM com função *kernel* sigmoide com proporção de 50% de treinamento e teste foi o mais preciso, com taxa de precisão de 78%. Para a classificação do estágio 2, concluiu-se que o classificador SVM com função *kernel* sigmoide e proporção de 70% para treinamento e 30% para teste obteve os melhores resultados, mensurando precisão e acurácia. Para a classificação do estágio 3, concluiu-se que o classificador SVM com função *kernel* sigmoide e também proporção de 50% para treinamento e teste obteve o melhor desempenho, mensurando taxa de precisão e de acurácia, com taxas de 80% de acurácia e precisão, respectivamente. Para a classificação do estágio 4 concluiu-se que o classificador SVM com função *kernel* sigmoide e proporção de 80% de treinamento e 20% de teste obteve os melhores resultados, com taxa de acurácia e precisão de 83%. Finalmente, para estágio 5, concluiu-se que o classificador SVM com função *kernel* sigmoide e proporção de 50% por 50%, obteve os melhores resultados com taxa de acurácia de 79% e taxa de precisão de 81%.

Por fim, foi possível estabelecer o método para a classificação de padrões da Lagarta do Cartucho (*Spodoptera frugiperda*) na cultura do milho baseado em processamento de imagens digitais e aprendizado de máquina.

4.2 Dificuldades e limitações encontradas

A principal dificuldade encontrada foi a limitação quanto à aquisição de imagens da Lagarta do Cartucho em campo na cultura do milho.

Também, a implementação da etapa de segmentação de imagens apresentou grande dificuldade, dada a complexidade das imagens, uma vez que a lagarta poderia ser encontrada em diferentes locais da planta do milho, o que demandou o desenvolvimento de técnica de segmentação adaptativa ao local, em folhas ou em espiga do milho.

Por fim, dada a complexidade da formação das imagens e também da natureza dos dados gerados para a criação do vetor de características, o desenvolvimento do conjunto de classificadores mostrou-se desafiador, uma vez que foram realizados diversos testes com um único classificador, o qual se mostrou ineficaz para a classificação dos diferentes estágios de desenvolvimento da Lagarta do Cartucho.

4.3 Sugestões para Trabalhos Futuros

- Evoluir o método de para classificação de padrões da Lagarta do Cartucho para operação e portabilidade em modo *'onFarm'*;
- Evoluir o método para a utilização de classificadores não supervisionados.

REFERÊNCIAS

ABDELGHAFOR, F.; ROSU, R.; KERESZTES, B.; GERMAIN, C.; COSTA, J. P. da. A Bayesian framework for joint structure and colour based pixel-wise classification of grapevine proximal images. *Computers and Electronics in Agriculture*, Elsevier, v. 158, n. December 2018, p. 345–357, 2019. ISSN 01681699. Disponível em: <<https://doi.org/10.1016/j.compag.2019.02.017>>. Citado na página 36.

ABDURRAZZAQ, A.; JUNOH, A. K.; MUHAMAD, W. Z. A. W.; YAHYA, Z.; MOHD, I. An overview of multi-filters for eliminating impulse noise for digital images. *Telkomnika (Telecommunication Computing Electronics and Control)*, v. 18, n. 1, p. 385–393, 2020. ISSN 23029293. Citado na página 46.

ABIMILHO. *Associação Brasileira das Indústrias do Milho*. 2021. Disponível em: <<http://www.abimilho.com.br/estatisticas>>. Citado na página 23.

AHMED, I.; AHMAD, Z.; ISLAM, M.; ADNAN, A. A real-time specific weed recognition system by measuring weeds density through mask operation. *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*, p. 221–225, 2008. Citado na página 33.

ARAÚJO, A. F.; TAVARES, J. M. R. S.; OLIVEIRA, R. B.; ROSSETTI, R. B.; MARRANGHELLO, N.; PEREIRA, A. S. Análise e caracterização de lesões de pele para auxílio ao diagnóstico médico. In: *Avanços em Visão Computacional*. [S.l.: s.n.], 2012. p. 27–46. ISBN 9788564619098. Citado na página 28.

AUGUSTO, K. S. *Identificação automática do grau de maturação de pelotas de minério de ferro*. 183 p. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro., 2012. Disponível em: <https://www.maxwell.vrac.puc-rio.br/21365/21365{_}6>. Citado na página 65.

BANSAL, S.; AGGARWAL, D. Color Image Segmentation using CIELab Color Space using Ant Colony Optimization. *International Journal of Computer Applications*, v. 29, n. 9, p. 28–34, 2011. Citado na página 54.

BAXES, G. A. *Digital image processing: principles and applications*. [S.l.]: Wiley New York, 1994. Citado na página 56.

BEN; SAID, A.; HADJIDJ, R.; MELKEMI, K. E.; FOUFOU, S. Multispectral image denoising with optimized vector non-local mean filter. *Digital Signal Processing: A Review Journal*, Elsevier Inc., v. 58, p. 115–126, 2016. ISSN 10512004. Disponível em: <<http://dx.doi.org/10.1016/j.dsp.2016.07.017>>. Citado na página 48.

BERGLAND, G. D. A guided tour of the fast Fourier Transform. *IEEE Spectrum*, v. 6, n. 7, p. 41–52, 1969. ISSN 00189235. Citado na página 35.

- BERTOLLA, A. B.; CRUVINEL, P. E. Band-Pass Filtering for Non-Stationary Noise in Agricultural Images to Pest Control Based on Adaptive Semantic Modeling. *Proceedings - 2021 IEEE 15th International Conference on Semantic Computing, ICSC 2021*, p. 398–403, 2021. Citado 2 vezes nas páginas 31 e 46.
- BEYER, K.; GOLDSTEIN, J.; RAMAKRISHNAN, R.; SHAFT, U. When is “nearest neighbor” meaningful? *International conference on database theory*, v. 1540, n. 3, p. 217–235, 1999. ISSN 03029743. Disponível em: <<http://www.mendeley.com/research/lecture-notes-computer-science-2/>>. Citado na página 66.
- BORTOLETTO, A. M.; ALCARDE, A. R. Dominante nos EUA, etanol de milho é opção, no Brasil, para safra excedente. In: *Revista visão agrícola*. [S.l.: s.n.], 2015. p. 135–137. Citado na página 24.
- BRITO, A. R. de. *Método para classificação de sementes agrícolas em imagens obtidas por tomografia de raios-X em alta resolução*. Tese (Doutorado) — Universidade Federal de São Carlos, 2020. Citado 2 vezes nas páginas 48 e 94.
- BRUNA, J. *Não cometa erros no manejo: 5 métodos de controle da lagarta-do-cartucho*. 2018. Disponível em: <<https://blog.aegro.com.br/lagarta-do-cartucho/>>. Citado 2 vezes nas páginas 10 e 26.
- BUADES, A.; COLL, B.; MOREL, J. M. A non-local algorithm for image denoising. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, II, n. 0, p. 60–65, 2005. Citado 3 vezes nas páginas 30, 47 e 48.
- BURGES, C. J. C. A Tutorial on support vector machines for pattern recognition. In: *Data mining and knowledge discovery*. [S.l.]: Kluwer Academic Publishers, 1998. v. 2, n. 2, p. 121–167. Citado 3 vezes nas páginas 66, 69 e 70.
- CAMPBELL, C. An introduction to kernel methods. In: *Studies in Fuzziness and Soft Computing*. [S.l.]: PHYSICA-VERLAG, 2001. cap. 7, p. 155–192. Citado na página 67.
- CANNY, J. A Computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, 1986. Citado na página 56.
- CASELA, C. R.; MENDES, S. M.; VIANA, P. A.; COTA, L. V.; DÉCIO, K.; Da Costa, R. V.; CRUZ, I. *Árvore do conhecimento do milho: pragas e doenças*. 2011. Citado na página 25.
- CHU, H.; ZHANG, D.; SHAO, Y.; CHANG, Z.; GUO, Y.; ZHANG, N. Using hog descriptors and uav for crop pest monitoring. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. p. 1516–1519. ISBN 9781728113128. Citado na página 60.
- CLARO, M.; VOGADO, L.; SANTOS, J.; VERAS, R. Utilização de Técnicas de Data Augmentation em Imagens: Teoria e Prática. In: *Minicursos da ERCEMAPI 2020*. [S.l.: s.n.], 2020. p. 47–71. Citado 2 vezes nas páginas 43 e 44.
- CORTES, C.; VAPNIK, V. Support-Vector Networks. *Machine learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 08859000. Citado 3 vezes nas páginas 66, 68 e 69.
- CRUVINEL, P. E.; VIEIRA, S. R.; CRESTANA, S.; MINATEL, E. R.; MUCHERONI, M. L.; NETO, A. T. Image processing in automated measurements of raindrop size and distribution. *Computers and Electronics in Agriculture*, v. 23, n. 3, p. 205–217, 1999. ISSN 01681699. Citado na página 29.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: . [S.l.: s.n.], 2005. I, p. 886–893. ISBN 0769523722. Citado na página 60.

DELGADO, R. C.; SEDIYAMA, G. C.; COSTA, M. H.; SOARES, V. P.; ANDRADE, R. G. Classificação espectral de área plantada com a cultura da cana-de-açúcar por meio da árvore de decisão. *Engenharia Agrícola*, v. 32, n. 2, p. 369–380, 2012. ISSN 01006916. Citado na página 28.

DEY, S. *Hands-on: Image processing with python*. [S.l.]: Packt Publishing Ltd, 2018. ISBN 9781789343731. Citado 3 vezes nas páginas 10, 26 e 27.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification*. [S.l.]: John Wiley & Sons, 2012. Citado na página 65.

DURMUS, D. CIELAB Color space boundaries under theoretical spectra and 99 test color samples. *Color Research and Application*, v. 45, n. 5, p. 796–802, 2020. ISSN 15206378. Citado na página 54.

EVANGELISTA, I. R. S. Bayesian wingbeat frequency classification and monitoring of flying insects using wireless sensor networks. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, IEEE, v. 2018-October, n. October, p. 2403–2407, 2019. ISSN 21593450. Citado na página 34.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. Inteligência Artificial: Uma abordagem de aprendizado de máquina. 2011. Citado 8 vezes nas páginas 63, 65, 67, 68, 69, 71, 72 e 76.

FAWZI, A.; SAMULOWITZ, H.; TURAGA, D.; FROSSARD, P. Adaptive data augmentation for image classification. *Proceedings - International Conference on Image Processing, ICIP*, IEEE, v. 2016-Augus, p. 3688–3692, 2016. ISSN 15224880. Citado na página 44.

FERNANDES, M. P.; SATO, J. R.; Busatto Filho, G.; THOMAZ, C. E. Classificação estatística e predição da doença de Alzheimer por meio de imagens médicas do encéfalo humano. In: NEVES, L. A. P.; NETO, H. V.; GONZAGA, A. (Ed.). *Avanços em Visão Computacional*. Curitiba: omnipax, 2012. cap. 1, p. 1–26. ISBN 9788564619098. Citado na página 28.

FERNÁNDEZ, C. C. C. *Novos algoritmos de aprendizado para classificação de padrões utilizando floresta de caminhos ótimos*. Tese (Doutorado) — Universidade Estadual de Campinas, 2011. Citado na página 65.

FUENTES, A.; YOON, S.; KIM, S. C.; PARK, D. S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors (Switzerland)*, v. 17, n. 9, 2017. ISSN 14248220. Citado na página 34.

GONZALEZ, R. C.; WOODS, R. C. *Processamento digital de imagens*. [S.l.]: Pearson Educación, 2009. Citado 4 vezes nas páginas 10, 47, 56 e 66.

GONZALEZ, R. C.; WOODS, R. E. *No Title Processamento de imagens digitais*. 3ª. ed. [S.l.: s.n.], 2010. ISBN 978-85-7605-401-6. Citado 3 vezes nas páginas 57, 58 e 60.

GRAMA, L.; TUNS, L.; RUSU, C. On the optimization of SVM kernel parameters for improving audio classification accuracy. *2017 14th International Conference on Engineering of Modern Electric Systems, EMES 2017*, IEEE, p. 224–227, 2017. Citado na página 35.

GUERRERO, J. M.; PAJARES, G.; MONTALVO, M.; ROMEO, J.; GUIJARRO, M. Support Vector Machines for crop/weeds identification in maize fields. *Expert Systems with Applications*, Elsevier Ltd, v. 39, n. 12, p. 11149–11155, 2012. ISSN 09574174. Disponível em: <<http://dx.doi.org/10.1016/j.eswa.2012.03.040>>. Citado na página 34.

HARALICK, R. M.; DINSTEN, I.; SHANMUGAM, K. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3, n. 6, p. 610–621, 1973. ISSN 21682909. Citado 3 vezes nas páginas 32, 33 e 36.

HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR, 1994. Citado na página 66.

HE, Q.; MA, B.; QU, D.; ZHANG, Q.; HOU, X.; ZHAO, J. Cotton pests and diseases detection based on image processing. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, v. 11, n. 6, p. 3445–3450, 2013. ISSN 2302-4046. Citado na página 31.

HEARST, M. A.; DUMAIS, S. T.; OSUNA, E.; PLATT, J.; SCHOLKOPF, B. Support vector machines. *IEEE Intelligent Systems and their applications*, v. 13, n. 4, p. 18–28, 1998. Citado na página 71.

HERBRICH, R. *Learning kernel classifiers: theory and algorithms*. [S.l.]: MIT press, 2001. Citado na página 72.

HONGYU, K.; SANDANIELO, V. L. M.; JUNIOR, G. J. d. O. Análise de Componentes Principais: Resumo Teórico, Aplicação e Interpretação. *ES Engineering and Science*, v. 5, n. 1, p. 83–90, 2016. ISSN 2358-5390. Citado na página 63.

HOTELLING, H. Analysis of a complex of statistical variables into Principal Components. *The Journal of Educational Psychology*, v. 24, p. 417–441, 1933. Citado na página 63.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. A practical guide to support vector classification. 2003. Citado na página 72.

HU, M.-K. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, v. 8, n. 2, p. 179–187, 1962. Disponível em: <<https://ieeexplore-ieee-org.biblioteca5.uc3m.es/ielx5/4547527/22787/01057692.pdf?tp={&}arnumber=1057692{&}isnumber=22787{&}ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2RvY3VtZW50LzE>>. Citado 2 vezes nas páginas 60 e 61.

HUANG, K. Y. Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. *Computers and Electronics in Agriculture*, 2007. ISSN 01681699. Citado na página 32.

IBRAHEEM, N. A.; HASAN, M. M.; KHAN, R. Z.; MISHRA, P. K. Understanding color models: A review. *ARNP Journal of Science and Technology*, v. 2, n. 3, p. 265–275, 2012. ISSN 2225-7217. Disponível em: <<http://www.ejournalofscience.org>>. Citado 2 vezes nas páginas 10 e 49.

JAHNE, B. *Practical handbook on image processing for scientific and technical applications*. [S.l.]: CRC Press, 2004. Citado na página 26.

JAIN, P.; TYAGI, V. Spatial and Frequency Domain Filters for Restoration of Noisy Images. *IETE Journal of Education*, v. 54, n. 2, p. 108–116, 2013. ISSN 0974-7338. Citado na página 46.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. [S.l.: s.n.], 1995. p. 338–345. ISBN 1-55860-385-9. Citado 2 vezes nas páginas 34 e 36.

KAMDI, S.; KRISHNA, R. K. Image Segmentation and Region Growing Algorithm. *International Journal of Computer Technology and Electronics Engineering*, v. 2, n. 1, p. 103–107, 2012. ISSN 2249-6343. Citado na página 56.

KANG, S. H.; KIM, J. Y. Application of fast non-local means algorithm for noise reduction using separable color channels in light microscopy images. *International Journal of Environmental Research and Public Health*, v. 18, n. 6, p. 1–12, 2021. ISSN 16604601. Citado na página 30.

KAUR, A.; KRANTHI, B. V. Comparison between YCbCr Color Space and CIELab Color Space for Skin Color Segmentation. *International Journal of Applied Information Systems (IJ AIS)*, v. 3, n. 4, p. 30–33, 2012. Citado na página 55.

KEERTHI, S. S.; LIN, C.-J. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, v. 15, n. 7, p. 1667–1689, 2003. Citado na página 72.

KIRSCH, R. A. SEAC and the start of image processing at the National Bureau of Standards. *IEEE Annals of the History of Computing*, v. 20, n. 2, p. 7–13, 1998. ISSN 10586180. Citado na página 39.

KIRSCH, R. A.; CAHN, L.; RAY, C.; URBAN, G. H. Experiments in processing pictorial information with a digital computer. *Proceedings of the Eastern Joint Computer Conference: Computers with Deadlines to Meet, IRE-AIEE-ACM 1958*, p. 221–229, 1957. Citado na página 39.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*, p. 1097–1105, 2012. Citado 2 vezes nas páginas 34 e 36.

KUHN, H. W.; TUCKER, A. W. Nonlinear programming. In: *Traces and Emergence of Nonlinear Programming*. [S.l.]: Birkhäuser, Basel, 2014. p. 247–248. ISBN 9783034804394. Citado na página 68.

KULKARNI, N. Color thresholding method for image segmentation of natural images. *International Journal of Image, Graphics and Signal Processing*, v. 4, n. 1, p. 28–34, 2012. ISSN 20749074. Citado na página 56.

KUMAR, Y.; DUBEY, A. K.; JOTHI, A. Pest detection using adaptative thresholding. In: *International Conference on Computing, Communication and Automation (ICCCA2017)*. [S.l.: s.n.], 2017. p. 42–46. Citado na página 32.

LEE, S. H.; CHAN, C. S.; WILKIN, P.; REMAGNINO, P. Deep-plant: Plant identification with convolutional neural networks. In: *International Conference on Image Processing, ICIP*. [S.l.]: IEEE, 2015. p. 452–456. ISBN 9781479983391. ISSN 15224880. Citado na página 34.

LI, X.; DAI, B.; SUN, H.; LI, W. Corn classification system based on computer vision. *Symmetry*, v. 11, n. 4, 2019. ISSN 20738994. Citado na página 36.

LIN, H.-T.; LIN, C.-J. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. *Neural Computation*, v. 3, n. 2, p. 1–32, 2003. Disponível em: <<http://home.caltech.edu/~jhtlin/publication/doc/tanh.p>>. Citado na página 72.

LIU, X.; ZHANG, R.; MENG, Z.; HONG, R.; LIU, G. On fusing the latent deep CNN feature for image classification. *World Wide Web*, World Wide Web, v. 22, n. 2, p. 423–436, 2019. ISSN 15731413. Citado na página 35.

LORENA, A. C.; CARVALHO, A. C. P. L. F. D. Uma introdução às support vector machines. *RITA*, v. 14, n. 2, 2007. Citado 5 vezes nas páginas 19, 67, 69, 70 e 72.

LUO, M. R.; HUNT, R. W. G. Inconstancy Index. *COLOR research and application*, 1997. Citado na página 32.

MACQUEEN, J. Some methods for classification and analysis fo multivariate observations. In: *fifth Berkeley symposium on mathematical statistics and probability*. [S.l.: s.n.], 1967. v. 1, n. 14, p. 281–297. ISSN 15441024. Citado na página 32.

MAHDI, A.; QIN, J. Line profile based segmentation algorithm for touching corn kernels. 2017. Disponível em: <<http://arxiv.org/abs/1706.00396>>. Citado na página 36.

MATOS, G.; MENDONÇA, M.; FREIRE, E.; MONTALVÃO, J.; MATOS, L. Sistema de visão artificial baseado em detecção de cores (para sistema de controle de robos celulares com realimentação visual). *VIII Simpósio Brasileiro de Automação Inteligente SBAI*, 2007. Citado 2 vezes nas páginas 10 e 50.

MENKE, A. B.; JUNIOR, O. A. d. C.; GOMES, R. A. T.; MARTINS, É. d. S.; OLIVEIRA, S. N. de. Análise de mudanças do uso agrícola da terra a partir de dados de sensoriamento remoto multitemporal no município de Luis Eduardo Magalhães (BA-Brasil). *Revista Brasileira de Ensino de Física*, v. 39, n. 1, p. 315–326, 2017. ISSN 01024744. Citado na página 28.

MIKOŁAJCZYK, A.; GROCHOWSKI, M. Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop, IIPhDW 2018*, IEEE, p. 117–122, 2018. Citado 2 vezes nas páginas 29 e 42.

MIRANDA, J. L.; GERARDO, B. D.; Tanguilig III, B. T. Pest detection and extraction using Image processing techniques. *International Journal of Computer and Communication Engineering*, v. 3, n. 3, p. 189–192, 2014. ISSN 20103743. Citado na página 28.

MISHRA, R.; MITTAL, N.; KHATRI, S. K. Digital Image Restoration using Image Filtering Techniques. In: *International Conference on Automation, Computational and Technology Management*. [S.l.]: IEEE, 2019. p. 268–272. ISBN 9781538680100. Citado na página 30.

MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997. 432 p. ISBN 0070428077. Citado 2 vezes nas páginas 65 e 66.

MOGHADDAM, R. F.; CHEN, S.; HEDJAM, R.; CHERIET, M. A maximal-information color to gray conversion method for document images: Toward an optimal grayscale representation for document image binarization. n. March 2014, 2013. Disponível em: <<http://arxiv.org/abs/1306.6058>>. Citado 2 vezes nas páginas 10 e 54.

MYTHILI, C.; KAVITHA, V. Efficient Technique for Color Image Noise Reduction. *The Research Bulletin of Jordan ACM*, v. 2, n. 3, p. 41–44, 2011. Disponível em: <<https://pdfs.semanticscholar.org/1afd/ddab22b25ae75bb05f3599d0921d058de7ff.pdf>>. Citado 2 vezes nas páginas 30 e 46.

- NALINI, N. J.; PALANIVEL, S. Music emotion recognition: The combined evidence of mfcc and residual phase. *Egyptian Informatics Journal*, Elsevier B.V., v. 17, p. 1–10, 3 2016. ISSN 11108665. Citado 2 vezes nas páginas 10 e 71.
- NANDA, M. A.; SEMINAR, K. B.; NANDIKA, D.; MADDU, A. A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information (Switzerland)*, v. 9, n. 1, 2018. ISSN 20782489. Citado na página 35.
- NOYUM, V. D.; MOFENJOU, Y. P.; FEUDJIO, C.; GÖKTUG, A.; FOKOUÉ, E. Boosting the predictive accuracy of singer identification using discrete wavelet transform for feature extraction. 1 2021. Disponível em: <<http://arxiv.org/abs/2102.00550>>. Citado 2 vezes nas páginas 10 e 67.
- OLIVEIRA, L. A.; GONZAGA, A. Localização, segmentação e reconhecimento de caracteres em placas de automóveis. In: *Avanços em Visão Computacional*. [S.l.: s.n.], 2012. p. 283–302. ISBN 9788564619098. Citado na página 28.
- OTSU, N. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. Citado 3 vezes nas páginas 32, 36 e 56.
- OYAMA, P. I. d. C.; JORGE, L. A. d. C.; RODRIGUES, E. L. L.; GOMES, C. C. Sistema para classificação automática de café em grãos por cor e forma através de imagens digitais. In: *Avanços em Visão Computacional*. [S.l.: s.n.], 2012. p. 119–140. ISBN 9788564619098. Citado na página 28.
- PEARSON, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901. Citado na página 63.
- PECHA, M.; HORÁK, D. Analyzing l1-loss and l2-loss support vector machines implemented in permon toolbox. In: . [S.l.]: Springer Verlag, 2020. v. 554, p. 13–23. ISBN 9783030149062. ISSN 18761119. Citado 2 vezes nas páginas 10 e 70.
- REGITANO-D'ARCE, M. A. B.; SPOTO, M. H. F.; CASTELLUCCI, A. C. L. Processamento e industrialização do milho para alimentação humana. In: *Revista visão agrícola*. [s.n.], 2015. p. 138–140. Disponível em: <<http://www.agencia.cnptia.>> Citado na página 23.
- RIBEIRO, M.; GOMES, A. Adaptação de Cor para Dicromatas na Visualização de Imagens. In: *Encontro Português de Computação Gráfica*. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 10 e 50.
- RIVEST, R. L. Learning decision lists. *Machine learning*, v. 2, n. 3, p. 229–246, 1987. Citado 2 vezes nas páginas 36 e 66.
- ROSA, A. P. S. A. D.; BARCELOS, H. T. Bioecologia e controle de Spodoptera frugiperda em milho. *Embrapa Clima Temperado. Documentos*, 344, p. 30, 2012. Citado na página 25.
- SÁ, A. d. O. *Detecção de agrupamento de microcalcificações em imagens de mamogramas digitalizados usando a transformada wavelet complexa de árvore dupla*. 148 p. Tese (Doutorado) — Universidade Federal de São Carlos, 2016. Citado na página 66.
- SANGWINE, S. J.; HORNE, R. E. *The colour image processing handbook*. London: Chapman & Hall, 1998. ISBN 0412806207. Citado na página 54.

- SANKARAN, S.; EHSANI, R.; ETXEBERRIA, E. Mid-infrared spectroscopy for detection of Huanglongbing (greening) in citrus leaves. *Talanta*, 2010. ISSN 00399140. Citado na página 28.
- SARAVANAN, G.; YAMUNA, G.; NANDHINI, S. Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. *International Conference on Communication and Signal Processing, ICCSP 2016*, IEEE, p. 462–466, 2016. Citado na página 49.
- SETTE, P. G. C.; MAILLARD, P. Análise de textura de imagem de alta resolução para aprimorar a acurácia da classificação da mata atlântica no sul da Bahia. In: *XV Simpósio Brasileiro de Sensoriamento Remoto*. [S.l.: s.n.], 2011. p. 2020. Citado na página 33.
- SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, Springer International Publishing, v. 6, n. 1, 2019. ISSN 21961115. Disponível em: <<https://doi.org/10.1186/s40537-019-0197-0>>. Citado na página 42.
- SIEBRA, H. d. A. *Segmentação Fuzzy de imagens coloridas com características texturais : Uma aplicação a Rochas sedimentares*. 69 p. Tese (Doutorado) — Universidade Federal do Rio Grande do Norte, 2013. Citado na página 39.
- SILVA, F. A.; ARTERO, A. O.; PAIVA, M. S. V.; BARBOSA, R. L. Reconhecimento de caracteres baseado em regras de transições entre pixels Vizinhos. In: *Avanços em Visão Computacional*. [S.l.: s.n.], 2012. p. 265–282. ISBN 9788564619098. Citado na página 28.
- SINDMILHO. *Milho e suas riquezas: história*. 2005. Disponível em: <<https://www.fiesp.com.br/sindmilho/sobre-o-sindmilho/curiosidades/milho-e-suas-riquezas-historia/>>. Citado na página 23.
- SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. *Statistics and Computing*, v. 14, n. 3, p. 199–222, 2004. ISSN 09603174. Citado 2 vezes nas páginas 68 e 69.
- SOLOMON, C.; BRECKON, T. *Fundamentos de processamento digital de imagens: uma abordagem prática com exemplos em Matlab*. [S.l.]: Grupo Gen-LTC, 2000. Citado 6 vezes nas páginas 19, 39, 41, 43, 47 e 60.
- SPACKMAN, K. A. Signal detection theory: Valuable tools for evaluating inductive learning. In: ELSEVIER. *Proceedings of the sixth international workshop on Machine learning*. [S.l.], 1989. p. 160–163. Citado na página 75.
- SRIWASTWA, A.; PRAKASH, S.; MRINALINI; SWARIT, S.; KUMARI, K.; SAHU, S. S. Detection of Pests Using Color Based Image Segmentation. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, IEEE, n. Icticct, p. 1393–1396, 2018. Citado na página 32.
- STRAZZI, S. Derivados do milho são usados em mais de 150 diferentes produtos industriais. In: *Visão agrícola*. [S.l.: s.n.], 2015. v. 13, p. 146–150. Citado 3 vezes nas páginas 19, 23 e 24.
- SWETS, D. L.; WENG, J. J. Using Discriminant Eigenfeatures for Image Retrieval. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, v. 18, n. 8, p. 831–836, 1996. Citado na página 94.
- VARELLA, C. A. A. Análise de componentes principais. *Seropédica: Universidade Federal Rural do Rio de Janeiro*, p. 38, 2008. Citado na página 64.

- VIANA, P. A.; CRUZ, I.; WAQUIL, J. M. *Árvore do conhecimento do milho: pragas da fase inicial*. 2011. Disponível em: <http://www.agencia.cnptia.embrapa.br/gestor/milho/arvore/CONTAG01{_}68{_}1682005112>. Citado na página 25.
- VICENT, L.; SOILLE, P. Watersheds on digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on pattern analysis and Machine Intelligence*, v. 13, n. 6, p. 583–598, 1991. Citado na página 56.
- VICKERS, A. L.; MODESTINO, J. W. A maximum likelihood approach to texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, n. 1, p. 61–68, 1982. Citado 2 vezes nas páginas 33 e 36.
- VISA, S.; RAMSAY, B.; RALESCU, A. L.; KNAAP, E. V. D. Confusion matrix-based feature selection. *MAICS*, v. 710, p. 120–127, 2011. Citado na página 75.
- WU, D.; SUN, D. W. Colour measurements by computer vision for food quality control - A review. *Trends in Food Science and Technology*, v. 29, n. 1, p. 5–20, 2013. ISSN 09242244. Citado na página 28.
- XAVIER, A. d. C.; SATO, J. R.; GIRALDI, G. A.; RODRIGUES, P. S.; THOMAZ, C. E. Classificação e extração de características discriminantes de imagens 2D de ultrassonografia mamária. In: NEVES, L. A. P.; NETO, H. V.; GONZAGA, A. (Ed.). *Avanços em Visão Computacional*. Curitiba: omnipax, 2012. cap. 4, p. 65–87. ISBN 9788564619098. Citado na página 28.
- XIA, C.; CHON, T. S.; REN, Z.; LEE, J. M. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. *Ecological Informatics*, Elsevier B.V., v. 29, n. P2, p. 139–146, 2015. ISSN 15749541. Disponível em: <<http://dx.doi.org/10.1016/j.ecoinf.2014.09.006>>. Citado na página 31.
- ZHAO, W.; WANG, J. Study of feature extraction based visual invariance and species identification of weed seeds. In: . [S.l.: s.n.], 2010. v. 2, p. 631–635. ISBN 9781424459612. Citado na página 61.