Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja

Alessandro dos Santos Ferreira

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE COMPUTAÇÃO - FACOM

Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja

Alessandro dos Santos Ferreira

Orientador: *Prof. Dr. Hemerson Pistori*

Dissertação apresentada ao curso de Pós Graduação em Ciências da Computação, da Universidade Federal de Mato Grosso do Sul, como requisito parcial à obtenção do título de Mestre em Ciências da Computação.

UFMS - Campo Grande
março/2017

Agradecimentos

A todos os amigos do grupo INOVISÃO, em especial ao meu Orientador Hemerson Pistori e aos colegas do grupo VANTAGRO, Gercina Gonçalves e Diogo Soares.

Aos amigos da Tendência Informações e Sistemas, empresa que possibilitou que eu realizasse o mestrado concorrentemente às minhas atividades na mesma.

A toda minha família, em especial ao meu pai, José de Oliveira Ferreira.

Este trabalho recebeu apoio financeiro da Universidade Católica Dom Bosco, UCDB e da Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia do Estado de Mato Grosso do Sul, FUNDECT. Também contou com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPQ e da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES.

Abstract

Weeds are undesirable plants that grow in agricultural crops, such as soybean crops, competing for elements such as sunlight and water, causing losses to crop yields. The objective of this work was to use Convolutional Neural Networks (ConvNets or CNNs) to perform weed detection in soybean crop images and classify these weeds among grass and broadleaf, aiming to apply the specific herbicide to weed detected. For this purpose, a soybean plantation was carried out in Campo Grande, Mato Grosso do Sul, Brazil, and the Phantom DJI 3 Professional drone was used to capture a large number of crop images. With these photographs, an image database was created containing over fifteen thousand images of the soil, soybean, broadleaf and grass weeds. The Convolutional Neural Networks used in this work represent a Deep Learning architecture that has achieved remarkable success in image recognition. For the training of Neural Network the CaffeNet architecture was used. Available in Caffe software, it consists of a replication of the well known AlexNet, network which won the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012). A software was also developed, Pynovisão, which through the use of the superpixel segmentation algorithm SLIC, was used to build a robust image dataset and classify images using the model trained by Caffe software. In order to compare the results of ConvNets, Support Vector Machines, AdaBoost and Random Forests were used in conjunction with a collection of shape, color and texture feature extraction techniques. As a result, this work achieved above 98% accuracy using ConvNets in the detection of broadleaf and grass weeds in relation to soil and soybean, with an accuracy average between all images above 99%.

Resumo

Ervas daninhas são plantas indesejadas que crescem em culturas agrícolas, como as de soja, competindo por diversos fatores como luz e água e causando prejuízos às lavouras. O objetivo deste trabalho foi utilizar Redes Neurais Convolucionais para realizar a detecção de ervas daninhas em imagens de lavouras de soja e classificar essas ervas daninhas entre gramíneas e folhas largas, visando direcionar o herbicida específico ao tipo de erva daninha detectado. Para esse objetivo foi realizada uma plantação de soja em Campo Grande, Mato Grosso do Sul, Brasil e com o uso do drone Phantom DJI 3 Professional foi capturado um grande número de imagens da cultura. Com essas fotografias foi construído um banco de imagens contendo mais de quinze mil imagens do solo, soja e ervas daninhas de folhas largas e gramíneas. As Redes Neurais Convolucionais utilizadas representam uma arquitetura de Aprendizado Profundo que vêm alcançando notável destaque no reconhecimento de imagens. Para o treinamento da Rede Neural foi utilizada a arquitetura CaffeNet, disponível no software Caffe, que consiste de uma replicação da conhecida rede AlexNet, que venceu a competição ImageNet LSRVC de 2012. Foi implementado também um software, Pynovisão, que através do uso do segmentador SLIC Superpixel, ajudou na construção de um banco de imagens robusto e na classificação das imagens utilizando o modelo treinado pelo software Caffe. Para comparar os resultados da Rede Neural Convolucional, foram utilizados os algoritmos Máquina de Vetores de Suporte, AdaBoost e Florestas Aleatórias em conjunto com uma coleção de extratores de atributos de forma, cor e textura. Como resultado, utilizando as Redes Neurais Convolucionais, este trabalho obteve precisão acima de 98% na detecção de ervas daninhas de folhas largas e gramíneas em relação ao solo e a soja, com média de precisão entre todas as imagens superior a 99%.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO

1

Introdução

A soja representa o papel de principal oleaginosa consumida mundialmente, tanto para o consumo animal através do farelo de soja quanto para consumo humano através do óleo [1]. No Brasil a sua relevância para o agronegócio começou a partir dos anos 70 através do aumento da área cultivada e principalmente pelo aumento da produtividade obtido através do uso de novas tecnologias. A partir dos anos 90, a agricultura brasileira passou por um processo de modernização fazendo com que o setor de soja alcançasse maior crescimento e dinamismo. A importância do complexo de soja para o Brasil pode ser observada tanto pelo grande crescimento de produção quanto pela arrecadação com a sua exportação e de seus derivados como óleo e farelo de soja [1].

De acordo com o portal do Ministério da Agricultura, a soja é a cultura agrícola que mais cresceu nas últimas três décadas e corresponde a 49% da área plantada com grãos no país. Cultivada principalmente nas regiões Centro-Oeste e Sul do país, a soja é um dos produtos mais destacados da agricultura nacional e na balança comercial, sendo o principal gerador de divisas cambiais do Brasil, com valores anuais que ultrapassam os 20 bilhões de dólares. A nível mundial, segundo dados de 2015, foram produzidas mais de 320 milhões de toneladas de soja. O Brasil se encontra na segunda posição, atrás apenas dos Estados Unidos, sendo responsável por aproximadamente 31% da produção mundial, o que correspondeu a 100 milhões de toneladas. Dada a importância da soja no contexto econômico é imprescindível o uso de técnicas visando maximizar a produtividade e qualidade do produto.

Ervas daninhas são plantas que nascem espontaneamente em lavouras,

como as de soja, competindo por luz, água e nutrientes, causandos prejuízos à colheita. Rizzardi e Fleck [2] citam que o conhecimento da infestação é um procedimento fundamental para a utilização de medidas preventivas no controle das ervas daninhas. Eles focam a importância de dispôr-se de métodos que realizem a quantificação e análise da distribuição da infestação de ervas daninhas de forma rápida e econômica, evidenciando a necessidade de um método mais prático que a realização de observações sistemáticas das lavouras.

Figura 1.1: Ervas daninhas na lavoura de soja. À esquerda, ervas daninhas de folhas largas e à direita, gramíneas.

As ervas daninhas podem ser classificadas com base no formato das suas folhas. As suas folhas têm uma infinidade de formatos mas as gramíneas, de folhas estreitas e longas, se distinguem claramente, permitindo assim classificar todas as outras como ervas daninhas de folhas largas, como pode ser visto na Figura 1.1. A separação nessas duas classes é adequada porque gramíneas e ervas daninhas de folhas largas são diferenciadas no tratamento devido à seletividade de alguns herbicidas ao grupo específico [3]. A aplicação de herbicidas consegue melhores resultados se o tratamento for direcionado ao tipo específico de ervas daninhas.

Neste trabalho nosso objetivo foi utilizar Redes Neurais Convolucionais Profundas ou *ConvNets* para realizar a identificação das ervas daninhas em relação à soja e ao solo e classificação delas em gramíneas e folhas largas. O Aprendizado Profundo (*Deep Learning*), em especial a arquitetura de redes convolucionais, representa atualmente o estado da arte no reconhecimento de imagens e objetos [4]. Classificadores utilizados para esse fim como máquinas de vetores de suporte e redes neurais tradicionais são dependentes de bons algoritmos extratores de atributos como FFT, SIFT e *Gabor wavelet* [5]. O fator chave do aprendizado profundo é que sua extração de características é aprendida automaticamente a partir do dado bruto [4]. Suas camadas de atributos não são modeladas manualmente, não sendo necessária a utilização de algoritmos extratores para a realização do

treinamento e classificação da rede convolucional. Assim, é esperado que esta técnica tenha muito mais sucesso num futuro próximo pois necessita de pouca engenharia manual, podendo se beneficiar do aumento de capacidade de computação e dados [4].

Para realizar a detecção automática de ervas daninhas em lavouras de soja, foi realizada uma plantação experimental de soja em Campo Grande, Mato Grosso do Sul. Imagens aéreas dessa lavoura foram coletadas por Veículos Aéreos não Tripulados, VANTs, também conhecidos como Aeronaves Remotamente Pilotadas (RPA), que têm se mostrado uma alternativa bastante atraente não apenas economicamente mas também superando outras limitações comuns das imagens obtidas por satélites e aeronaves [6]. Essas imagens foram segmentadas utilizando o algoritmo SLIC [7] possibilitando a criação de um banco com mais de quinze mil segmentos identificando soja, solo e ervas daninhas, separadas por folhas largas e gramíneas. Para efeitos de comparação, além dos testes utilizando redes neurais convolucionais, realizamos testes de detecção com outros classificadores tradicionais como máquinas de vetores de suporte e florestas aleatórias.

Este trabalho encontra-se organizado em sete capítulos. O segundo capítulo é composto pela fundamentação teórica, abordando visão computacional, ervas daninhas e VANTs. Também serão descritas as técnicas de segmentação, extração de atributos e classificação utilizadas neste trabalho para a detecção de ervas daninhas. No capítulo três são listados trabalhos correlatos que abordam problemas similares ao desta pesquisa. O capítulo quatro possui a fundamentação teórica relativa ao Aprendizado Profundo e Redes Neurais Convolucionais. O quinto capítulo é composto pela metodologia do trabalho e o sexto descreve os resultados e discussões. Por fim este trabalho traz a conclusão como último capítulo.

CAPÍTULO

2

Fundamentação Teórica

2.1 Visão Computacional

Prince [8] define que o objetivo da visão computacional é extrair informação útil das imagens. Esta tarefa tem se demonstrado surpreendentemente desafiante, despertando grande interesse dos pesquisadores nas últimas décadas. De acordo com Shapiro e Stockman [9], o seu propósito é tomar decisões úteis sobre objetos físicos e cenas através de imagens. Para tomar decisões sobre objetos reais é quase sempre necessário construir alguma descrição ou modelo deles a partir das imagens. Devido a esse fato, muitos pesquisadores definem o objetivo da visão computacional como a construção da descrição de cenas obtidas das imagens.

Na visão computacional tenta-se descrever o mundo que vemos em imagens e reconstruir suas propriedades como forma, iluminação e distribuição de cor. Como humanos e animais tendem a realizar o reconhecimento de imagens e objetos com facilidade, a dificuldade da resolução do problema computacionalmente pode ser subestimada [10]. De qualquer forma, tem sido alcançado um notável progresso recente em nosso entendimento da visão computacional e na última década se viu uma grande escalada de desenvolvimento de tecnologias relacionadas à área para o consumidor. Um exemplo é que atualmente a maioria das câmeras digitais possuem algoritmos embutidos para reconhecimento de face [8].

A visão computacional é relacionada a inúmeras outras áreas. Muitas técnicas desenvolvidas em outras áreas podem ser utilizadas para recuperar informações das imagens. Jain et al. [11] citam processamento de imagens,

computação gráfica, reconhecimento de padrões e inteligência artificial como áreas que contribuem com técnicas úteis à visão computacional.

Dado o grande crescimento da população mundial faz-se necessário o aumento da produtividade agrícola. Porém uma variedade de malefícios como pragas, doenças e deficiências minerais impactam na produtividade e qualidade das culturas. O monitoramento convencional das plantações é custoso e de baixa eficiência [12]. Para reduzir o trabalho manual e aumentar a eficácia dos tratamentos utilizados, técnicas de processamento de imagem e visão computacional estão sendo utilizadas na agricultura.

Nas seções 2.4, 2.5 e 2.6 serão descritas as técnicas de visão computacional assim como os algoritmos utilizados neste trabalho para realizar a detecção de ervas daninhas.

2.2 Ervas Daninhas e a Soja

Ervas daninhas, também conhecidas como plantas daninhas, invasoras, inços e tingueras, são plantas que crescem espontaneamente em solos agrícolas onde não são desejadas. O crescimento dessas plantas competindo com culturas econômicas, como a soja, causa prejuízos dificultando a operação de máquinas colhedoras e aumentando a impureza e umidade dos grãos [13].

Os efeitos negativos das plantas daninhas nas lavouras incluem a competição de água, luz, nutrientes e espaço, aumento de custos de produção, dificuldade de colheita, depreciação da qualidade do produto, hospedagem de pragas e doenças e diminuição do valor comercial das áreas cultivadas [2]. Para controlar a competição das ervas daninhas são utilizados quatro tipos de manejos: exclusão, prevenção, supressão e erradicação [13].

Através de observações e levantamentos nas regiões produtoras de soja no Brasil, podemos indicar como mais frequentes as seguintes invasoras, classificadas pelo formato das suas folhas [13]:

- gramíneas: capim-custódio (*Pennisetum setosum*), c.-marmelada (*Brachiaria plantaginea*), braquiária (*B. decumbens*), c.-carrapicho (*Cenchrus echinatus*), c.-colchão (*Digitaria spp.*) e trapoeraba (*Commelina benghalensis*).

- folhas largas: carrapicho-rasteiro (*Acanthospermum australe*), picão-preto (*Bidens pilosa*), corda-de-viola (*Ipomoea spp.*), amendoim-bravo (*Euphorbia heterophylla*), caruru (*Amaranthus spp.*), erva-quente (*Spermacoce latifolia*), joá (*Solanum spp.*), falsa-serralha (*Emilia sonchifolia*), guanxuma (*Sida rhombifolia*), poaia-branca

(*Richardia brasilienses*), cheirosa (*Hyptis suaveolens*), mentrasto (*Agetarum conizoides*) e o desmodio (*Desmodium tortuosum*).

No estado do Mato Grosso do Sul, de acordo com dados do projeto SIGA MS, as plantas daninhas de maior incidência na safra 2014/2015 foram a buva (*Conyza bonariensis*), o capim amargoso (*Elionurus candidus*), o carrapicho (*Acanthospermum australe*) e o picão preto (*Bidens pilosa*). Destas, a buva e o capim amargoso tiveram maior incidência nas lavouras da região.

2.2.1 Estádios Fenológicos da Soja

Farias et al. [14] ressaltam a necessidade de uma linguagem unificada na descrição dos estádios fenológicos da soja, uma metodologia que necessita ser objetiva, precisa e universal. A metodologia de descrição mais utilizada no mundo é a proposta por Fehr e Caviness [15] que identifica precisamente o estádio de desenvolvimento de uma planta ou lavoura de soja. Este sistema divide os estádios de desenvolvimento da soja em estádios vegetativos, representados pela letra V, e reprodutivos, representados pela letra R. Com exceção dos estádios VE (emergência) e VC (cotilédone), as letras V e R, são seguidas de índices numéricos que identificam sequencialmente os estádios específicos de desenvolvimento nessas duas fases da planta.

Os dois primeiros estádios de desenvolvimento da planta são, respectivamente, os estádios VE e VC. A partir destes estádios, as subdivisões dos estádios vegetativos são numerados sequencialmente (V1, V2, V3, V4, ..., Vn), onde n é o número de nós acima do nó cotiledonar [14]. Após esses estádios temos os estádios reprodutivos que são denominados pela letra R seguida dos número de um até oito e descrevem detalhadamente o período de florescimento-maturação da soja. Eles abrangem quatro fases distintas do desenvolvimento reprodutivo da planta: florescimento (R1 e R2), desenvolvimento da vagem (R3 e R4), desenvolvimento do grão (R5 e R6) e maturação da planta (R7 e R8).

O período crítico de controle de ervas daninhas é um importante fator em um sistema de combate às ervas daninhas. Ele representa o intervalo durante o ciclo de crescimento da cultura onde as ervas daninhas devem ser controladas para prevenir perdas. Mohammadi e Amiri [16] citam trabalhos que sugerem que as ervas daninhas precisam ser removidas entre os estádios V1 e V3 e entre V3 e V4. Infelizmente não há uma regra trivial para se determinar esse período [17]. Há uma gama de fatores que influenciam esse intervalo como densidade da cultura, condições ambientais, técnicas de produção, característica da cultura e das ervas daninhas.

Van Acker et al. [18] conduziram um estudo para definir o período crítico de controle na cultura de soja em Ontario, Canadá. Eles verificaram que o

Figura 2.1: À esquerda a soja no estádio VE (emergência). À direita, a soja no estádio VC (cotilédone). Ao contrário do estádio VE as bordas das folhas unifolioladas não mais se tocam.

período crítico para remoção das ervas daninhas coincidiram com os estádios V3, V2 e V1 para culturas espaçadas com 19, 38 e 76 centímetros, respectivamente. Todavia eles também concluíram que esse período é extremamente variável por regiões e ao longo dos anos. Como resultado realmente conclusivo eles apontam que caso as ervas daninhas não sejam controladas nos estádios iniciais de desenvolvimento, elas obrigatoriamente precisam ser removidas antes do estágio R1 para evitar perdas na lavoura.

Do ponto de vista da Visão Computacional, é interessante avaliar as mudanças visuais nos estádios vegetativos da soja, para estudarmos a necessidade de classificação por estádios. Para este trabalho, de detecção das ervas daninhas, é relevante avaliar as mudanças durante o estádio vegetativo, estádio onde as ervas daninhas devem obrigatoriamente ser removidas. Cotilédones são as primeiras folhas que surgem quando a semente germina, e cuja função é nutrir a planta no início do crescimento. De acordo com Farias et al. [14], o estádio VE representa a emergência dos cotilédones. Em outras palavras uma planta recém emergida é considerada no estádio VE. Uma planta pode ser considerada emergida quando encontra-se com os cotilédones acima da superfície do solo e os mesmos formam um ângulo igual ou superior a 90 graus com seus respectivos hipocótilos, a parte do caule logo abaixo dos cotilédones.

No estádio VC os cotilédones se encontram completamente abertos e expandidos e as bordas das folhas unifolioladas não mais se tocam. Nesse estádio a soja se apresenta visualmente bastante diferente dos outros estádios pois ainda não possui folhas completamente desenvolvidas e os cotilédones estão completamente abertos, ao contrário do estádio VE, onde os cotilédones estão fechados.

Figura 2.2: Da esquerda para a direita temos imagens de soja nos estádios V1, V6 e R1. Os estádios V1 e V6 se diferenciam pela quantidade de folhas trifolioladas completamente desenvolvidas. No estádio R1 já temos flores abertas.

Considera-se que a soja está no estádio V1 quando todas as suas folhas unifolioladas estão completamente desenvolvidas. Deste estádio em diante a classificação é baseada na quantidade de folhas trifolioladas completamente desenvolvidas. No estádio V2, temos a primeira folha trifoliolada completamente desenvolvida, no estádio V3 a segunda folha trifoliolada e assim sucessivamente. Nos estádios Vn, a soja já se apresenta visualmente de maneira mais uniforme, se distinguindo basicamente na quantidade de folhas completamente desenvolvidas. Além disso nesses estádios se torna mais fácil a identificação da soja em imagens aéreas que nos estádios VE e VC.

2.3 VANTs – Veículos Aéreos Não Tripulados

Os VANTs (Veículos Aéreos Não Tripulados) ou Aeronaves Remotamente Pilotadas (RPA) são aeronaves capazes de serem operadas por controle remoto ou autonomamente. Também são conhecidos como *Unmanned Aerial Vehicles* (UAVs), *Uninhabited Aerial Vehicles* e *Unmanned Aircraft Systems* (UASs). O conceito de construir aeronaves não tripuladas para aplicações diversas surgiu, inicialmente, em virtude de necessidades militares. Após as operações de 1991, com o VANT Pioneer sendo utilizado em mais de 300 missões durante a operação Tempestade no Deserto é que a utilização de VANTs foi impulsionada [19]. Com o avanço tecnológico nos setores de processamento de dados e miniaturização de componentes eletrônicos ocorridos nas últimas duas décadas, diversas aplicações militares de VANTs foram desenvolvidas ao redor do mundo.

O desenvolvimento de pequenos VANTs tem se tornado possível graças à

Figura 2.3: Imagem do VANT DJI Phantom 3 Professional utilizado nas capturas de imagem da safra 2015/2016.

minituarização e redução de custos de componentes eletrônicos como microprocessadores, sensores, baterias e unidades de comunicação wireless. Mais recentemente, usos científicos e civis têm sido desenvolvidos devido ao fato que esses veículos aéreos desprovidos de tripulação podem apresentar vantagens técnicas e econômicas nas mais diversas áreas. Floreano e Wood [20] citam várias atividades civis onde VANTs podem ser potencialmente utilizados como envio de ajuda de organizações humanitárias a campos de refugiados, entrega de produtos em regiões sem uma rede de transporte adequada, inspeções em áreas de riscos, monitoramento de áreas fora do alcance das câmeras de vigilância por companhias de segurança, entre outras. Entre as vantagens obtidas no sensoriamente remoto utilizando VANTs podemos citar redução dos custos para obtenção de imagens aéreas, maior flexibilidade para a aquisição de imagens em alta resolução, possibilidade de execução dos mais variados tipos de missão sem colocar em risco a vida do piloto ou operador de câmera, economia no gasto do treinamento de pilotos e maior facilidade e velocidade de incorporação de novas tecnologias [21].

Entre as atividades civis onde o uso de VANTs vem sendo cada vez mais utilizado nos últimos anos, podemos citar a agricultura. Devido às vantagens citadas anteriormente, a utilização de VANTs vem sendo aproveitada com os mais variados objetivos na agricultura tendo como foco reduzir os custos e aumentar a produtividade no campo. Imagens capturadas por VANTs capazes de voar alguns metros acima do solo representam uma alternativa entre as imagens fornecidas por satélites e as imagens obtidas por veículos limitadas pela perspectiva humana e acessibilidade das estradas. Através de contínuo monitoramento da qualidade das plantações, os VANTs permitem aos agricultores medir o progresso do trabalho em tempo real [20].

A expansão do uso de VANTs em larga escala para aplicações civis depende de dois pré-requisitos relacionados: a capacidade dos mesmos de

Figura 2.4: Imagem de VANTs multirotores e de asa fixa.

realizarem de forma autônoma manobras seguras em ambientes confinados e a remoção da exigência legal de operação supervisionada. Todavia, a exigência legal de um operador certificado dentro do campo de visão de cada VANT é uma barreira que deve perdurar nos Estados Unidos e Europa até o final desta década e remover esta barreira depende da confiabilidade e segurança dos pequenos VANTs [20]. No Brasil, ainda não foi aprovada regulamentação para a utilização de VANTs de maneira segura e legal, mas há previsão que seja lançada a primeira regulamentação em breve.

A área científica também tem se beneficiado da popularização do uso dos VANTs. Eles já são uma eficiente ferramenta na coleta de dados permitindo avanços importantes em campos como pesquisa polar, estudo de vulcões e biodiversidade selvagem [22]. Mas questões legais ainda são um empecilho na expansão do uso. De qualquer modo, pesquisadores estão empenhados em melhorar a autonomia, confiabilidade das manobras e duração dos vôos.

2.4 Segmentação

Segmentação da imagem é o processo de dividir uma imagem em um conjunto de regiões que a compõem. Estas regiões devem ser obtidas de forma a representar áreas significativas em imagens como plantações, áreas urbanas, florestas ou outras imagens obtidas por satélite [9]. A segmentação também desempenha um papel vital em inúmeros casos da imagiologia médica, como diagnósticos, localização de patologias, estudo de estruturas anatômicas e no planejamento de tratamentos [23].

Um dos objetivos da segmentação é decompôr a imagem em partes

menores para facilitar a análise posterior. Outro objetivo da segmentação é mudar a forma de representação de uma imagem. Ela permite organizar os pixels da imagem em agrupamentos que representam um maior nível de informação que os pixels brutos [9]. Entre as técnicas de segmentação, a utilização de superpixels vem se destacando pelo baixo custo computacional e alta qualidade da segmentação [24].

2.4.1 SLIC Superpixels

Algoritmos de superpixel agrupam os pixels em regiões atômicas que podem ser utilizadas como substitutas da grade de pixels. Eles capturam a redundância na imagem gerando uma estrutura que diminui significativamente a complexidade das tarefas de processamento de imagens. Esse algoritmos têm demonstrado grande utilidade em aplicações como localização de objetos e segmentação de imagens. Entre os algoritmos para geração de superpixels, o algoritmo *Simple Linear Iterative Clustering* (SLIC) se destaca pela simplicidade de uso além de baixa utilização de memória e processamento [7].

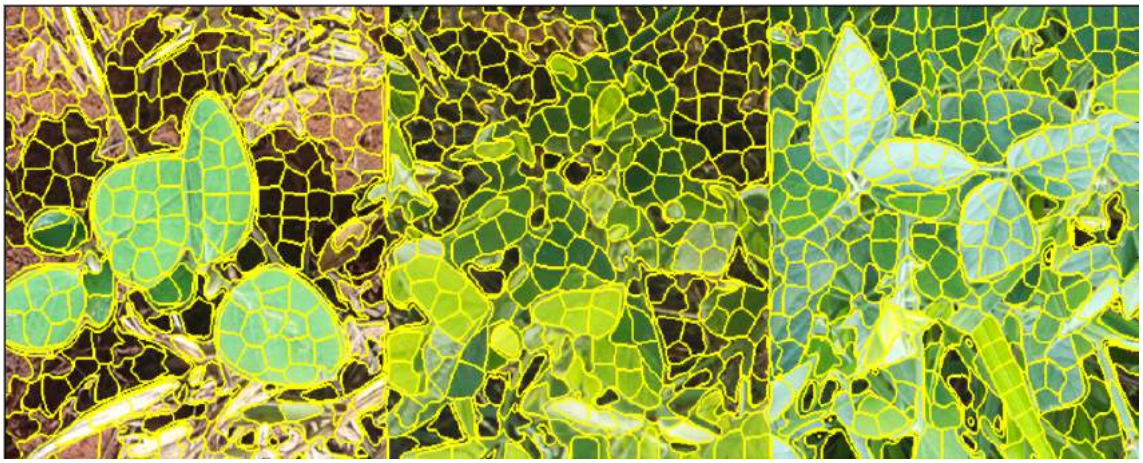

Figura 2.5: Imagem de soja após a aplicação do algoritmo *Simple Linear Iterative Clustering* (SLIC Superpixels). É possível realizar parametrizações para definir o formato do superpixel.

A extratégia deste algoritmo consiste em agrupar pixels baseado na similaridade da cor e proximidade espacial na imagem. Para isso é utilizado o espaço de cinco dimensões $[labxy]$, onde $[lab]$ é a cor do pixel no espaço de cores CIELab e $[xy]$ representa a posição do pixel na imagem. O algoritmo recebe como entrada o número de superpixels, de aproximadamente mesmo tamanho, K. Sendo assim, para uma imagem com N pixels o tamanho aproximado de cada superpixel é N/K pixels. No caso de superpixels de mesmo tamanho, haveria um centro de superpixel em cada grade no intervalo $S = \sqrt{N/K}$.

O algoritmo SLIC constroi agrupamentos de pixels utilizando uma variação do algoritmo k-means que realiza a busca num espaço reduzido, proporcional a região do superpixel, ao contrário do algoritmo tradicional que realiza a comparação com todos os centros dos agrupamentos. Esse comportamento aumenta de maneira significante a eficiência do algoritmo.

Para isso o algoritmo define K centros de superpixels $C_k = [l_k, a_k, b_k, x_k, y_k]$, com $k = [1, K]$ em cada grade no intervalo S. Então, para cada pixel $P_i = [l_i, a_i, b_i, x_i, y_i]$ da imagem, é calculada a proximidade aos centros C_k através da distância definida em 2.1:

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy} \end{aligned} \tag{2.1}$$

onde D_s representa a soma da distância no espaço CIELab, representada pelas variáveis lab, com a distância xy normalizada no plano pelo intervalo S. A variável m representa a compacidade do superpixel e tem o valor padrão 10, definido no trabalho original.

2.5 Extração de Atributos

A pesquisa em aprendizado de máquina tem como foco encontrar relacionamentos nos dados e analisar os processos para extrair tais relações dos mesmos. Um problema de aprendizado de máquina é uma tarefa executada utilizando uma aprendizagem sobre um conjunto de casos ou exemplos em substituição a tradicional execução a partir de um conjunto de regras predefinidas. Tais problemas são encontrados em uma grande variedade de aplicações como reconhecimento de padrões ou aplicações médicas [25].

Os dados a serem analisados são representados por um conjunto de características ou atributos. Encontrar uma boa representação dos dados é específico por domínio e geralmente dependentes de especialistas, embora possa ser complementada por técnicas de extração automática [25]. No reconhecimento de imagem são extraídas informações dos pixels brutos da imagem de forma a representar características como cor, forma e textura. Neste trabalho foi utilizado um conjunto composto por extratores que serão brevemente descritos nas próximas subseções.

2.5.1 Matrizes de Coocorrência - GLCM

Uma Matriz de Coocorrência GLCM (*Gray-Level Co-occurrence Matrix*) armazena informações sobre a textura de uma imagem. Esta informação é armazenada em uma matriz de frequências P com dois pixels vizinhos separados por uma distância d na imagem, sendo um deles com tom de cinza i e o outro com tom de cinza j. Estas frequências representam uma função do relacionamento angular e distância entre os pixels vizinhos [26].

Seja $p(i,j)$ a (i,j)-ésima posição da matriz P. Os desvios médio e padrão das linhas e colunas da matriz são definidos nas Equações 2.2:

$$\begin{aligned} \mu_x &= \sum_i \sum_j i \cdot p(i,j), \quad \mu_y = \sum_i \sum_j j \cdot p(i,j) \\ \sigma_x &= \sum_i \sum_j (i-\mu_x)^2 \cdot p(i,j), \quad \sigma_y = \sum_i \sum_j (j-\mu_y)^2 \cdot p(i,j) \end{aligned} \tag{2.2}$$

A partir dessas definições, temos as seguintes propriedades de texturas:

$$\begin{aligned} \textit{1) Energia:} \quad & f_1 = \sum_i \sum_j p(i,j)^2, \\ \textit{2) Contraste:} \quad & f_2 = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \,\middle|\, |i-j| = n \right\}, \\ \textit{3) Correlação:} \quad & f_3 = \frac{\sum_i \sum_j (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}, \\ \textit{4) Homogeneidade:} \quad & f_4 = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i,j), \\ \textit{5) Dissimilaridade:} \quad & f_5 = \sum_i \sum_j |i-j| \cdot p(i,j). \end{aligned} \tag{2.3}$$

2.5.2 Histograma de Gradientes Orientados

Histograma de Gradientes Orientados (*Histogram of Oriented Gradients*) é uma técnica de extração de atributos apresentada por Dalai e Triggs em 2005 [27], com o objetivo inicial de auxiliar na detecção de pessoas em imagens. A ideia básica do algoritmo é que a forma e aparência do objeto pode ser definida a partir da distribuição local de gradientes ou direções das bordas. Para isso divide-se a imagem em células e para cada célula acumula-se um histograma de gradientes de cada pixel contido naquela célula. Para melhorar a invariância à iluminação é aplicada uma normalização de contraste em blocos de células sobrepostos na imagem. Os blocos normalizados são definidos como descritores HOG.

O algoritmo apresentado pode ser resumido nos seguintes passos descritos em 1:

Algoritmo 1: HOG Histograma de Gradientes Orientados

1. Normalização da imagem
2. Cálculo dos gradientes
3. Cálculo do histograma de gradientes
4. Normalização de contraste dos blocos sobrepostos
5. Coleta dos descritores HOG gerando um vetor de atributos

2.5.3 Padrões Binários Locais

Padrões Binários Locais (Local Binary Patterns) são considerados um dos melhores extratores de textura, sendo amplamente utilizados em diversas aplicações [28]. Tem como vantagens sua invariância a mudanças em tons de cinza e eficiência computacional. Sua estratégia para detecção de textura é observar para um ponto central a variação da sua cor em relação aos seus vizinhos. Esse procedimento é realizado para todos os pixels da imagem, sendo cada pixel definido como o ponto central e tendo seu rótulo atribuído a partir do cálculo em relação aos seus vizinhos, conforme ilustrado em 2.6.

Percorrendo a imagem no sentido anti-horário, a partir da célula central esquerda, obtém-se o valor binário 011001100, correspondente ao valor decimal 204. Após realizar esse cálculo para todos os pixels presentes na imagem, o histograma dos rótulos dos pixels é então utilizado como um extrator de textura.

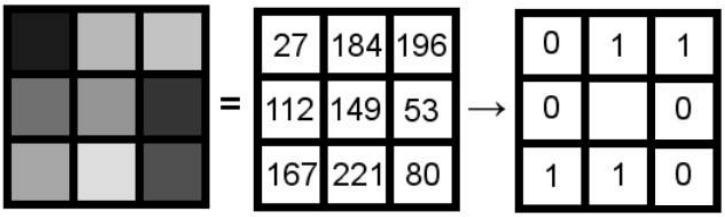

Figura 2.6: Esquema ilustrando o funcionamento dos Padrões Binários Locais.

2.5.4 Espaços de cores RGB, HSV e CIELab

Foram utilizados como atributos de cores, informações dos espaços de cores RGB, HSV e CIELab. RGB é o espaço de cores formado a partir das cores vermelho, verde e azul, cujo nome vem do inglês *Red*, *Green* e *Blue*. No modelo RGB a cor é resultante da adição dos três componentes: vermelho, verde e azul, em diferentes intensidades.

HSV é um sistema de coordenadas cilíndricas correspondente ao modelo RGB. O nome HSV vem do inglês, *Hue*, *Saturation* e *Value* que significam Matiz, Saturação e Valor. Seu objetivo é fornecer uma representação geométrica mais intuitiva e visualmente significativa que a representação cartesiana do sistema RGB.

CIELab é um espaço de cores que descreve todas as cores visíveis ao olho humano e foi concebido para servir como um modelo independente de dispositivo. As três coordenadas do CIELab representam a luminosidade da cor, sua posição entre as cores vermelho e verde e sua posição entre as cores amarelo e azul. Na Imagem 2.7, temos as representações gráficas que ajudam a descrever esses espaços.

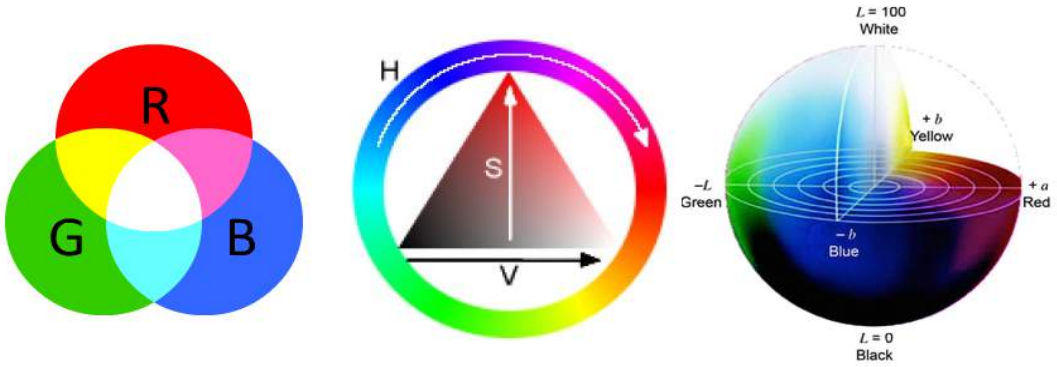

Figura 2.7: Representações gráficas tradicionais dos espaços de cores RGB, HSV e CIELab.

2.6 Classificadores

O aprendizado de máquina supervisionado é o processo de aprender um conjunto de regras a partir de instâncias ou exemplos de um conjunto de treinamento. Também pode ser definido como a criação de um classificador que pode ser generalizado para novas instâncias, não presentes no conjunto de treinamento. Um classificador é um modelo que após treinado pode ser utilizado para definir classes a instâncias de testes, cujas classes são desconhecidas, utilizando a informação dos seus atributos [29]. Árvores de decisão, redes neurais artificiais, redes bayesianas e máquinas de vetores de suporte são exemplos de classificadores. Nas próximas subseções serão descritos os classificadores utilizados neste trabalho para realizar a comparação com as Redes Neurais Convolucionais.

2.6.1 C4.5

C4.5 é um algoritmo que utiliza árvore de decisão para realizar a classificação de um conjunto de casos de testes. Uma árvore de decisão é uma estrutura que consiste de uma folha, que indica uma classe contida em um conjunto de teste, ou um nó de decisão, que especifica um teste para ser realizado, gerando uma ramificação para cada possível resultado do teste. Esta estrutura é utilizada para realizar a classificação de uma entrada partindo da raiz da árvore e movendo-se através dela, realizando os testes presentes nos nós de decisão, até que uma folha seja encontrada.

O processo fundamental desse algoritmo é a geração inicial de uma árvore a partir de um conjunto de casos de testes. O método para a construção da árvore de decisão utiliza a estratégia de divisão e conquista, baseando-se no trabalho de Hoveland e Hunt [30]. O esqueleto do método de Hunt para a construção da árvore de decisão para um conjunto de treinamento T, pode ser definido da seguinte maneira: sejam as classes do conjunto de treinamento denotadas por $C_1, C_2, ..., C_k$, temos três possibilidades:

- T possui uma ou mais entradas, todas pertencentes à classe C_j: a árvore de decisão para T é representada por uma única folha identificando a classe C_j.

- T não contém nenhuma entrada: a árvore de decisão novamente é representada por uma única folha, porém nesse caso a classe dessa folha é determinada de alguma informação que não esteja presente T. O algoritmo C4.5 utiliza a classe mais frequente no pai desse nó.

- T contém entradas que pertencem a classes variadas: neste cenário divide-se T em subconjuntos que tendam a possuir uma única classe. Isso é feito a partir de um nó de decisão contendo um teste de um determinado atributo do conjunto de entrada, que possui um ou mais resultados mutualmente exclusivos $O_1, O_2, ..., O_n$. A saída desse nó é composta por ramificações T_i, que representam subconjuntos de T, construídas a partir de todos os possíveis resultados do atributo testado. Esta abordagem é aplicada recursivamente em cada ramificação T_i, até que todos os subconjuntos possuam entradas pertencentes a uma mesma classe.

A escolha do atributo utilizado para a geração dos subconjuntos T_i é feita de modo que cada subconjunto tenha o menor número de ramificações possíveis, ou seja, deve ser escolhido um teste em cada nível de modo que o tamanho final da árvore seja o menor possível. Dada a inviabilidade de se testar todas as possíveis combinações de testes, o algoritmo usa uma estratégia gulosa

nesta escolha, baseada numa informação definida como a *Entropia* de cada subconjunto T_i [30].

Neste trabalho utilizamos o algoritmo J48 disponível no software Weka, que consiste da re-implementação em Java do software C4.5 Release 8, que contém características adicionais à versão descrita em [30].

2.6.2 AdaBoost

Boosting é uma técnica utilizada para melhorar a performance de um algoritmo de aprendizado. Na teoria, o *boosting* pode ser usado para melhorar significativamente qualquer algoritmo que gere classificadores cujos resultados são pouco melhores que um palpite aleatório. AdaBoost é um algoritmo de *boosting* proposto por Freund e Schapire em 1996 [31]. Em seu trabalho eles descreveram duas versões do algoritmo, AdaBoost.M1 e AdaBoost.M2. As duas versões são equivalentes para classificação binária, se diferenciando na classificação de problemas com mais de duas classes.

A técnica de *boosting* funciona executando um algoritmo de aprendizado fraco repetidas vezes utilizando várias distribuições do conjunto de treinamento e então combinando os classificadores obtidos em um único classificador agregado. Embora a técnica tenha como foco melhorar a performance de classificadores fracos, ela também pode ser utilizada com bons classificadores como o C4.5 [31]. Neste caso ainda é perceptível o *boost* mas os resultados são menos significativos.

O algoritmo AdaBoost recebe como entrada um conjunto S contendo m casos de treinamento e um algoritmo de aprendizado denotado originalmente como *WeakLearn*. Ele então executa o *WeakLearn* repetidamente por T iterações. A cada iteração a entrada do algoritmo *WeakLearn* é uma distribuição D_t sobre o conjunto S. Essa distribuição inicialmente é uniforme, onde $D_1(i) = 1/m$, para todo i. Nas iterações subsequentes a distribuição D_{t+1} é atualizada com base na classificação retornada do algoritmo *WeakLearn* na distribuição D_t.

No final de T rodadas são obtidas T instâncias de *WeakLearn* que serão utilizadas para realizar a classificação de um conjunto de caso de testes. Essa classificação é feita submetendo o caso de teste a cada um dos classificadores gerados e calculando a soma ponderada das classes informadas por eles, onde o peso da resposta de cada classificador é inversamente proporcional à sua taxa de erro. A classe com maior peso nesta soma é considerada como a correta pelo algoritmo AdaBoost [31].

2.6.3 Florestas Aleatórias

Florestas Aleatórias consistem de uma combinação de árvores de decisão formadas a partir de várias amostragens aleatórias e independentes de um conjunto de entrada e com a mesma distribuição para todas as árvores [32]. Essa técnica é conhecida como *bagging*, onde várias árvores geradas de maneira independente são agrupadas para fornecer uma classificação [33]. A construção dessas árvores é realizada através da seleção de uma amostragem aleatória, com reposição, do conjunto de treinamento S. Para cada amostragem selecionada é construída uma árvore correspondente àquele conjunto.

O algoritmo Florestas Aleatórias também propõe uma camada adicional de aleatoriedade ao funcionamento tradicional do *bagging*. Além de construir cada árvore utilizando uma amostragem diferente, também é modificado o modo como cada árvore é construída. No modelo tradicional cada nó é particionado usando a melhor partição entre todos os M atributos das entradas. Nesse algoritmo a escolha dos atributos é baseada em um subconjunto de atributos de tamanho m, onde m é um valor fixo para todas as árvores construídas e $m < M$. Esses atributos são escolhidos aleatoriamente e a construção de cada árvore é realizada utilizando apenas o subconjunto de atributos escolhidos [33]. A seleção do subconjunto também é independente entre todas as árvores geradas.

Para se obter a classificação de um caso de teste, é realizado o agrupamento dos resultados fornecidos por todas as árvores para aquela entrada. A classe informada pelas Florestas Aleatórias é definida a partir da maioria de votos fornecidos por todas as árvores.

A taxa de erro do classificador depende de dois fatores: a correlação entre cada par de árvores e a força individual de cada árvore na floresta. Uma maior correlação aumenta a taxa de erro enquanto a maior força diminui a taxa [32]. Esses dois fatores são diretamente proporcionais ao valor m escolhido como o número de atributos a ser utilizado na construção de cada árvore. O objetivo do algoritmo é gerar um classificador que possua simultaneamente baixa taxa de erros sem a ocorrência de *overfitting* no conjunto de treinamento, um problema comum nas árvores de decisão.

2.6.4 Máquinas de Vetores de Suporte

Máquinas de vetores de suporte *(Support Vector Machines - SVM)* é uma técnica de aprendizagem de máquina definida originalmente por Vladmir Vapnik [34]. Em sua forma mais simples, a forma linear, máquinas de vetores de suporte constituem um hiperplano de tal forma que haja uma

margem separando um conjunto de exemplos positivos e negativos em um espaço com um alto número de dimensões [34]. Dado o fato que podem haver infinitas escolhas para a margem que faça a separação desses exemplos, o objetivo é maximizar a distância dessa margem aos exemplos negativo e positivo mais próximos.

Em alguns casos não é possível separar linearmente todos os casos de um conjunto de dados, logo não há um hiperplano que divida todos os pontos negativos e positivos. Nesse caso é aplicada uma penalidade a um exemplo que falhe em se posicionar na sua margem correta. Além disso, máquinas de vetores de suporte podem ser generalizadas para classificadores não lineares. Para isso são utilizadas funções *Kernel* que incluem, por exemplo, não-linearidades gaussianas e polinomiais [34].

Apesar do algoritmo fornecer uma solução para problemas binários, essa solução pode ser generalizada para problemas com múltiplas classes. Isso pode ser alcançado através de duas estratégias: realizando o teste de uma classe contra todas as outras ou uma abordagem com vários testes entre duas classes [37].

O treinamento de uma máquina de vetores de suporte requer a solução do problema de otimização de programação quadrática. Platt [34], em 1998, propôs o algoritmo *Sequential Minimal Optimization - SMO* que apresenta uma solução onde quebra o problema de programação quadrática original em uma série de problemas menores. Esta abordagem resultou em uma queda no tempo de computação do treinamento e deixou a utilização de memória linear no tamanho do conjunto de treinamento, tornando possível o treinamento de enormes conjuntos de testes de forma eficiente. Utilizamos nesse trabalho o algoritmo *Sequential Minimal Optimization* para o treinamento das máquinas de vetores de suporte.

CAPÍTULO

3

Trabalhos Correlatos

Vários projetos que utilizam técnicas de visão computacional direcionados à agricultura vêm sendo implementados. Pesquisas têm destacado as possibilidades da utilização de sistemas de visão em áreas da agricultura como análise do comportamento animal, agricultura de precisão e orientação das máquinas, silvicultura, análise de medida e crescimento das plantações [35]. Diversos trabalhos vêm sendo realizados recentemente na identificação e classificação das ervas daninhas.

No trabalho de Herrera et al. [3] foram usados descritores de forma e *Fuzzy Decision-Making* para reconhecimento e classificação de ervas daninhas entre gramíneas e folhas largas. O conjunto de descritores de forma foi composto pelos sete momentos de Hu e seis descritores geométricos: perímetro, diâmetro, comprimento do menor eixo, comprimento do maior eixo, excentricidade e área. Máquinas de vetores de suporte foram utilizadas para comparação dos resultados, dado seu sucesso no reconhecimento de ervas daninhas em conjunto com extratores de cor, forma e textura. Das 66 imagens avaliadas, 28 apresentavam ambos tipos de ervas daninhas, 19 apenas gramíneas e 19 apenas folhas largas. O trabalho obteve como melhores resultados 85.8% de classificação utilizando todos os extratores e 92.9% utilizando o melhor conjunto de extratores avaliado, compostos por três momentos de Hu e o comprimento do maior eixo.

A utilização de máquina de vetores de suporte vem sendo adotada e conseguindo bons resultados para classificação e discriminação de ervas daninhas. Ahmed et al. [37] utilizou máquina de vetores de suporte em conjunto com extratores de cor, forma e momentos invariantes da imagem, em um total de catorze atributos. Através de validação cruzada foi verificada

a classificação de seis espécies de ervas daninhas. Esse trabalho alcançou 97.3% de precisão com a melhor combinação de extratores avaliados em um conjunto de 224 imagens, onde cada uma das seis espécies analisadas possuía entre 31 e 45 imagens.

Siddiqi et al [5] realizaram a classificação de ervas daninhas em três classes: gramíneas, folhas largas e desconhecido, classe constituída por imagens que não representavam ervas daninhas. Foi avaliado um conjunto composto por 1200 imagens, sendo 500 imagens de gramíneas, 500 imagens de folhas largas e 200 imagens definidas como desconhecido. Dessas imagens, 600 foram reservadas para o treinamento e as 600 restantes para os testes. Utilizando máquinas de vetores de suporte, o trabalho atingiu 98.1% de precisão na classificação média destas três classes.

Tellaeche et al. [38] também adotaram máquinas de vetores de suporte na identificação de ervas daninhas. O foco deste trabalho consistiu em segmentar a imagem em células e, baseado na quantidade de ervas daninhas presentes na célula, determinar a necessidade da aplicação de herbicida. O processo foi composto por uma fase de segmentação, baseada no conhecimento prévio das faixas de cultura da imagem a ser analisada, e pela fase de classificação utilizando máquinas de vetores de suporte. Foram analisadas um total de 86 imagens e 3096 células, com o objetivo de verificar se a célula em questão deveria receber a aplicação do herbicida. Os melhores resultados obtidos foram 85% de porcentagem de classificação correta.

No trabalho de Saha et al. [39] foi utilizado máquina de vetores de suporte no desenvolvimento de um sistema de detecção de ervas daninhas. O sistema proposto utiliza os passos de segmentação, extração de atributos e classificação. Para o treinamento e testes foi utilizado um conjunto de imagens composto por 60 imagens de lavouras de cenouras. Esse conjunto foi dividido em 40 imagens para treinamento e 20 imagens para os testes. As 20 imagens de testes foram divididas em quatro grupos e avaliadas separadamente. No total foram localizadas 1780 regiões dentro das 20 imagens. O sistema proposto conseguiu identificar as regiões de plantas e ervas daninhas nas imagens com sucesso de 96.37% e teve uma precisão de 88.99% na detecção de ervas daninhas.

Ishak et al. [40] utilizaram uma combinação de *Gabor wavelet* e *gradient field distribution* para obter um conjunto de vetores de atributos que permitisse a classificação das ervas daninhas como gramíneas e folhas largas. O classificador aplicado foi uma rede neural artificial SLP (*single layer perceptron*). Um conjunto com 100 imagens de gramíneas e 100 imagens de ervas daninhas de folhas largas foi utilizado para o treinamento. Um total de 400 imagens, sendo 200 imagens de gramíneas e 200 de folhas largas, com

diferentes condições de iluminação, foram utilizadas para testar a performance da rede. A rede conseguiu 93.75% de precisão média na classificação dos dois tipos de ervas daninhas.

Hung et al. [41] utilizou aprendizagem automática de atributos com *sparse autoencoders* para classificar ervas daninhas em imagens capturadas por VANTs. Essas imagens foram utilizadas na criação de mosaicos que foram analisados para a detecção das ervas daninhas. Foram realizados testes de detecção com três espécies diferentes de ervas daninhas. Embora tenham sido usadas as mesmas configurações para os algoritmos utilizados nas três espécies de ervas daninhas, a performance da rede teve perceptível diferença de desempenho nos testes. Como melhor resultado, foi alcançada precisão de 72,2%, 92,9% e 94.3% para as três diferentes espécies de ervas daninhas testadas.

Em relação aos trabalhos utilizando VANTs, um novo mercado tem sido criado com um grande potencial de expansão nos próximos anos [42], fazendo com que inúmeras pesquisas relacionadas ao uso de VANTs na agricultura venham sendo realizadas. Peña et al. [43] destacam o grande potencial do uso de imagens capturadas por VANTs no tratamento de ervas daninhas.

Entre os trabalhos envolvendo a utilização de VANTs na agricultura podemos citar Costa et al. [44], que utilizaram modelos equipados com sensores sem fio para realizar a aplicação de pesticida e fertilizantes. O objetivo dessa abordagem foi lidar com possíveis problemas da aplicação utilizando aeronaves, causados por fatores como condições climáticas. A direção e intensidade do vento, por exemplo, pode ser um fator de difícil controle na precisão da aplicação desses produtos por aeronaves, tornando o uso de VANTs uma opção atraente para esse trabalho.

O trabalho de Primicerio et al. [45] utilizou um VANT modelo VIPtero no auxílio de uma aplicação direcionada a agricultura de precisão em um vinhedo na Itália Central. Como conclusão do seu trabalho, eles apontaram que a aplicação da tecnologia no setor da agricultura pode melhorar significativamente a eficiência, sustentabilidade ambiental e os lucros do agricultor. Também afirmam que embora melhorias sejam necessárias, os resultados preliminares foram animadores.

Torres-Sánchez et al. [46] realizaram um estudo fornecendo o detalhamento das especificações e configurações técnicas de um drone utilizado com o objetivo de capturar imagens de ervas daninhas, focando no plano de missão, vôo, captura e pré-processamento das imagens. Foram realizados vários testes com diferentes câmeras RGB e multiespectrais em diferentes alturas de vôo. A conclusão é que essas configurações são dependentes de objetivos específicos. Para discriminação individual de ervas

daninhas é necessário que os vôos sejam realizados a uma altura inferior a 100 metros. Também foi verificado que em vôos com altura superior a 30 metros, pixels da cultura e ervas daninhas tem valores espectrais similares, o que aumenta a chance de erros na detecção.

Gomez-Candon et al. [47] avaliaram a precisão dos mosaicos gerados através das imagens capturadas por VANTs. Esses mosaicos são de grande importância para representar com precisão toda a plantação monitorada, possibilitando a discriminação da cultura e ervas daninhas. Eles concluíram que a altura de vôo é um importante parâmetro para se avaliar na utilização de VANTs na aquisição de imagens para detecção de ervas daninhas. Dois fatores são cruciais para se determinar essa altura: a resolução necessária para se obter uma imagem com qualidade suficiente para se discriminar a cultura das ervas daninhas e o número de imagens necessárias para se representar o mosaico. Um grande número de imagens pode tornar o processo de criação do mosaico mais difícil de ser realizado. A autonomia da bateria também é um fator limitante a ser considerado.

Este trabalho apresenta uma abordagem mais ampla em relação aos trabalhos citados. Além de classificar as imagens separadas por classes, como alguns dos trabalhos mencionados, foi construído o software Pynovisão. Este software, através da utilização da segmentação do algoritmo SLIC Superpixels, permitiu realizar a detecção das ervas daninhas em imagens de plantação capturadas por VANTs, retornando uma classificação visual e dados quantitativos sobre a presença de ervas daninhas na área fotografada. O Pynovisão também permitiu a construção de um banco de imagens robusto, contendo mais de quinze mil imagens, um valor significativamente superior aos dos trabalhos citados. Por fim, este trabalho realiza a classificação utilizando redes neurais convolucionais, uma arquitetura que vêm conseguindo excelentes resultados no reconhecimento de imagens, e faz a comparação dos resultados com classificadores que vêm sendo tradicionalmente adotados com sucesso no problema de detecção e discriminação de ervas daninhas, como máquinas de vetores de suporte.

CAPÍTULO

4

Aprendizado Profundo

Aprendizado Profundo (*Deep Learning*) é uma nova área de pesquisa de aprendizado de máquina que foi apresentada com a intenção de aproximá-lo de um dos seus objetivos originais: a inteligência artificial [48]. Deng e Yu [49], entre várias definições, definem aprendizado profundo como uma classe de técnicas de aprendizado de máquina que exploram muitas camadas de processamento de informação não linear para extração e transformação supervisionada ou não-supervisionada e para análise de padrões e classificação. Vários estudos vêm demonstrando a eficiência do aprendizado profundo em uma grande variedade de aplicações. Podemos citar o seu uso em aplicações de reconhecimento facial, reconhecimento e detecção de fala, reconhecimento de objetos em geral, processamento de linguagem e robótica. O interesse em aprendizado profundo não tem se limitado à pesquisa na área acadêmica, sendo também objeto de interesse do DARPA (*Defense Advanced Research Projects Agency*) que anunciou um projeto de pesquisa focado exclusivamente na área [50].

A sua aplicação na área de visão computacional tem alcançado um notável progresso nos últimos anos, em especial no campo do reconhecimento de objetos. Visão computacional pode ser considerada a segunda área onde a aplicação das técnicas de aprendizado profundo foi utilizada com sucesso, seguindo o reconhecimento de fala. Durante muitos anos o reconhecimento de imagem em visão computacional ficou dependente de técnicas como SIFT (*Scale Invariant Feature Transform*) e HOG (*Histogram of Oriented Gradients*). Entretanto essas técnicas tem maior facilidade em capturar baixo nível de informação, apresentando dificuldades para capturar maior nível de informação como intersecção de bordas ou identificar partes de objetos. O

aprendizado profundo visa superar essas dificuldades através de aprendizado supervisionado e não-supervisionado dos dados da imagem [49].

Vários trabalhos vêm sendo realizados e gradativamente provando a eficácia do uso de aprendizado profundo no reconhecimento de imagens. Ciresan et al. [51] utilizando uma arquitetura de redes neurais artificiais profundas conseguiram bater a performance humana no reconhecimento de dígitos escritos a mão e sinais de trânsito nos bancos de dados MNIST, NORB, entre outros.

Para alcançar esse objetivo foram utilizadas redes neurais convolucionais com várias (6-10) camadas, cada uma delas contendo centenas de mapas. Esse número de camadas é comparável ao número de camadas encontradas entre a retina e o córtex visual dos macacos [51]. A implementação de código cuidadosamente modelado para GPUs permitiu um ganho de velocidade de 50-100 vezes em relação a computadores tradicionais. Como resultado, esta implementação pela primeira vez conseguiu um resultado competitivo com o reconhecimento humano em um grande conjunto de dados. Em muitos conjuntos de imagens o algoritmo melhorou o estado da arte em 30-80% [51].

Outro avanço notável do uso de aprendizado profundo na área de reconhecimento de imagens foi obtido na competição ImageNet LSVRC de 2012. A competição consiste de um treinamento baseado em 1.2 milhão de imagens em alta resolução, para então classificar 1000 diferentes classes de imagens desconhecidas. Logo após a divulgação dos resultados obtidos nessa competição, houve intenso estudo dessas arquiteturas em visão computacional [49].

O feito em questão, alcançado por Krizhevsky et al. [52], utilizou abordagens similares ao trabalho de Ciresan et al. [51]. descrito anteriormente, com o uso de redes neurais convolucionais e GPUs para otimizar o tempo de treinamento dos conjuntos. Entretanto o reconhecimento em imagens realísticas, utilizadas nesse caso, exigem conjuntos de treinamento muito superiores que os dígitos escritos a mão ou sinais de trânsito. Os resultados alcançados mostraram que uma rede neural convolucional profunda é capaz de obter excelentes resultados em conjuntos de dados utilizando puramente aprendizado supervisionado. Também foi observado que a retirada de uma única camada reduz a performance da rede, mostrando que a profundidade da rede é determinante para o alcance dos resultados. Avanços nos anos seguintes, utilizando melhorias em abordagem similares [49] vêm comprovando quão promissora é a utilização de aprendizado profundo no reconhecimento de imagens.

O aprendizado profundo está produzindo avanços em resolver problemas que há muito anos resistiam as melhores soluções da inteligência artificial.

Sua utilização está quebrando recordes em reconhecimento de imagem e fala e tem superado outras técnicas de aprendizado de máquina em prever a atividade de potenciais drogas moleculares, análise de dados de acelerador de partículas e reconstrução de circuitos cerebrais. Além disso está gerando resultados extremamente promissores para tarefas como processamento de linguagem natural, análise de sentimentos, respostas a questões e tradução de linguagens [4].

Entre as várias arquiteturas de aprendizagem profunda, dois tipos de redes neurais profundas vêm alcançando um notável destaque em suas principais áreas. Redes neurais recorrentes tem obtido grande sucesso na manipulação de dados sequenciais como texto e processamento de linguagem natural. As redes neurais convolucionais, principal foco deste trabalho, vêm gerando grandes avanços no reconhecimento de fala, processamento de áudio, vídeo e imagens [4].

4.1 Redes Neurais Artificiais

A motivação no estudo de redes neurais tem origem na análise da maneira que o cérebro humano realiza com velocidade uma infinidade de tarefas complexas como reconhecimento visual. O cérebro é estruturado como um computador paralelo, complexo e não-linear, com capacidade de organizar suas estruturas básicas, os neurônios, para realizar os cálculos de forma rápida e paralela [53]. Cada um desses neurônios biológicos possuem complexidade e até mesmo velocidade similares a um microprocessador [54].

Os cientistas estão apenas começando a entender o funcionamento destas redes neurais biológicas [54]. É comumente aceito que todas as funções neurais biológicas são armazenadas nos neurônios e nas conexões entre eles. O processo de aprendizado é definido como a criação de novas conexões entre os neurônios e a modificação das conexões existente entre eles [54]. Baseado neste conhecimento foi modelado um conjunto de neurônios artificiais que agrupados constituem as chamadas redes neurais artificiais.

O entendimento moderno das redes neurais artificiais tem início em meados de 1940 com o trabalho de Warren McCulloch e Walter Pitts [55], que demonstraram que redes neurais artificiais poderiam calcular funções aritméticas ou lógicas. Este trabalho é frequentemente reconhecido como a origem do campo de pesquisa das redes neurais artificiais [54]. A primeira aplicação prática para redes neurais artificiais surgiu no final dos anos 50, com a invenção da arquitetura *perceptron* e seu algoritmo de aprendizado por Frank Rosenblatt [56].

Infelizmente as primeiras redes neurais sofreram de limitações que não

puderam ser superadas na época e, devido a problemas como o baixo poder de processamento dos computadores, vários pesquisadores abandonaram a área [54]. Porém na década de 1980, avanços como o desenvolvimento do algoritmo de *backpropagation* revigoraram o estudo de redes neurais artificiais que se mantém em constante evolução nas últimas décadas.

Um neurônio artificial é a unidade fundamental básica de uma rede neural artificial. Ele é definido por três elementos básicos: um conjunto de sinapses definidas como um peso, mais especificamente, um sinal de entrada x_j conectado ao neurônio k é multiplicado pelo peso da sinapse w_{kj}. O segundo elemento é um somador reponsável pela adição do resultado da multiplicação dos sinais de entrada pelas sinapses do neurônio. Por fim, o neurônio possui uma função de ativação, que define a amplitude do sinal de saída a um valor finito [53].

Em termos matemáticos o neurônio pode ser descrito nas equações abaixo:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{4.1}$$

$$y_k = \varphi(u_k + b_k) \tag{4.2}$$

onde x_1, x_2, ..., x_m são os sinais de entrada, w_{k1}, w_{k2}, ..., w_{km} são os pesos sinápticos do neurônio k e b_K corresponde ao *bias*, responsável por realizar o deslocamento da função de ativação, definida por $\varphi(\cdot)$.

Para constituir uma rede neural, os neurônios são agrupados em estruturas denominadas camadas. Essas camadas são geralmente agrupadas definindo a profundidade da rede neural. O modo como essas camadas são agrupadas e a forma de aprendizado definem a arquitetura de uma rede neural artificial. Entre as várias arquiteturas de redes neurais, as Redes Neurais Convolucionais têm como uma das características principais o uso de mapas de convolução como conjunto de pesos compartilhados entre os vários neurônios das camadas de convolução.

4.2 Convolução

Convolução é uma operação matemática entre duas funções f e g, produzindo uma terceira função, que pode ser interpretada como uma função modificada de f. No processamento de imagens, onde a imagem é definida como uma função bidimensional, a convolução é útil para detecção de bordas, suavização de imagem, extração de atributos, entre outras aplicações [57].

Sejam as funções f e g, para uma variável contínua x, a convolução é

definida como:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) \cdot g(x-\tau) d\tau \tag{4.3}$$

onde $*$ representa o operador de convolução. Para as funções f e g, quando x está definido no conjunto Z de inteiros, a equação da convolução discreta é definida como:

$$f[x] * g[x] = \sum_{n=-\infty}^{\infty} f[n] \cdot g[x-n] \tag{4.4}$$

Estendendo esta definição para funções com duas variáveis x e y, obtém-se as seguintes equações:

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1, y-\tau_2) d\tau_1 d\tau_2 \tag{4.5}$$

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2] \tag{4.6}$$

A convolução de uma imagem pode ser interpretada como o somatório da multiplicação de cada elemento da imagem, junto com seus vizinhos locais, pelos elementos da matriz que representa o filtro de convolução. Esse cálculo é ilustrado na imagem 4.1.

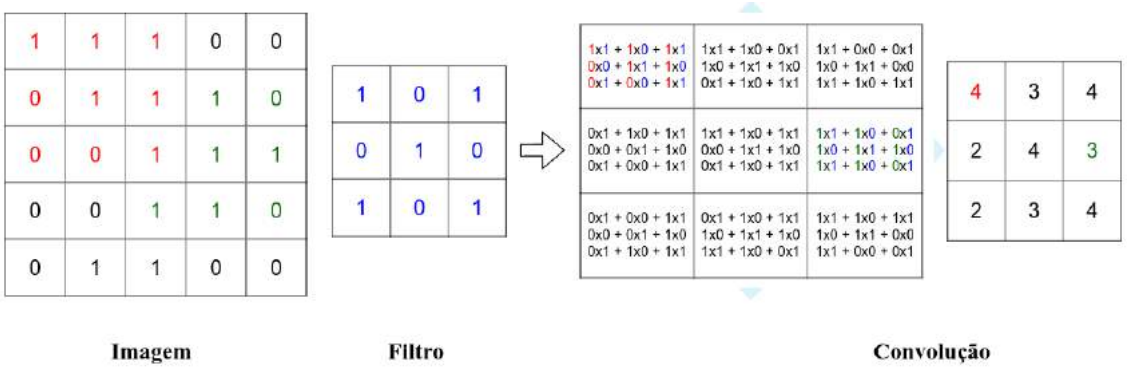

Figura 4.1: Exemplo de aplicação da convolução na imagem. À esquerda, uma imagem hipotética representada por um único canal com dimensões 5x5 que recebe a aplicação de um filtro 3x3. Á direita, uma matriz ilustrando o somatório que fornece o resultado da convolução.

Na abordagem utilizada nesse exemplo, após a aplicação do filtro de

convolução a imagem original é reduzida em tamanho proporcional às dimensões do filtro utilizado. Todavia existem outras abordagens de aplicação utilizando, por exemplo, criação de novas células adjacentes às bordas da imagem, fazendo com que a imagem resultante mantenha as mesmas dimensões da imagem original após a aplicação do filtro de convolução.

4.3 Redes Neurais Convolucionais

Em meados de 2006, um grupo de pesquisadores do *Canadian Institute for Advanced Research* (CIFAR), utilizando redes neurais artificiais, introduziram procedimentos de aprendizagem não supervisionada que poderiam criar camadas de detecção de atributos sem necessitarem de informação pré-rotulada [4]. A primeira grande aplicação desta abordagem de treinamento foi o reconhecimento de fala, sendo possível graças às novas GPUs que permitiram os pesquisadores realizarem o treinamento 10 a 20 vezes mais rápido. Em 2009, esse novo modelo de treinamento quebrou recordes em um benchmark de reconhecimento de fala que usava um pequeno vocabulário e foi rapidamente modificado para alcançar excelentes resultados com um vocabulário extenso.

Todavia, embora redes neurais geralmente tenham sido consideradas difíceis de serem bem treinadas [58], um tipo particular de rede neural profunda se mostrou muito mais fácil de treinar e generalizar do que redes completamente conectadas: as Redes Neurais Convolucionais, que alcançaram um notável sucesso prático e têm sido largamente adotadas recentemente pela comunidade de Visão Computacional [4].

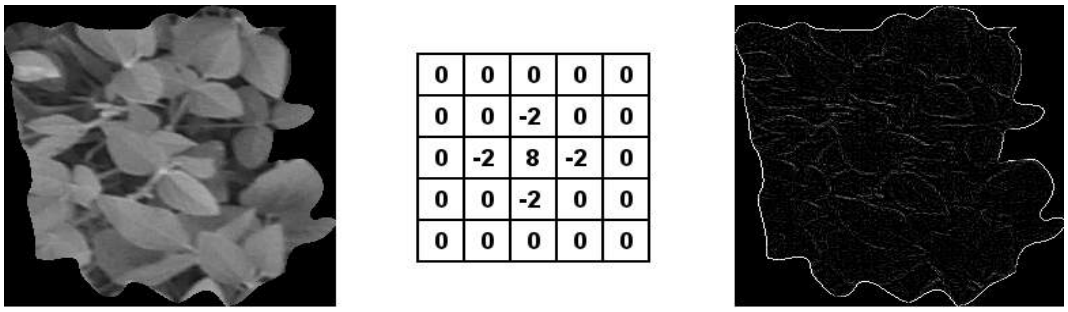

Figura 4.2: Exemplo de aplicação de um filtro de convolução sobre uma imagem de soja, em tons de cinza. O mapa de convolução utilizado visa identificar as bordas da imagem.

Redes Neurais Convolucionais ou *ConvNets* foram inspiradas na estrutura do sistema visual. Os primeiros modelos computacionais baseados nestas conectividades locais entre neurônios e em transformações da imagens

organizadas hierarquicamente são encontradas no Neocognitron de Fukushima [59]. Posteriormente LeCun, usando a arquitetura de Redes Convolucionais, alcançou o estado da arte em várias tarefas de reconhecimento de imagens [60]. O entendimento moderno da fisiologia do sistema visual é consistente com o estilo de processamento encontrado nas redes convolucionais [58]. Em alguns casos, filtros de Gabor têm sido utilizados como um pré-processamento inicial para emular a resposta visual humana às percepções visuais [50].

Redes Neurais Convolucionais Profundas foram o primeiro sucesso confiável de treinamento onde múltiplas camadas de uma hierarquia foram treinadas de maneira robusta. Elas constituem uma escolha de topologia ou arquitetura projetadas para reduzir o número de parâmetros a serem aprendidos otimizando o tempo de treinamento através de *backpropagation*. *ConvNets* são projetadas para processar dados armazenados na forma de múltiplas matrizes de uma dimensão para sinais e sequências, incluíndo linguagens, duas dimensões para imagens e espectogramas e três dimensões para vídeos e imagens volumétricas.

A Rede Convolucional proposta por LeCun, em 1989, era organizada em dois tipo de camadas, camadas convolucionais e camada de *subsampling*. Cada camada possui uma estrutura topográfica, ou seja, cada neurônio é associado com um posição bidimensional da imagem de entrada junto com um campo receptivo. Em cada localização de cada camada há um número diferente de neurônios, cada um com seu próprio conjunto de parâmetros, associados com os neurônios de uma mapa retangular da camada anterior. O mesmo conjunto de parâmetros, mas com uma diferente região, é associado com neurônios de diferentes localizações.

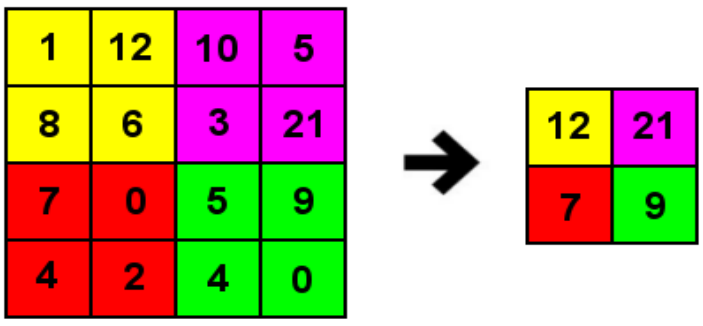

Figura 4.3: Aplicação de max pooling em uma imagem 4x4 utilizando um filtro 2x2. Além de reduzir o tamanho da imagem, consequentemente reduzindo o processamento para as próximas camadas, essa técnica também auxilia no tratamento de invariâncias locais.

Atualmente uma típica arquitetura de uma Rede Convolucional é dividida

em uma série de estágios. Os primeiros estágios são compostos de dois tipos de camadas, as camadas de convolução e as camadas de *pooling* [4]. A camada de convolução consiste em mapas de atributos, similares ao demonstrado na Figura 4.2, conectados a cada unidade da camada anterior através de um conjunto de parâmetros compartilhados entre todas as unidades e possui ReLUs (*Rectified Linear Units*), neurônios com função de ativação definida como a não-linearidade na forma descrita na Equação 4.7:

$$f(x) = max(0,x) \tag{4.7}$$

aplicados na saída de cada camada convolucional [52].

As camadas de *pooling* são uma forma de *down-sampling*. Uma típica camada de *pooling* computa o máximo local de uma determinada região do mapa de atributos, como pode ser visto na Figura 4.3. Elas são úteis por eliminar valores não máximos, reduzindo a dimensão da representação dos dados e consequentemente a computação necessária para as próximas camadas, além de criar uma invariância a pequenas mudanças e distorções locais. Dois ou três estágios de convolução, não-linearidade e *pooling* são empilhados, seguidos por mais camadas de convolução e camadas completamente conectadas. As camadas convolucionais e de *pooling* são diretamente inspiradas por noções clássicas de células simples e células complexas na neurociência visual [4].

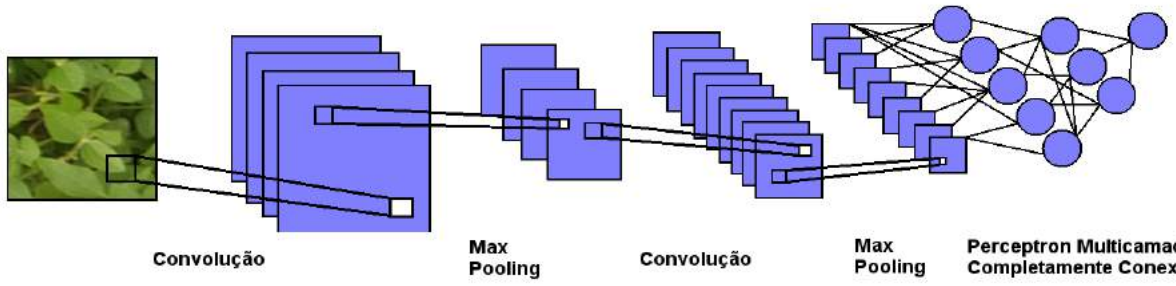

Figura 4.4: Arquitetura tradicional das Redes Neurais Convolucionais. Os primeiros estágios são compostos de camadas de convolução e *pooling*. As últimas camadas são completamente conexas.

A arquitetura utilizada por Krizhevsky no ImageNet LSVRC de 2012, considerada o marco na utilização de Redes Neurais Convolucionais para o reconhecimento de imagens, consistia de 8 camadas. As 5 primeiras camadas são convolucionais e as 3 últimas são camadas completamente conexas. A saída da última camada alimenta uma *1000-way softmax*, que produz a distribuição probabilística sobre as 1000 classes utilizadas na competição. As não-linearidades ReLUs foram utilizadas na saída de todas as camadas convolucionais e completamente conexas [52]. As imagens originais

fornecidas para o treinamento foram acrescidas das reflexões horizontais das mesmas e recortes das imagens originais e suas reflexões, para ampliar o conjunto de treinamento. Além disso foram alteradas as intensidades dos canais RGB das imagens. Para reduzir o tempo de treinamento deste conjunto de imagens e evitar o *overfitting*, foi utilizada a técnica de *dropout*, que consiste de remover aleatoriamente metade dos neurônios das camadas ocultas a cada iteração de treinamento, readicionando-os na iteração seguinte. Essa técnica também dá à rede habilidade de aprender atributos mais robustos, já que um neurônio não pode depender da presença específica de outros neurônios.

A rede proposta por Krizhevsky inspirou dezenas de outras Redes Convolucionais para reconhecimento de imagens, algumas redes bem mais profundas, possuíndo um número significativamente superior de camadas, como a *GoogLeNet*, uma arquitetura com 22 camadas, que venceu o ImageNet LSVRC de 2014 [61]. Recentes arquiteturas de Redes Neurais Convolucionais possuem de 10 a 20 camadas, centenas de milhões de pesos e bilhões de conexões. Graças ao progresso em hardware, software e algoritmos paralelos, redes como estas, que levariam semanas para serem treinadas, hoje podem ser treinadas em questão de horas [4].

Além da utilização no reconhecimento de imagens, variações de Redes Neurais Convolucionais também vêm obtendo excelentes resultados em outras áreas. Kalchbrenner et al. [62] utilizaram Redes Neurais Convolucionais Dinâmicas para uma série de tarefas de Processamento Natural de Linguagens como análise de sentimentos e classificações de tipos de perguntas. Dos Santos e Gatti [63] alcançaram o estado de arte no *Stanford Sentiment Treebank*, que contém avaliações de filmes, em classificação binárias de frases como positivo ou negativo, utilizando Redes Neurais Convolucionais Profundas. AtomNet é uma Rede Convolucional profunda projetada para predição da bioatividade de pequenas moléculas visando a descoberta de drogas, conseguindo excelentes resultados [64]. Além disso, combinações de Redes Convolucionais Neurais e Rede Neurais Recorrentes estão possibilitando novos mecanismos para a geração de legendas a partir de imagens [4].

CAPÍTULO

5

Metodologia

5.1 Visão Geral

A abordagem proposta neste trabalho para a detecção de ervas daninhas é composta por cinco fases. A primeira fase consiste na captura de imagens de plantação de soja, para a qual utilizamos VANTs. A segunda fase é a segmentação destas imagens utilizando o algoritmo de superpixels, cujos segmentos extraídos foram anotados manualmente e utilizados na construção de um banco de imagens de soja e ervas daninhas. Na terceira fase, exclusiva para os classificadores que serão utilizados na comparação à performance das *ConvNets*, realizamos a extração do vetor de atributos dos segmentos do banco de imagem com uma coleção de extratores de cor, forma e textura.

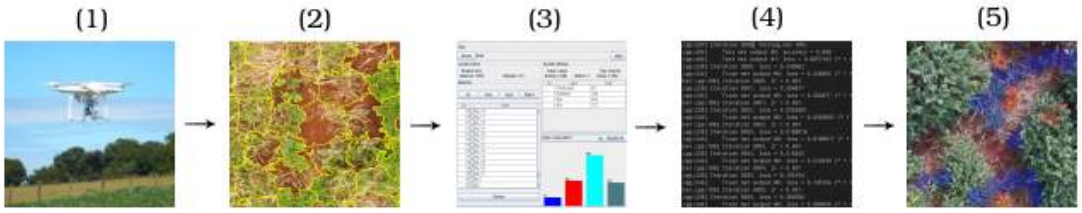

Figura 5.1: Fluxograma da metodologia: (1) Captura de imagens. (2) Segmentação. (3) Extração de atributos. (4) Treinamento. (5) Classificação.

A quarta fase consiste do treinamento dos classificadores. As redes neurais realizam este treinamento utilizando os segmentos do banco de imagens à medida que os outros classificadores utilizam o vetor de atributos

obtido na terceira fase deste processo. A última fase consiste da segmentação e classificação da imagem de uma plantação de soja, retornando dados quantitativos relativos à presença de ervas daninha na imagem.

5.2 Plantio da Soja

Foi conduzida uma plantação experimental na fazenda São José, localizada sob as coordenadas geográficas de latitude 20°24'9.88"S e longitude 54°36'31.49"O, em uma área de uma hectare. Este experimento foi feito para ser utilizado por vários projetos do grupo VANTAGRO, que envolvem pesquisas visando a solução de diversos problemas que ocorrem na soja como ataque de doenças, infestação de insetos e infestação de plantas daninhas. Deste modo a plantação foi feita com quatro quadrantes distintos, conforme Figura 5.2, com objetivo de contemplar cada um desses objetivos.

R1	R2	R3	R4	R5
L11	L12	L13	L14	L15
L21	L22	L23	L24	L25
L31	L32	L33	L34	L35
L41	L42	L43	L44	L45

R1	R2	R3	R4	R5
E11	E12	E13	E14	E15
E21	E22	E23	E24	E25
E31	E32	E33	E34	E35
E41	E42	E43	E44	E45

R1	R2	R3	R4	R5
P11	P12	P13	P14	P15
P21	P22	P23	P24	P25
P31	P32	P33	P34	P35
P41	P42	P43	P44	P45

R1	R2	R3	R4	R5
D11	D12	D13	D14	D15
D21	D22	D23	D24	D25
D31	D32	D33	D34	D35
D41	D42	D43	D44	D45

Figura 5.2: Diagrama ilustrando o plantio na fazenda São José. As letras L, E, P e D representam, respectivamente, os blocos reservados à pesquisa de lagartas, ervas daninhas, percevejos e doenças.

O plantio foi realizado em quadrantes com 4 tratamentos e 5 repetições (parcelas 6x10m). Os quadrantes foram reservados à pesquisa de lagartas, ervas daninhas, percevejos e doenças. A plantação experimental de soja transgênica foi controlada no que tange as deficiências nutricionais e infestações de plantas daninhas para que fossem obtidas imagens que permitam a busca por padrões visuais causados por deficiências específicas e infestações.

O calendário do plantio pode ser visualizado nas Tabelas 5.1 e 5.2.

Estádio	Denominação	Data	Características
VE	Emergência (1° ao 7° dia)	10/12 a 16/12	Cotilédones acima da superfície do solo formando um ângulo de 90° com seus respectivos hipocótilos.
VC	Cotilédone (8° ao 14° dia)	17/12 a 23/12	Cotilédones completamente abertos e expandidos. As bordas de suas folhas unifolioladas não mais se tocam.
V1	Primeiro nó (15° ao 21° dia)	24/12 a 30/12	Folhas unifolioladas completamente abertas.
V2	Segundo nó (a partir do 22° dia)	31/12 a 05/01	Primeira folha trifoliolada aberta.
V3 e V4	Terceiro e quarto nó (a partir do 29° dia)	06/01 a 12/01	Segunda e terceira folha trifoliolada aberta.
V5	Quinto nó (a partir do 36° dia)	13/01 a 19/01	Quarta folha trifoliolada aberta.
V6 a V(n)	Enésimo nó (Até o 49° dia)	20/01 a 26/01	Da quinta folha trifoliolada aberta ao enésimo nó ao longo da haste principal com trifólio aberto.

Tabela 5.1: Calendário do plantio com as datas para os estádios vegetativos da soja.

Estádio	Denominação	Data	Características
R1	Florescimento (50° ao 58° dia)	27/01 a 04/02	Início da floração: até 50% das plantas com flor.
R2	Pleno Florescimento (59° ao 65° dia)	05/02 a 11/02	Floração plena: maioria dos racemos com flores abertas.
R3	Início da Formação de Vagens (66° ao 75° dia)	13/02 a 22/02	Final da floração: flores e vagens com até 1,5cm.
R4	Plena Formação das vagens (76° ao 87° dia)	23/02 a 05/03	Maioria das vagens no terço superior com 2-4cm.
R5	Início do enchimento das sementes (88° ao 100° dia)	06/03 a 18/03	R5.1. Grãos perceptíveis ao tato a 10% da granação; R5.2.Maioria das vagens com granação de 10%-25%; R5.3. Maioria das vagens entre 25-50% de granação; R5.4. Maioria das vagens entre 50-75% de granação, e R5.5. Maioria das vagens entre 75-100% de granação.
R6	Pleno do enchimento das vagens (101° ao 111° dia)	19/03 a 29/03	Vagens com granação de 100% e folhas verdes.
R7	Início da maturação (112° ao 118° dia)	30/03 a 05/04	R7.1. Início: 50% de amarelecimento de folhas e vagens R7.2. Entre 51-75% de folhas e vagens amarelas, e R7.3. Mais de 76% de folhas e vagens amarelas.
R8	Maturação plena (119° ao 125° dia)	06/04 a 12/04	R8.1. Início a 50% de desfolha, e R8.2. Mais de 50% de desfolha à pré-colheita.
R9	126° dia.	13/04	R9. Ponto de maturação de colheita.

Tabela 5.2: Calendário do plantio com as datas para os estádios reprodutivos da soja.

As aplicações foram realizadas conforme mostrado na Tabela 5.3, sendo que o 'X' indica quando as aplicações foram feitas e o produto utilizado.

Herbicida	Glifosato (3,5L/ha) + Nimbus (600mL/ha)	Gramoxone (2L/ha) + Nimbus (600mL/ha)	Glifosato (2L/ha) + Verdict (0,5L/ha) + Nimbus (600mL/ha)	Glifosato (2L/ha) + Verdict (0,5L/ha) + Nimbus (600mL/ha)
Tratamento / Época de Aplicação	15-20 dias antes do plantio	1-2 dias antes do plantio	V2 / V4	Se precisar
100%	X	X	X	X
60%	X		X	
30%	X			
0%				

Tabela 5.3: Estádios fenológicos em que foram realizadas as aplicações para o manejo de plantas daninhas.

Na Tabela 5.4 seguem as quantidades necessárias para a condução das aplicações:

Item	Dose (mL/ha)	Tamanho Parcela (m2)	Volume por Parcela (mL)	Repetições	Aplicações	Volume Total (mL)	Volume com Margem de Segurança (mL)
Nimbus	600	100	6	5	27	810	972
Glifosato	3500	100	35	5	3	525	630
Glifosato	2000	100	20	5	2	200	240
Gramoxone	2000	100	20	5	1	100	120
Verdict	500	100	5	5	2	50	60

Tabela 5.4: Dose de cada item aplicado na plantação.

5.3 Captura de imagens

5.3.1 Materiais

Os registros foram realizados em forma de imagem entre 3 e 6 metros de altura. Elas foram coletadas utilizando-se do equipamento DJI Phantom 3 Professional, com peso de 1.280 gramas e velocidade máxima de 16 m/s. A bateria LiPo 4S 15.2 V tem autonomia de vôo de aproximadamente 23

minutos. O equipamento é equipado com uma câmera Sony EXMOR 1/2.3", 12.4 M (total de pixels: 12.76 M), lente FOV 94° 20 mm, suportando os formatos de arquivo FAT32/exFAT, JPEG, DNG, MP4 e MOV (MPEG-4 AVC/H.264). Possui também um gimbal com estabilização nos 3 eixos e suporte a Micro SD com capacidade máxima de 64 GB.

Figura 5.3: Imagem da lavoura de soja da Fazenda São José, capturada pela câmera Sony EXMOR 1/2.3"do VANT DJI Phantom 3 Professional, em dezembro de 2015. É notável a presença de ervas daninhas na plantação.

O delineamento foi realizado em bloco com 4 tratamentos e 5 repetições. Antes do início do período de captura, cada bloco foi sinalizado com uma estaca de bambu de 1,3 metros de altura, conforme Figura 5.2. Os vôos foram realizados entre os meses de dezembro de 2015 e março de 2016 pelo menos uma vez por semana, geralmente no período das oito às dez horas da manhã. Foram realizadas coleta de imagens da plantação, sendo a proporção da imagem utilizada 4x3, com resolução de 4000x3000px com todos os parâmetros na configuração original de fábrica.

5.3.2 Planejamento de voo

Através do DJI GO APP é possível utilizar no Phantom 3 cinco categorias de vôos inteligentes: *Follow Me, Course Lock, Waypoints, Home Lock e Point of Interest*. O plano original do vôo previa usar a categoria *Waypoints*, que consiste em setar múltiplos pontos através do GPS, com o mínimo de 5 metros de distância entre pontos adjacentes, fazendo com que o Phantom sobrevoe o percurso pré-definido enquanto efetuamos o controle manual do gimbal e da câmera. Com isso poderíamos certificar que todos os trajetos efetuados em cada trajetória respeitasse um padrão pré-definido. Todavia essa trajetória planejada para o vôo e o uso de *Waypoints* não foi utilizado

durante as visitas devido às limitações de tempo de vôo causadas por restrições em relação à quantidade e autonomia das baterias do VANT. Sendo assim a captura de imagens foi realizada através de controle manual da trajetória de vôo do Phantom.

5.3.3 Banco de imagens

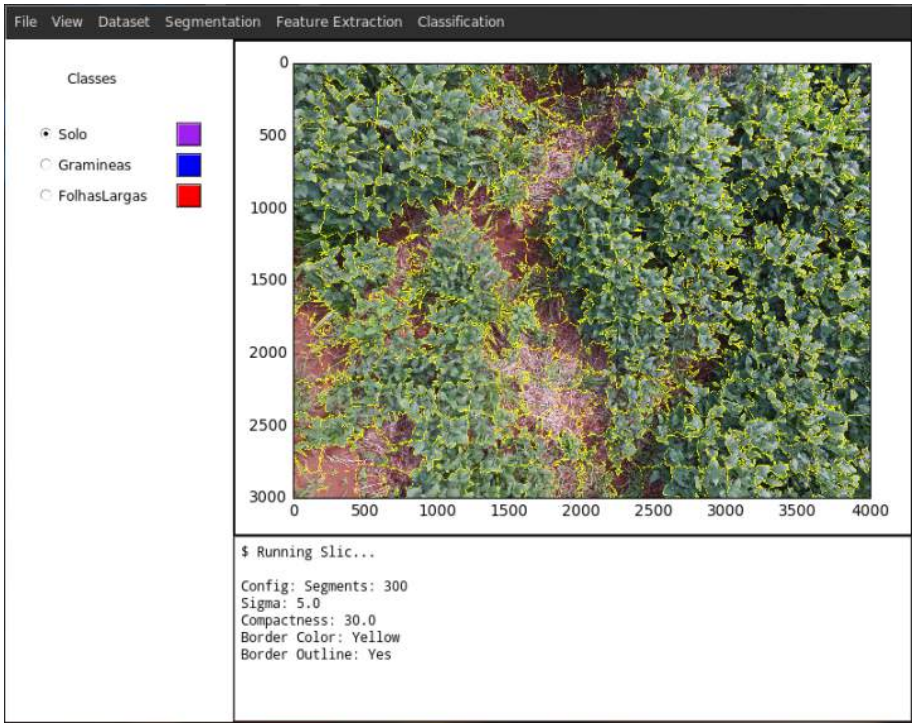

Figura 5.4: Software Pynovisão. Este software foi desenvolvido nesse trabalho e realiza segmentação, extração de atributos e classificação da imagem.

As imagens capturadas pelo VANT DJI Phantom 3 Professional foram separadas por datas, como pode ser visto na Tabela 5.5.

Data	29/12	07/01	15/01	25/01	26/01	27/01	30/01
Imagens	248	77	157	126	19	108	525

Data	02/02	03/02	05/02	08/02	11/02	16/02	18/02
Imagens	61	106	265	173	189	202	79

Data	19/02	25/02	26/02	01/03	04/03	08/03	11/03	15/03
Imagens	257	226	247	268	314	335	250	402

Tabela 5.5: Coleta de imagens por dia.

A partir desse conjunto foram selecionadas todas as imagens com ocorrência de ervas daninhas, resultando um total de 400 imagens. Através do software Pynovisão, mostrado na imagem 5.4, essas imagens foram

segmentadas e os segmentos anotados manualmente com sua respectiva classe. Esses segmentos foram utilizados na construção do banco de imagens dos testes finais,

Foram anotados segmentos de cada imagem que identificassem de maneira bem definida uma das quatro classes utilizadas neste experimento: solo, soja, ervas daninhas de folhas largas e gramíneas. Dada a grande presença de soja nas imagens, vários segmentos de soja foram ignorados arbitrariamente, para evitar um desbalanceamento no banco[1]. As imagens do mês de janeiro de 2016 não puderam ser aproveitadas devido à baixa qualidade causada por problemas configuração da câmera do VANT DJI Phantom 3.

O banco foi finalizado com 15336 segmentos, sendo 3249 de solo, 7376 de soja, 3520 gramíneas e 1191 de ervas daninhas de folhas largas. Na Tabela 5.6 pode ser vista a divisão de imagens selecionadas por data e o número de segmentos por classes.

Data	Imagens	Solo	Soja	Gramíneas	Folhas Largas	Segmentos
29/12/2015	22	541	53	152	509	1255
05/02/2016	69	875	1997	682	140	3694
16/02/2016	40	127	305	490	88	1010
18/02/2016	12	48	288	341	36	713
19/02/2016	38	315	935	373	4	1627
25/02/2016	40	11	82	267	36	396
26/02/2016	50	411	1791	128	158	2488
01/03/2016	29	3	6	221	8	238
04/03/2016	100	918	1919	866	212	3915
Total	400	3249	7376	3520	1191	15336

Tabela 5.6: Lista de imagens contendo ervas daninhas e a quantidade de segmentos selecionados, por classe. As imagens do mês de janeiro de 2016 não puderam ser aproveitadas por baixa qualidade.

5.4 Segmentação

O algoritmo SLIC Superpixels [7] foi utilizado para fazer a segmentação nas imagens e auxiliar na construção do banco. Por padrão, o único parâmetro de entrada do algoritmo SLIC Superpixels é o número de superpixels, de aproximadamente mesmo tamanho, K. Todavia, opcionalmente é possível ajustar o parâmetro compacidade que permite controlar a forma do superpixel tornando-a mais quadrada. Na implementação utilizada neste trabalho, disponível na biblioteca scikit-image

[1] É importante destacar que esta decisão introduz uma artificialidade extra no banco de imagens.

[65], também é possível configurar o parâmetro sigma, que permite aplicar uma suavização na imagem, utilizando filtros gaussianos, antes da segmentação.

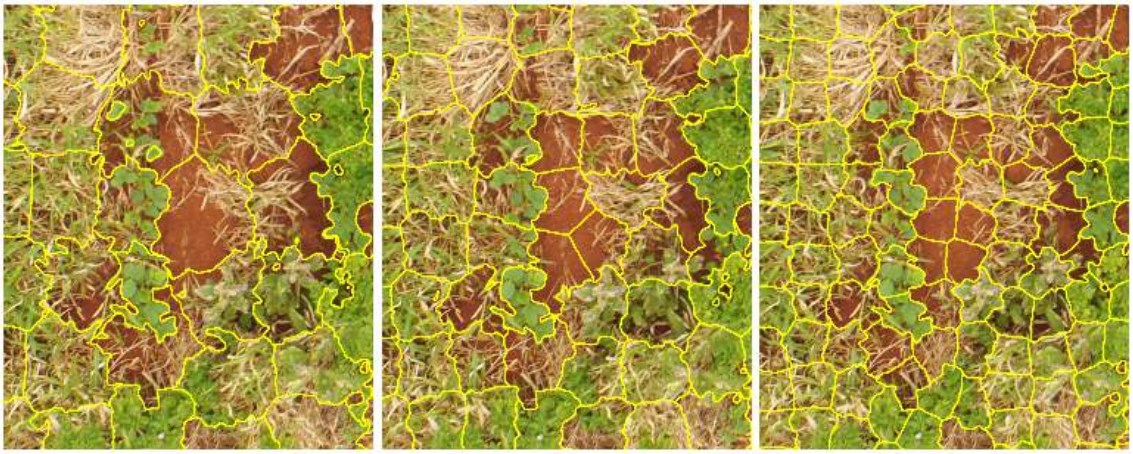

Figura 5.5: Da esquerda para a direita, temos a imagem segmentada com $K = 300,\ 600\ e\ 1200$. Para o valor 1200 os segmentos englobam as classes de maneira bem definida. Todavia a quantidade de informação relevante no segmento é bastante inferior à segmentação com o valor 300.

Foram realizados vários testes para definir o valor de K. No nosso problema o objetivo não é segmentar a imagem a nível de folha, mas separar a imagem em segmentos que contivessem várias folhas de soja ou ervas daninhas. Segmentar a imagem com um alto valor para K resulta em superpixels com menores dimensões, consequentemente, segmentando a imagem em classes bem definidas. Entretanto, superpixels com pequenas dimensões tendem a armazenar poucas características descritivas de cada classe, impactando a qualidade da classificação. Para definir o valor de K foram testados valores múltiplos de 100, no intervalo de 100 a 1200.

O valor $K = 300$ foi escolhido por lidar satisfatoriamente com ambos os problemas. As dimensões de um superpixel com essas configuração, para as imagens de dimensões $4000x3000$px utilizadas nesse trabalho, são aproximadamente $200x200$px. Estas dimensões permitiram, simultaneamente, que cada segmento contivesse uma quantidade relevante de informações da classe que representa além de uma segmentação da imagem principal em classes bem definidas.

Para o parâmetro compacidade, foram avaliados valores múltiplos de 10 no intervalo de 10 a 50. O objetivo dessa avaliação foi encontrar o valor que melhor segmentasse a imagem, mantendo no mesmo superpixel apenas elementos pertencentes a mesma classe. Em imagens capturadas em dias nublados, com pouca alternância de luz e sombra na plantação, se mostrou viável utilizar valores menores, deixando as bordas do superpixel mais rígidas às bordas dos elementos da imagem. Todavia em dias ensolarados, o

Figura 5.6: À esquerda temos a imagem segmentada com compacidade 20 e à direita com compacidade 40. Para o valor 40 os segmentos são mais sensíveis à variação de soja e ervas daninhas na imagem, agrupando no mesmo superpixel, folhas de soja com diferentes taxas de iluminação. Para o valor 20 os segmentos são mais sensíveis as mudanças de cores provocadas pelo efeito da luz e sombra.

efeito da alternância da luz e sombra na imagem tornou os superpixels mais sensíveis à iluminação da imagem, incluíndo no mesmo superpixel segmentos da imagem pertencentes a diferentes classes. Somado a isso, o algoritmo agrupava, em superpixels distintos, elementos adjacentes de uma mesma classe, devido a variação de luz e sombra.

Sendo assim fez-se necessária a utilização de valores mais altos para que a informação de proximidade espacial tivesse um maior peso em relação à similaridade de cor e iluminação. Para padronizar um valor próximo ao ideal a imagens com variados tipos de iluminação, foi escolhido o valor 40 para a compacidade.

5.5 Extração de Atributos

Para os classificadores que comparamos às redes neurais, após o passo da segmentação, realizamos a extração de atributos de cada segmento do banco utilizando uma coleção de extratores de forma, cor, textura e orientação da imagem implementados nas bibliotecas OpenCV [66] e scikit-image [65].

A coleção de extratores foi composta por um total de 218 atributos. Foram extraídos atributos da Matriz de Coocorrência GLCM, para matrizes $4x4$ nas distâncias 1 e 2 e com ângulos $0°$, $45°$ e $90°$. A partir dessas configuração foram utilizadas as seguintes propriedades: energia, contraste, correlação, homogeneidade e dissimilaridade [26], em um total de 36 características. Para o algoritmo Histograma de Gradientes Orientados [27], os 128 atributos de forma e orientação extraídos correspondem às 128 posições do vetor de HOGs calculados sobre a imagem original redimensionada para as dimensões 128x128px.

GLCM	**HOG**	**LBP**	**Cores**
glcm_cont_1_0	hog_0	lbp_0	cor_rmin
glcm_cont_1_45	hog_1	lbp_1	cor_rmax
glcm_cont_1_90	hog_2	lbp_2	cor_rmedia
glcm_cont_2_0	hog_3	lbp_3	cor_rdesvio
glcm_cont_2_45	hog_4	lbp_4	cor_gmin
glcm_cont_2_90	hog_5	lbp_5	cor_gmax
glcm_diss_1_0	hog_6	lbp_6	cor_gmedia
glcm_diss_1_45	hog_7	lbp_7	cor_gdesvio
glcm_diss_1_90	hog_8	lbp_8	cor_bmin
glcm_diss_2_0	hog_9	lbp_9	cor_bmax
glcm_diss_2_45	hog_10	lbp_10	cor_bmedia
glcm_diss_2_90	hog_11	lbp_11	cor_bdesvio
glcm_homo_1_0	hog_12	lbp_12	cor_hmin
glcm_homo_1_45	hog_13	lbp_13	cor_hmax
glcm_homo_1_90	hog_14	lbp_14	cor_hmedia
glcm_homo_2_0	hog_15	lbp_15	cor_hdesvio
glcm_homo_2_45	hog_16	lbp_16	cor_smin
glcm_homo_2_90	hog_17	lbp_17	cor_smax
glcm_asm_1_0	hog_18		cor_smedia
glcm_asm_1_45	hog_19		cor_sdesvio
glcm_asm_1_90	hog_20		cor_vmin
glcm_asm_2_0	hog_21		cor_vmax
glcm_asm_2_45	hog_22		cor_vmedia
glcm_asm_2_90	hog_23		cor_vdesvio
glcm_ener_1_0	hog_24		cor_cielmin
glcm_ener_1_45	hog_25		cor_cielmax
glcm_ener_1_90	hog_26		cor_cielmedia
glcm_ener_2_0	hog_27		cor_cieldesvio
glcm_ener_2_45	hog_28		cor_cieamin
glcm_ener_2_90	hog_29		cor_cieamax
glcm_corr_1_0	hog_30		cor_cieamedia
glcm_corr_1_45	hog_31		cor_cieadesvio
glcm_corr_1_90	hog_32		cor_ciebmin
glcm_corr_2_0	hog_33		cor_ciebmax
glcm_corr_2_45	...		cor_ciebmedia
glcm_corr_2_90	hog_127		cor_ciebdesvio

Tabela 5.7: Nome dos atributos separados por extrator. Em relação ao extrator HOG, existem ainda os atributos contíguos entre hog_33 e hog_127, correspondente às 128 posições do vetor de HOGs.

Para o algoritmo Padrões Binários Locais [28], foram calculados os valores para todos os pixels da imagem. Esses valores foram separados em um histograma de 18 faixas de mesmo tamanho e utilizados como 18 atributos de textura da imagem. Como extratores de cores, foram utilizados os atributos mínimo, máximo, média e desvio padrão da imagem representada nos espaços de cores RGB, HSV e CIELab, resultando em 36 atributos de cores.

5.6 Classificação

Para comparação com a performance das Redes Neurais Convolucionais foram realizados testes com outros classificadores. Os algoritmos utilizados foram AdaBoost M1 [31], Florestas Aleatórias [32] e Máquina de Vetores de Suporte (SVM) [67], utilizando para o seu treinamento o algoritmo SMO, a implementação da *Sequential Minimal Optimization* [34]. Para o algoritmo de *boosting* AdaBoost M1, o classificador escolhido foi o algoritmo J48, que consiste da implementação de uma evolução do algoritmo C4.5 [30].

Para estes algoritmos foi utilizada a implementação disponível no software Weka e através da biblioteca python-weka-wrapper. O software Weka consiste de uma coleção de algoritmos de aprendizado de máquina que têm como entrada arquivos no formato ARFF (Attribute-Relation File Format), um arquivo de texto ASCII que descreve uma lista de instâncias, representando a matriz de atributos de entrada para o classificador. Os ARFFs utilizados como entrada para os testes no Weka foram gerados a partir dos extratores de atributos citados na Seção 5.5. Todos os algoritmos foram executados com as configurações definidas por padrão no software Weka versão 3.6.6 [70].

Para o teste da performance das Redes Neurais Convolucionais foi utilizado o software Caffe [68]. Caffe é um framework para Aprendizado Profundo implementado em C++/CUDA para o treinamento e desenvolvimento de redes neurais. Sua motivação principal é o reconhecimento de imagens, sendo amplamente utilizado diretamente ou como base para implementações de redes neurais convolucionais. A topologia da rede neural utilizada foi a rede CaffeNet, uma replicação da topologia AlexNet [52], com 8 camadas, sendo as 5 primeiras camadas convolucionais e as 3 últimas camadas completamente conexas. Na saída da última camada foi utilizada uma *4-way softmax*, que produziu a distribuição probabilística sobre as 4 classes utilizadas neste problema.

As imagens do banco final geradas no passo da segmentação foram salvas no formato .TIF e com tamanho proporcional ao menor retângulo que englobasse todo o superpixel marcado. A rede CaffeNet utiliza como entrada

uma imagem no formato JPEG e com dimensões 256x256px. Para lidar com essa restrição, cada imagem .TIF foi colada no quadrante superior esquerdo de uma imagem composta por um fundo preto, no formato JPEG, com dimensões 512x512px. Após esse passo, a imagem foi recortada, sem redimensionamento, em relação ao quadrante superior esquerdo com dimensões 256x256px, para então ser submetida ao treinamento. Esse processo pode ser visto na imagem 5.7. Como a maior parte das imagens do banco possuía altura e largura inferiores 256 pixels, em menos de 5% casos o recorte causou perda de informação.

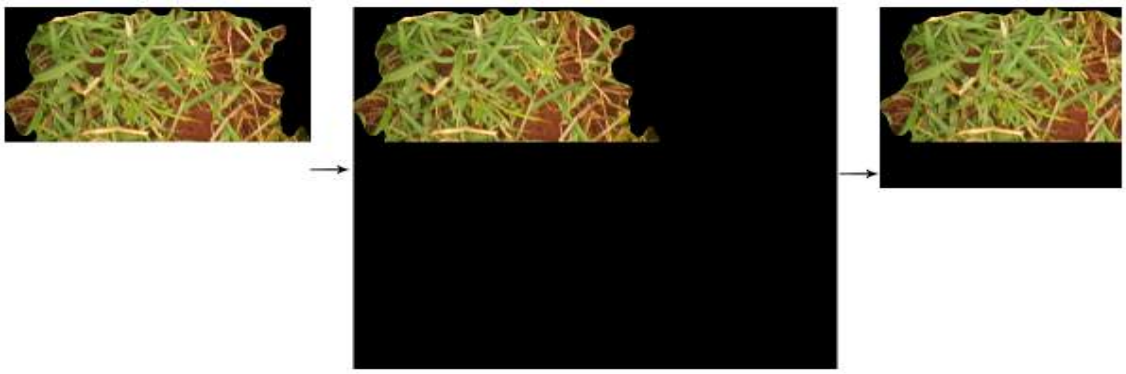

Figura 5.7: A imagem original .TIF com dimensões 324x191px é copiada para uma imagem .JPEG de dimensões 512x512px. No passo seguinte, a imagem é recortada com tamanho 256x256px.

Para ampliar o conjunto de treinamento foram aplicados os mesmos passos de pré-processamento descritos no trabalho original da AlexNet [52]. Em cada imagem do conjunto de treinamento foram aplicados 5 recortes de tamanho 227x227px, sendo um recorte partindo de cada diagonal da imagem acrescido de um recorte central. Todos esses recortes foram espelhados horizontalmente, resultando um total de 10 recortes para cada imagem. que foram utilizados no treinamento da rede neural convolucional. O procedimento é ilustrado na imagem 5.8.

Para realização dos testes de avaliação foram criados dois conjuntos a partir do banco de imagens. O primeiro conjunto foi composto com todas as classes balanceadas, ou seja, as imagens foram divididas de modo que todas as classes contivessem o mesmo número de imagens[2]. Como a classe com o menor número de imagens, folhas largas, era composta por 1191 imagens, o valor escolhido foi 1125, resultando em 4500 imagens no total . Essas 4500 imagens foram selecionadas automaticamente do banco de imagens principal através de um script que fez a seleção de maneira aleatória. Dessas imagens, 3000 foram utilizadas para o treinamento, 500 para validação e 1000 para os

[2]É importante destacar que esta decisão introduz uma artificialidade extra no banco de imagens.

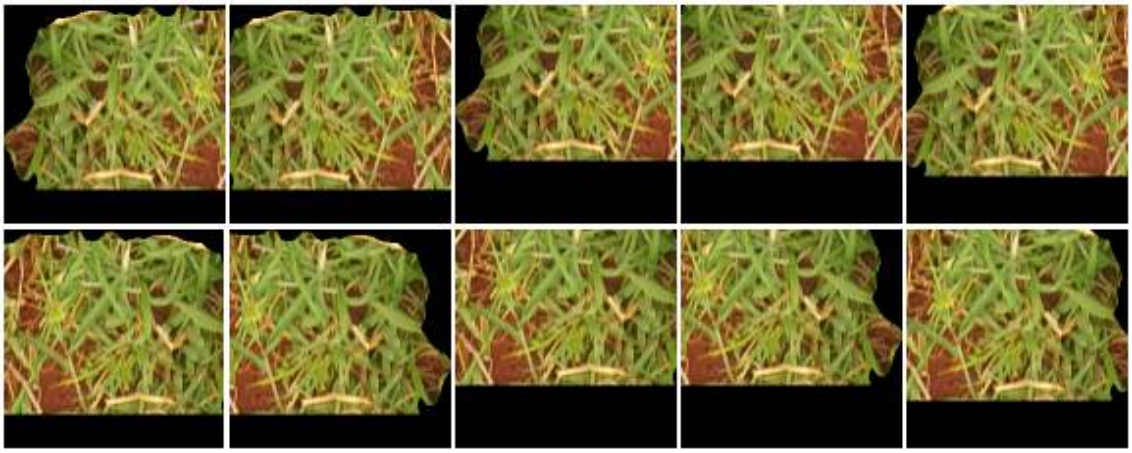

Figura 5.8: Da esquerda da direita temos a imagem com 256x256px sendo recortada para as dimensões 227x227px, a partir de cada diagonal e com um recorte central. Na fileira inferior temos o mesmo procedimento sendo aplicado com a imagem espelhada horizontalmente.

testes. As imagens de validação não foram utilizadas no treinamento dos algoritmos comparado às redes neurais, sendo então descartadas.

O segundo conjunto avaliado foi formado sem a restrição de que as classes fossem balanceadas. Sem essa restrição foi possível utilizar 15000 imagens das 15336 imagens totais do banco. As 336 imagens descartadas foram selecionadas de maneira análoga ao método utilizado no primeiro grupo, com a premissa que todas pertencessem à classe soja, por possuir um número de imagens superior às outras classes, reduzindo o impacto do descarte. Para as outras classes todas as imagens do banco foram utilizadas. Esse banco foi dividido com 70% das imagens utilizadas para o treinamento, 10% das imagens utilizadas para validação e 20% das imagens utilizadas para os testes finais, seguindo essa proporção em todas as classes. Assim como no grupo das imagens balanceadas, as imagens de validação foram descartadas para os algoritmos comparados às redes neurais.

Para o conjunto com classes balanceadas, foram realizadas 7500 iterações no conjunto de treinamento, onde em cada iteração foi utilizado um mini-batch de 50 imagens para o treinamento da rede. A taxa de aprendizado inicial da rede foi de 10^{-3}, sendo a cada 3000 iterações multiplicada por 10^{-1}. Ou seja, na iteração 3000 a taxa de aprendizado foi modificada para 10^{-4} e taxa final do treinamento foi de 10^{-5}. Para o conjunto com classes desbalanceadas, foram realizadas 15000 iterações, com a taxa de aprendizado inicial de 10^{-3}. Todavia nesse conjunto ela foi modificada apenas uma vez, com 10000 iterações, para o valor de 10^{-4}.

Os valores finais dos parâmetros utilizados neste treinamento foram obtidos após tentativas empíricas de combinações. Esses cálculos empíricos foram baseados na avaliação da precisão da rede no conjunto de validação e

através da utilização de *snapshots*. *Snapshots* representam uma cópia do modelo da rede em uma determinada iteração do treinamento, que pode ser utilizada para modificar os parâmetros de treinamento posteriores àquela iteração. Utilizando essa estrutura, foi possível testar várias ramificações de parâmetros até se chegar na combinação mais consistente para o conjunto de validação.

Para a definição do valor da taxa de aprendizado inicial, foram avaliadas as potências negativas de 10 no intervalo entre 1 e 5, ou seja, 10^{-i}, onde $1 \leq i \leq 5$. Classificações no conjunto de testes finais só foram realizadas após a definição dos parâmetros finais, ou seja, os testes não tiveram influência na escolha dos parâmetros. Dada a robustez do modelo, nenhuma modificação na arquitetura original da rede CaffeNet se mostrou necessária.

Também foi realizada a análise dos resultados baseada no nível de confiança fornecido pelas redes neurais, no conjunto com classes balanceadas, e a análise de cada classificador comparado às redes neurais usando os atributos extraídos por cada algoritmo individualmente. Esta avaliação foi feita para se testar o poder individual de cada extrator e foi realizada utilizando o conjunto de imagens com classes desbalanceadas. Além das avaliações na classificação dos segmentos do banco de imagem, foi feita a descrição do software Pynovisão e avaliação da sua performance na detecção e classificação de ervas daninhas em imagens de lavouras de soja capturadas por VANTs.

5.7 Métricas de Avaliação

Como o problema consiste de múltiplas classes, é preciso estender as definições de exemplos positivos e negativos testando cada classe contra todas as outras. Sendo assim a classe de interesse é definida como a classe positiva enquanto todas as outras são definidas como negativas [69]. O relacionamento entre as classificações como positiva e negativa pode ser definida como uma matriz de confusão 2x2, onde obtém-se as seguintes definições:

- Verdadeiro Positivo (VP): Instâncias corretamente classificadas como a classe de interesse.
- Verdadeiro Negativo (VN): Instâncias corretamente classificadas como não sendo a classe de interesse.
- Falso Positivo (FP): Instâncias incorretamente classificadas como a classe de interesse.

- Falso Negativo (FN): Instâncias incorretamente classificadas como não sendo a classe de interesse.

Assumindo essas definições, para avaliar a performance dos classificadores, além da matriz de confusão para cada classe, utilizamos as métricas precisão ou valor preditivo positivo e sensibilidade.

A precisão é definida na equação 5.1:

$$Precisão = \frac{VP}{VP + FP} \tag{5.1}$$

onde VP representa a taxa de verdadeiros positivos e FP a taxa de falsos positivos, nos fornece a proporção de todos os segmentos identificados a uma determinada classe de fato pertencerem àquela classe. Essa análise se mostra relevante ao problema porque a falsa identificação de ervas daninhas na lavoura poderia levar a um falso alerta de infestação e custos com tratamento desnecessário.

A sensibilidade é definida na equação 5.2:

$$Sensibilidade = \frac{VP}{VP + FN} \tag{5.2}$$

onde FN representa a taxa de falsos negativos, nos fornece a capacidade do algoritmo identificar cada classe nos segmentos. Baixa sensibilidade na detecção das classes correspondentes à ervas daninhas, poderia levar a uma demora na identificação da infestação das ervas daninhas, causando prejuízos à lavoura.

CAPÍTULO

6

Resultados e discussões

6.1 Avaliação com Classes Balanceadas

Na avaliação com classes balanceadas, utilizando 4500 imagens similares às demonstradas na Figura 6.1, os resultados alcançaram alta precisão e sensibilidade, como pode ser visto na Tabela 6.1. Estes resultados demonstram a eficiência da rede convolucional em lidar com um problema contendo poucas classes, quando fornecido um conjunto robusto de treinamento. Ela apresentou resultados superiores a todos os outros classificadores em todas as métricas avaliadas.

Figura 6.1: Exemplares do banco de imagens final. Na fileira superior exemplos das classes folhas largas e gramineas. Na fileira inferior exemplos das classes soja e solo.

No desempenho por classe, a classe solo teve o melhor desempenho em todos os algoritmos analisados, por apresentar valores completamente distintos de todas outras classes nos espaços de cores RGB e CIELab. Na identificação das ervas daninhas, as redes neurais apresentaram uma

superioridade mais nítida no desempenho, em relação aos outros algoritmos, apresentando valores superiores a 0.98 na precisão e sensibilidade nas gramíneas e ervas daninhas de folhas largas. Todavia é importante ressaltar que, com exceção das Florestas Aleatórias na precisão das folhas largas, todos os algoritmos comparados apresentaram resultados acima de 90% na precisão e sensibilidade de todas as classes se apresentando como boas alternativas ao problema.

	Solo	**Soja**	**Gramí-neas**	**Folhas Largas**	**Precisão**	**Sensibi-lidade**
Redes Neurais Convolucionais						
Solo	250	0	0	0	1.000	1.000
Soja	0	247	1	2	0.988	0.988
Gramíneas	0	2	246	2	0.991	0.984
Folhas Largas	0	1	1	248	0.984	0.992
Média					**0.991**	**0.991**
Máquina de Vetores de Suporte						
Solo	250	0	0	0	1.000	1.000
Soja	0	241	3	6	0.953	0.964
Gramíneas	0	9	237	4	0.967	0.948
Folhas Largas	0	3	5	242	0.960	0.968
Média					**0.970**	**0.970**
AdaBoost M1 - C4.5						
Solo	249	0	1	0	0.996	0.996
Soja	0	238	5	7	0.967	0.952
Gramíneas	1	6	236	7	0.948	0.944
Folhas Largas	0	2	7	241	0.945	0.964
Média					**0.964**	**0.964**
Florestas Aleatórias						
Solo	246	3	0	1	0.996	0.984
Soja	1	229	10	10	0.954	0.916
Gramíneas	0	6	228	16	0.908	0.912
Folhas Largas	0	2	13	235	0.897	0.940
Média					**0.938**	**0.938**

Tabela 6.1: Matriz de confusão da avaliação com classes balanceadas para todos os classificadores avaliados.

Na Figura 6.4 podemos analisar as nove imagens classificadas erroneamente pelas Rede Neural Convolucional. Na primeira imagem de folhas largas, vemos que também há na imagem a presença de gramíneas, o que induziu a classificação nessa classe. O mesmo comportamento se reflete na segunda imagem, onde há algumas folhas de soja próximas às ervas daninhas, causando o erro na identificação. As duas primeiras imagens de soja correspondem às fotografias capturadas no mês de dezembro. Essas imagens foram capturadas nos primeiros estádios da soja e com uma altura

superior às imagens obtidas em 2016, se assemelhando às imagens de ervas daninhas de folhas largas capturadas neste período.

A terceira imagem de soja possui hastes que se assemelham ao formato das gramíneas. No entanto é necessário ressaltar que essa imagem foi identificada como gramínea com uma confiança inferior a 70%. Em relação às imagens de gramíneas, é notável que a grande presença de luz e sombra dificulta a identificação da classe em duas imagens. Todavia novamente é interessante perceber que oito das nove classificações incorretas da rede foram realizadas com uma confiança inferior a 90%. Este fato nos leva a análise da confiança das classificações da rede neural também nas imagens que foram classificadas corretamente, como pode ser vista na tabela 6.2.

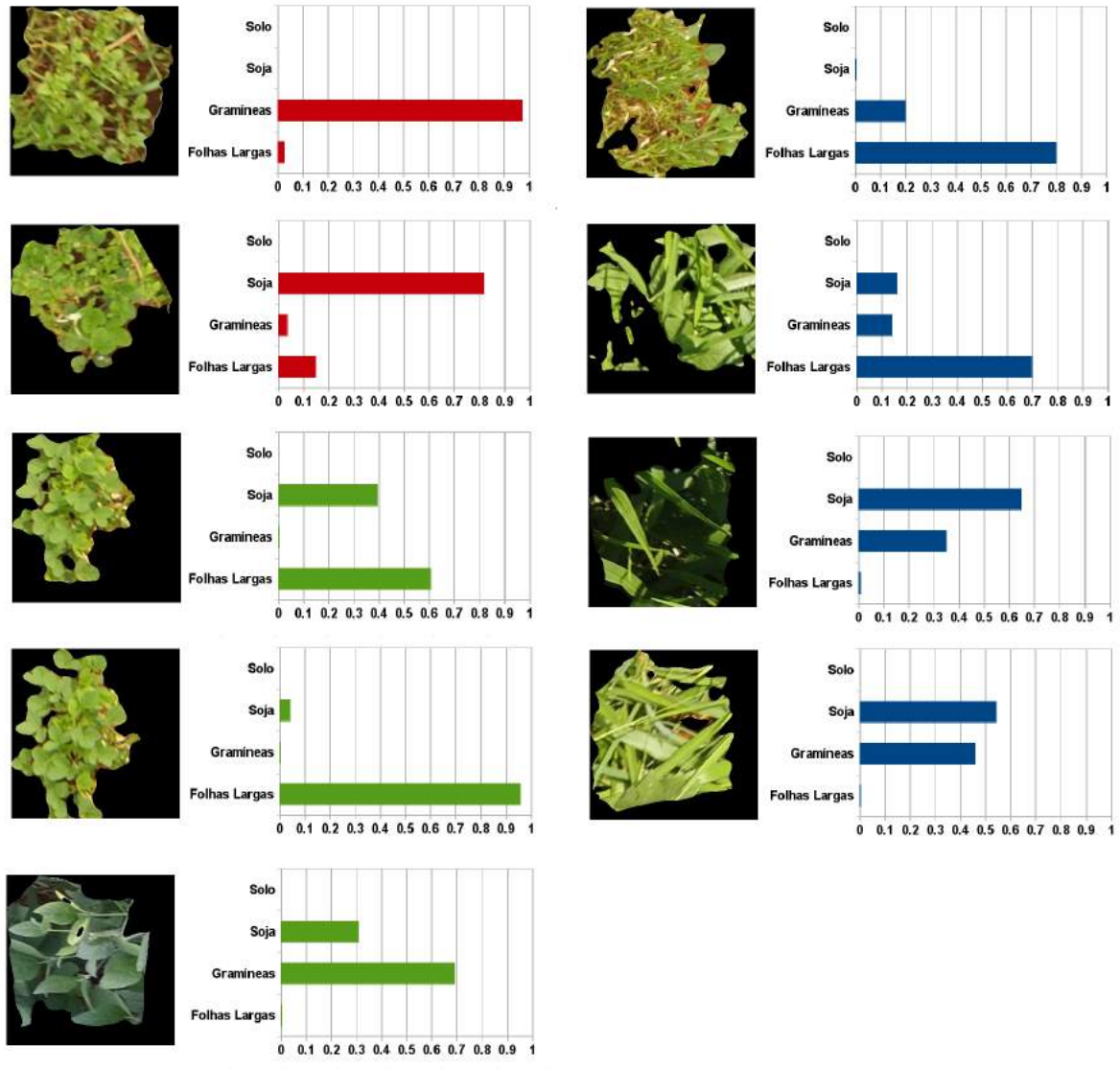

Figura 6.2: Imagens identificadas com a classe incorreta pela Rede Neural Convolucional na avaliação com classes balanceadas. As cores vermelha, verde e azul no gráfico representam, respectivamente, as ervas daninhas de folhas largas, a soja e as gramíneas. O gráfico mostra a probabilidade na classificação de cada imagem nas quatros classes do problema.

	Classificado		
Confiança	**Corretamente**	**Incorretamente**	**Não classificado**
=1	480	0	520
>= 0.999	860	0	140
>= 0.99	947	0	53
>= 0.98	963	0	37
>= 0.96	973	1	26
>= 0.94	978	2	20
>= 0.90	981	2	17
>= 0.75	985	4	11
>= 0.50	991	9	0

Tabela 6.2: Tabela ilustrando a confiança das classificações da Rede Neural Convolucional na avaliação com classes balanceadas.

Podemos ver na tabela que das 1000 imagens classificadas pela rede neural, 480 foram classificadas corretamente com 100% de probabilidade. Se estabelecermos um limiar de 0.98, temos que 96.3% das imagens foram classificadas corretamente e nenhuma delas recebeu identificação incorreta. A primeira imagem classificada incorretamente aparece apenas com o limiar de 0.96 e corresponde às ervas daninhas de folhas largas que foram classificadas como gramíneas com confiança de 0.9738.

É esperado que em algumas imagens mesmo especialistas treinados possam cometer erros. Esses erros podem ser causados por baixa qualidade da imagem, influência de iluminação ou em casos da imagem englobar duas ou mais classes em proporções similares. Utilizando a informação de confiança é possível adequar a aplicação para retornar identificação positiva apenas em casos que a confiança seja superior a um limiar predefinido, que pode ser escolhido por um especialista ou através de análise estatísticas dos resultados.

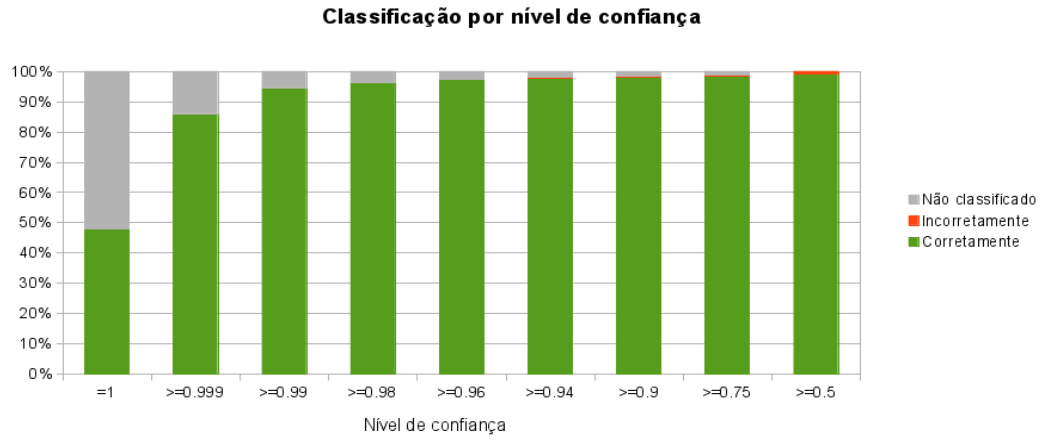

Figura 6.3: Gráfico ilustrando a confiança das classificações da Rede Neural.

6.2 Avaliação com Classes Desbalanceadas

Nesta segunda avaliação, as médias de desempenho tiveram um aumento para todos os algoritmos analisados, conforme a Tabela 6.3. No caso dos algoritmos comparados às redes neurais o aumento foi ainda mais significativo. É um comportamento interessante, pois era esperado que a rede neural precisasse de uma grande massa de dados pra conseguir resultados satisfatórios. Todavia ela já obteve resultados próximos ao ideal com uma quantidade significativamente menor de entradas, utilizadas na avaliação anterior, enquanto os outros classificadores precisaram de uma maior quantidade de dados para se aproximar do resultado das redes neurais.

	Solo	**Soja**	**Gramí-neas**	**Folhas Largas**	**Precisão**	**Sensibi-lidade**
Redes Neurais Convolucionais						
Solo	650	0	0	0	1.000	1.000
Soja	0	1406	2	0	0.995	0.998
Gramíneas	0	4	696	4	0.997	0.988
Folhas Largas	0	3	0	235	0.983	0.987
Média					**0.995**	**0.995**
Máquina de Vetores de Suporte						
Solo	649	0	1	0	0.997	0.998
Soja	0	1392	13	3	0.980	0.989
Gramíneas	2	25	669	8	0.974	0.950
Folhas Largas	0	3	4	231	0.955	0.971
Média					**0.980**	**0.980**
AdaBoost M1 - C4.5						
Solo	650	0	0	0	0.995	1.000
Soja	0	1395	9	4	0.982	0.991
Gramíneas	3	14	686	1	0.972	0.974
Folhas Largas	0	12	11	215	0.977	0.903
Média					**0.982**	**0.982**
Florestas Aleatórias						
Solo	649	1	0	0	0.991	0.998
Soja	0	1367	37	4	0.966	0.971
Gramíneas	5	29	658	12	0.928	0.935
Folhas Largas	1	18	14	205	0.928	0.861
Média					**0.960**	**0.960**

Tabela 6.3: Matriz de confusão da avaliação com classes desbalanceadas para todos os classificadores avaliados.

Além disso, apesar do aumento das médias de desempenho, a precisão e sensibilidade às ervas daninhas de folhas largas tiveram redução de desempenho em alguns algoritmos, mesmo sendo treinada uma amostragem de imagens de ervas daninhas de folhas largas de tamanho similar nas duas

avaliações. Esse comportamento demonstra como a proporção desigual de elementos treinados por classe pode levar a um impacto no resultado da classificação. Classes com mais elementos treinados podem se tornar mais sensíveis a identificação em relação a classes com menos elementos treinados, como aparenta ter acontecido no caso das folhas largas. Mesmo as redes neurais sofreram com esse problema, embora de maneira menos conclusiva.

Para analisar mais a fundo esse problema nas redes neurais, temos na Figura 6.4 as treze imagens classificadas incorretamente. Pode-se ver nas três imagens incorretas de folhas largas que embora semelhanças dessas com imagens de soja não sejam visualmente perceptíveis, elas foram classificadas como soja com grande confiança. Entretanto das 1408 imagens de soja avaliadas, as únicas duas classificadas incorretamente, foram classificadas com confiança inferior a 0.75. Nas gramíneas também temos dois casos de imagens classificadas como soja com confiança superior a 0.90. Esse comportamento não foi observado nos erros de classificação na avaliação com classes balanceadas.

Também é importante notar que apesar de ser a classe com mais imagens incorretas, sendo oito no total, as gramíneas apresentaram, na classificação pela rede neural, números superiores na precisão e sensibilidade em relação à avaliação com classes balanceadas. Na precisão o aumento foi de 0.991 para 0.997 e na sensibilidade de 0.984 para 0.988.

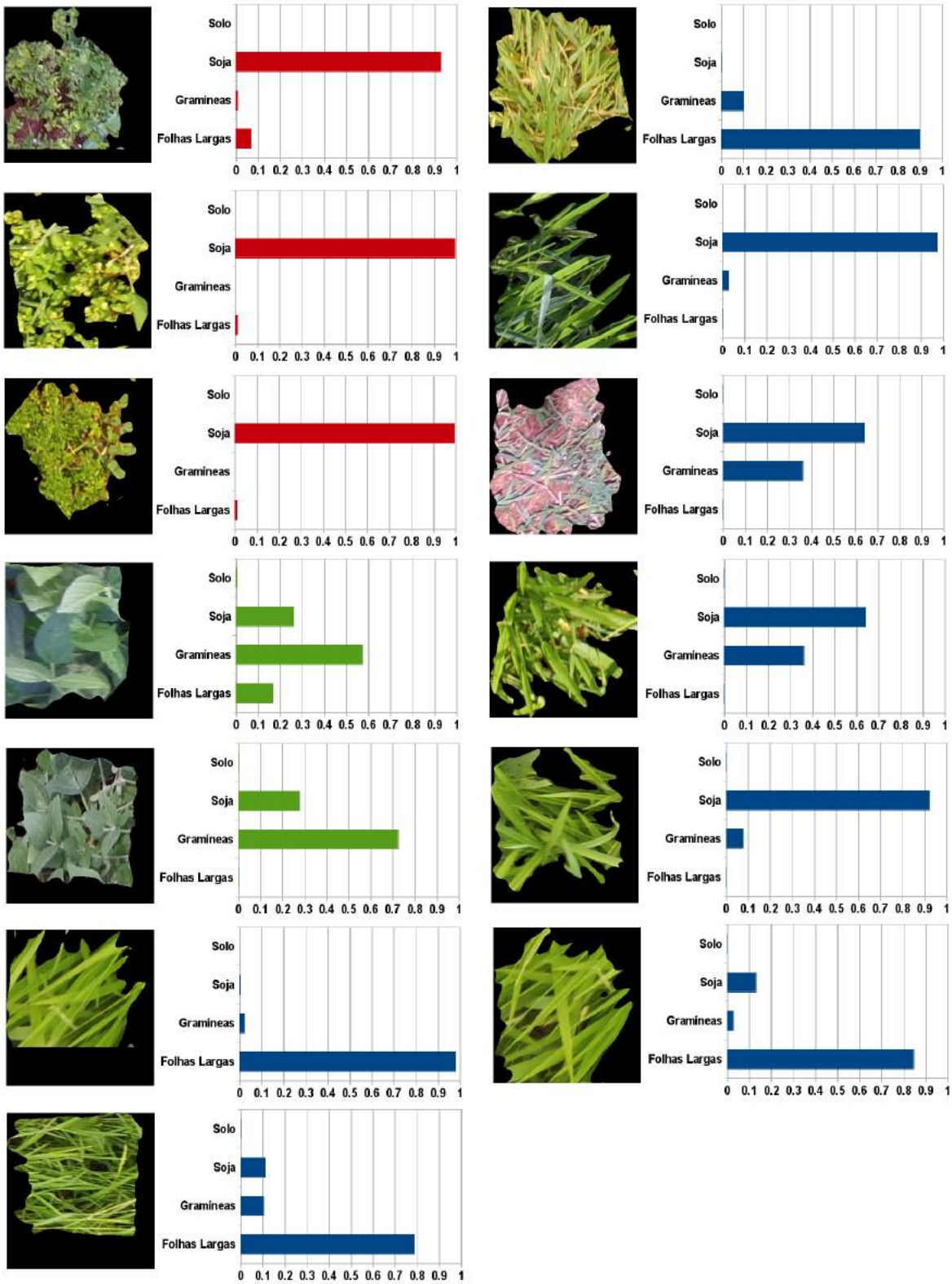

Figura 6.4: Imagens identificadas com a classe incorreta pela Rede Neural Convolucional na avaliação com classes desbalanceadas. As cores vermelha, verde e azul no gráfico representam, respectivamente, as ervas daninhas de folhas largas, a soja e as gramíneas. O gráfico mostra a probabilidade na classificação para cada imagem nas quatros classes do problema.

6.3 Avaliação por Extratores

Um fator determinante na eficácia da classificação dos algoritmos comparados às redes neurais é a qualidade dos atributos fornecidos pelos extratores escolhidos. Para avaliar esse impacto, foi feita a análise de cada classificador usando os atributos extraídos por cada algoritmo individualmente. Com essa avaliação visa-se analisar o poder de cada extrator e se o uso dos mesmos em conjunto não poderia ter de alguma forma prejudicado a performance dos classificadores.

	Todos	**GLCM**	**HOG**	**LBP**	**Cores**
Máquina de Vetores de Suporte					
Precisão	**0.980**	0.663	0.562	**0.899**	0.885
AdaBoost M1 - C4.5					
Precisão	**0.982**	0.702	0.552	0.877	**0.941**
Florestas Aleatórias					
Precisão	**0.960**	0.697	0.550	0.865	**0.930**

Tabela 6.4: Matriz de confusão dos classificadores comparados às Redes Neurais, com performance avaliada individualmente por extrator.

Vemos na Tabela 6.4 que houve uma queda perceptível na performance dos classificadores quando utilizados os extratores individualmente. Os algoritmos AdaBoost e Florestas Aleatórias conseguiram seu melhores resultados usando os atributos de cores enquanto a máquina de vetores de suporte teve seu melhor resultado usando o extrator de textura LBP. Outra análise interessante mostra que todos os classificadores analisados não tiveram bons resultados quando usados os extratores GLCM e HOG, tendo o extrator de forma HOG um desempenho muito inferior aos outros. Esse comportamento sugere que atributos de cor e textura são mais adequados para esse problema. Enquanto a cor é determinante na discriminação do solo e plantas, a textura é um fator essencial na discriminação entre a soja e as ervas daninhas.

Dada a alta precisão alcançada por todos classificadores comparados pode ser questionada a necessidade da aplicação de redes convolucionais, uma estrutura que necessita substancialmente de mais tempo e memória para o treinamento do conjunto de dados. Todavia essa última avaliação demonstra como a força da coleção de extratores de atributos utilizada foi determinante para que esses classificadores se apresentassem de maneira competitiva frente às redes neurais, desempenho que não foi alcançado usando os extratores de maneira individual.

Portanto embora seja possível deduzir que utilizando outras combinações de extratores de atributos fosse possível alcançar resultados até superiores

às redes convolucionais, temos que levar em consideração que seriam necessários exaustivos testes ou pesquisas com especialistas de domínio até encontrar a melhor combinação de extratores e classificadores para esse problema em específico. Com a rede neural temos a vantagem de abrir mão dessa parte do trabalho, chegando à solução de maneira mais direta e também flexível, afinal uma combinação de extratores que funcione bem para um determinado problema ou mesmo banco de imagens, não necessariamente pode ser utilizada em outro problema.

Através do uso de redes neurais temos uma boa chance de aplicar a mesma técnica utilizada na identificação de ervas daninhas em lavouras de soja a outros tipos de cultura apenas modificando a composição do banco de imagens de treinamento. Esse aspecto pode ser considerado um fator de compensação em relação aos altos custos de tempo e memória gastos no treinamento das redes convolucionais, principalmente pelo fato desses custos já estarem sendo significativamente reduzidos pelos recentes avanços de hardware.

6.4 Software Pynovisão

O Pynovisão foi um software desenvolvido neste trabalho com o objetivo de ser um módulo integrado de técnicas de visão computacional. Ele realiza tarefas como segmentação, extração de atributos e classificação da imagem. Para a segmentação ele fornece três algoritmos de superpixels, sendo SLIC, o algoritmo utilizado neste trabalho. Após a divisão das imagens em superpixels ele permite a criação de um banco de imagens, clicando no superpixel correspondente à sua classe específica.

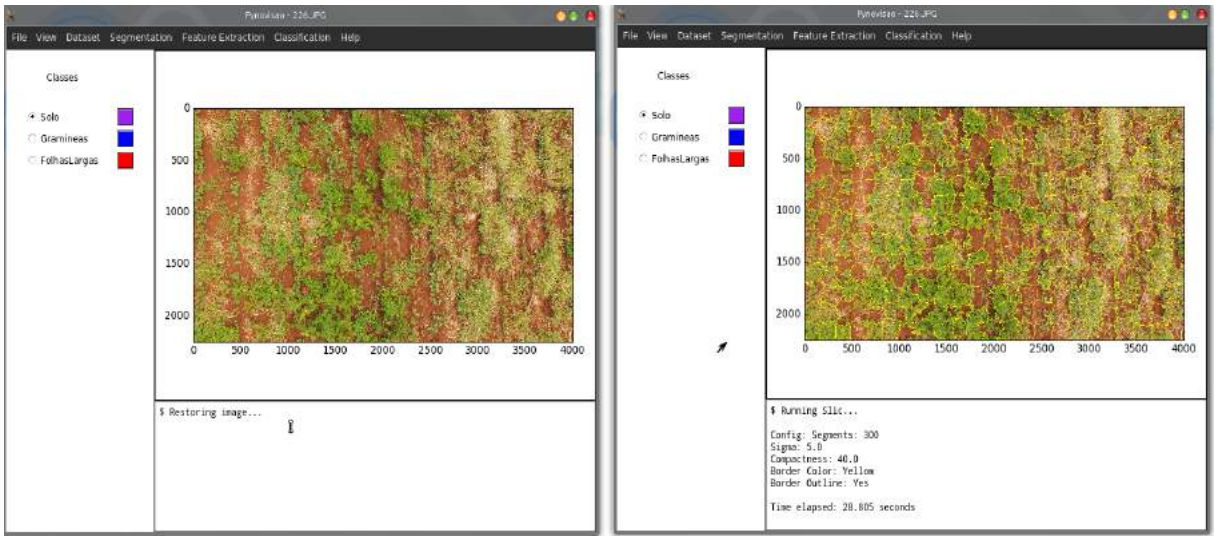

Figura 6.5: Telas de abertura de imagem e segmentação do software Pynovisão.

Após o clique no superpixel o software automaticamente seleciona o menor retângulo que engloba todo o segmento e salva a imagem desse retângulo na pasta correspondente à classe selecionada. Para realizar a extração é realizado o rastreamento de todas as pastas relativas às classes informadas pelo usuário e feita a extração de atributos de todas as imagens contidas nestes diretórios. O software também permite a configuração dos extratores de atributos e classificadores utilizados. Essa característica auxiliou na avaliação de classes por extratores, mostrada na Seção 6.3.

Todavia a funcionalidade mais importante do Pynovisão, para este trabalho, é a detecção de ervas daninhas em uma imagem de plantação de soja. Para atingir esse objetivo o software realiza a segmentação utilizando o algoritmo SLIC, através de parâmetros definidos manualmente. Após o passo de segmentação, ele salva todos os segmentos da imagem e realiza a extração de atributos nos mesmos. Por fim, ele realiza o treinamento do classificador, usando imagens do banco de imagens armazenadas previamente nos diretórios das classes. Com o classificador treinado e o vetor de atributos correspondentes aos segmentos da imagem, ele realiza a classificação individual de cada superpixel da imagem e mostra o resultado visual pintando cada segmento com a cor característica à sua classe.

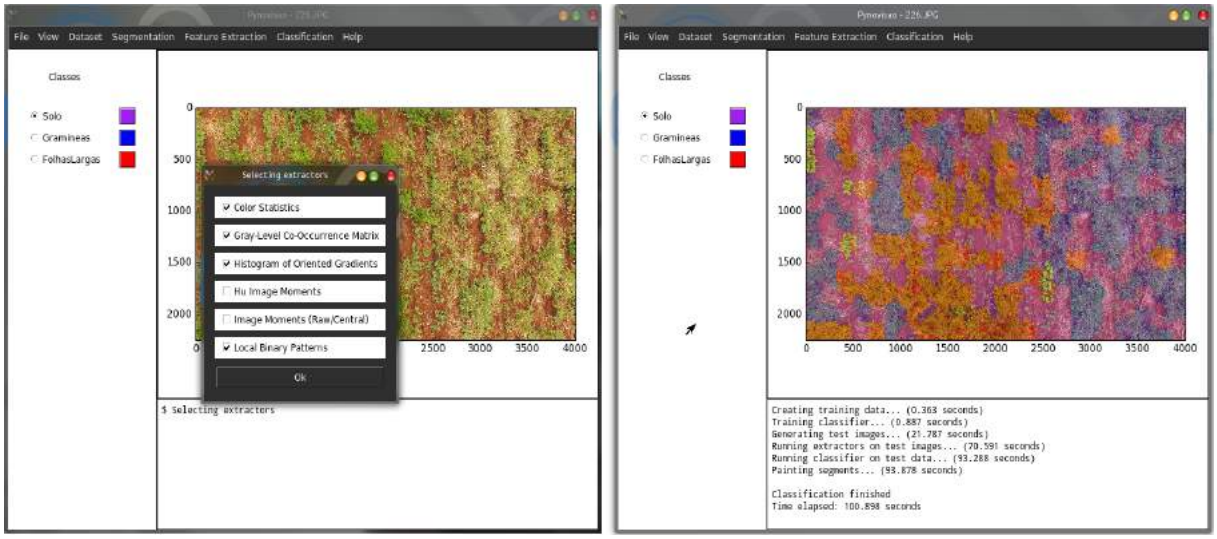

Figura 6.6: Telas de seleção de extratores de atributos e classificação da imagem do software Pynovisão.

Para a rede convolucional não é necessária a realização da extração dos atributos, pois a classificação é feita utilizando o dado bruto da imagem de cada segmento. Entretanto o treinamento da rede neural não é realizado pelo Pynovisão, sendo necessário que o mesmo seja feito por outro software. Para este trabalho, o treinamento e classificação, utilizando redes neurais, foram realizados de maneira integrada com o software Caffe.

6.5 Classificação em Imagens

Além das avaliações de classificação dos segmentos do nosso banco de imagens, através do software Pynovisão, foi possível realizar a classificação das imagens capturadas pelo VANT na plantação de soja. Para isso o software realiza a segmentação da imagem utilizando o algoritmo SLIC e após esse passo classifica cada segmento de maneira independente em uma das quatro classes definidas.

Para realizar a classificação utilizamos o modelo treinado para a avaliação com classes desbalanceadas. Para garantir que os segmentos gerados na imagem de teste não fossem idênticos a algum segmento usado no treinamento dos nossos algoritmos, foi utilizado, como parâmetro do segmentador SLIC, o valor de compacidade 30, enquanto na nossa geração do banco de imagens foi utilizado o valor 40. Isso também põe a prova a capacidade do nosso modelo de adequar a classificação a variações em relação aos parâmetros que foi treinado. De qualquer forma, é perceptível que mesmo com essa variação parte da informação do treinamento continua sendo reaproveitada.

	Solo	**Soja**	**Gramíneas**	**Folhas Largas**
Imagem de Dezembro				
Redes Neurais Convolucionais	51.49%	4.70%	24.57%	19.24%
Máquinas de Vetores de Suporte	51.53%	1.10%	25.01%	22.36%
Imagem de Fevereiro				
Redes Neurais Convolucionais	15.70%	45.11%	25.88%	13.31%
Máquinas de Vetores de Suporte	19.25%	41.35%	25.89%	13.51%
Imagem de Março				
Redes Neurais Convolucionais	12.32%	76.50%	0.00%	11.18%
Máquinas de Vetores de Suporte	11.73%	74.59%	3.19%	10.49%

Tabela 6.5: Tabela com a distribuição da classificação realizada pelo software Pynovisão nas três imagens analisadas.

Realizamos o teste com três imagens, correspondentes aos meses de dezembro, fevereiro e março. Para comparar os resultados das redes convolucionais realizamos o mesmo experimento com a máquina de vetores de suporte. O resultado da classificação em cada imagem pode ser visto na Tabela 6.5. Como essas imagens não foram marcadas manualmente por um

conjunto de especialistas antes da classificação automática, para que fosse possível realizar uma análise quantitativa do resultado, a análise da classificação pode ser feita de maneira qualitativa baseada no resultado visual fornecido pelo software.

A primeira imagem, na Figura 6.7, contém segmentos de todas as classes avaliadas. As ervas daninhas foram corretamente identificadas. Todavia houve uma certa discrepância entre os algoritmos analisados na discrimação das mesmas. Uma das possíveis causas desse comportamento é relacionada ao fato que vários trechos da lavoura contêm a presença de ambos tipos de ervas daninhas, fazendo com que vários dos superpixels englobassem mais de uma classe do problema. Sem a definição de um limiar mínimo para que a classificação de um segmento seja considerada válida, o software é obrigado a fornecer classificação a todos os segmentos, mesmo sem alto nível de confiança. Nessa imagem ambos algoritmos atingiram uma identificação precisa dos segmentos de soja.

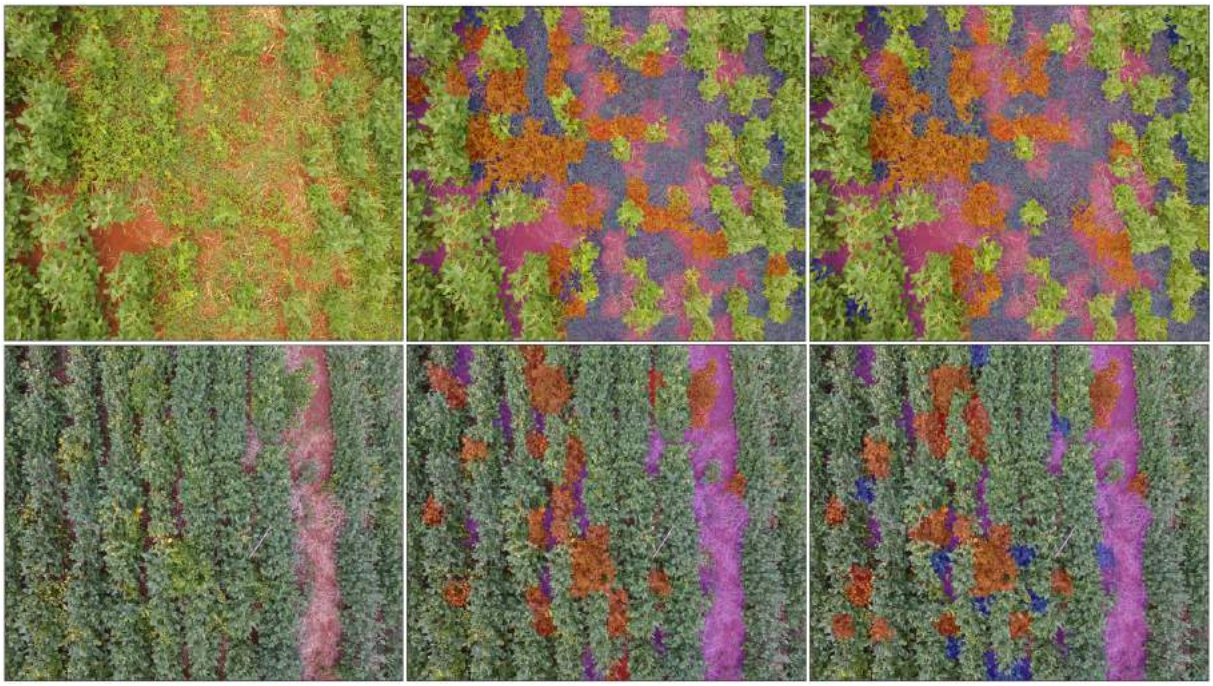

Figura 6.7: Da esquerda para a direita, a imagem original, a imagem classificada pelas Redes Neurais Convolucionais e classificada pela Máquina de Vetores de Suporte. Em vermelho as ervas daninhas de folha larga, em azul as gramíneas e em roxo o solo. A soja foi mantida na sua cor original.

Na segunda imagem analisada, correspondente à plantação no mês de março, havia apenas exemplares de ervas daninhas de folhas largas. A máquina de vetores de suporte obteve 3.19% de falsos positivos em relação às gramíneas, como pode ser visto nos segmentos em azul, no canto inferior direito da imagem 6.7. As redes neurais, corretamente, não classificaram nenhum segmento da imagem como gramíneas. Os dois algoritmos

conseguiram detectar os focos de ervas daninhas de folhas largas na imagem, apesar de classificar alguns segmentos como soja.

A última imagem analisada, correspondente ao mês de dezembro, possui algumas características a serem analisadas. Ela foi fotografada em uma altura superior ao padrão utilizado e com resolução 16x9, ao contrário do padrão 4x3 adotado nas imagens utilizadas na geração do banco. Além disso devido aos problemas na aquisição de imagens do mês de janeiro, a grande maioria das imagens utilizadas na criação do banco foi de imagens correspondentes aos estádios reprodutivos da soja. Essa imagem representa a soja no seu estádio vegetativo.

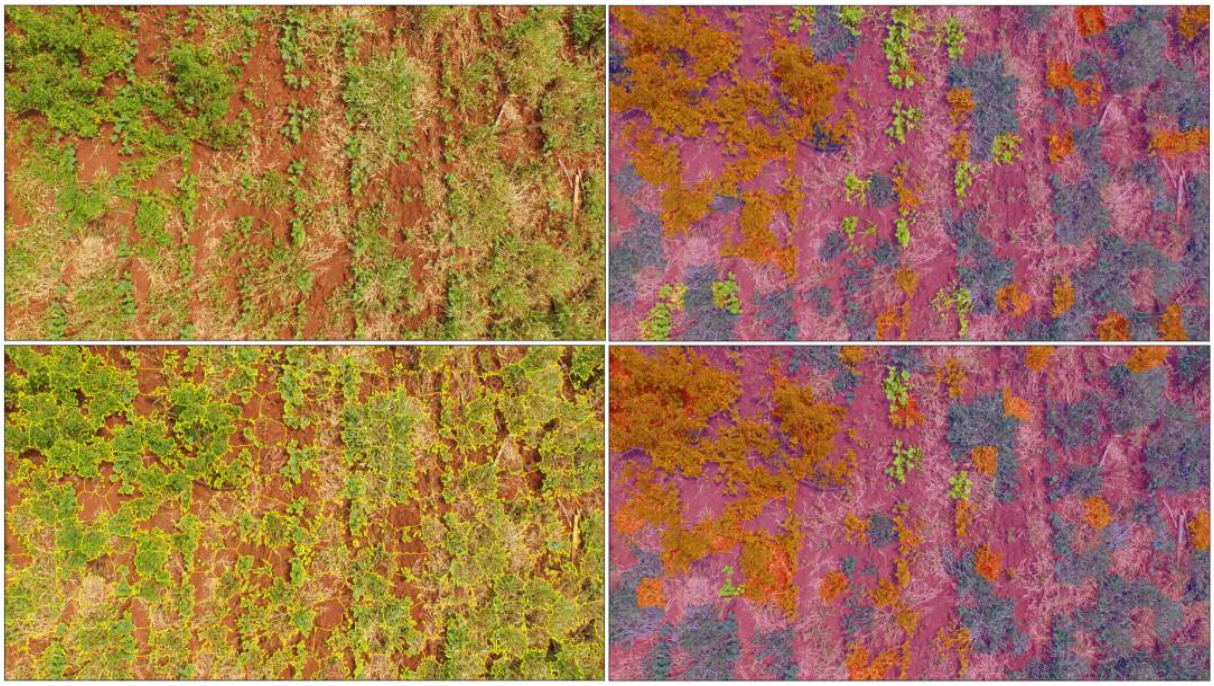

Figura 6.8: Imagem do mês de dezembro segmentada e classificada através do software Pynovisão. Na fileira superior temos a imagem original e a imagem classificada pelas Redes Neurais Convolucionais. Na fileira inferior a imagem segmentada pelo SLIC e classificada pela Máquina de Vetores de Suporte.

Mesmo com essas variações, os algoritmos analisados conseguiram uma alta precisão de detecção e discriminação das ervas daninhas na foto. Em resultados quantitativos as redes convolucionais detectaram 24.57% de gramíneas e 19.24% de folhas largas na área da plantação. A máquina de vetores de suporte obteve um resultado similar, detectando 25.01% de gramíneas e 22.36% de folhas largas. Todavia, analisando o resultado visual da imagem 6.8, é possível identificar que a diferença de 3.12% entre os dois algoritmos em relação às folhas largas correspondem a falsos positivos da máquina de vetores de suporte em relação a segmentos que representavam soja.

De qualquer modo é possível analisar a baixa taxa de falsos positivos, em

ambos algoritmos, na detecção de ervas daninhas. As redes neurais convolucionais classificaram com boa sensibilidade os segmentos de soja. Isso demonstra a capacidade do software reconhecer a soja independente do seu estádio fenológico. Mesmo tendo sido utilizados majoritariamente exemplares de soja no estádio reprodutivo, o software conseguiu uma alta taxa de precisão e sensibilidade na classificação de segmentos contendo soja no estádio vegetativo, que é o estágio onde as ervas daninhas necessitam ser identificadas e controladas para impedir os prejuízos na safra.

CAPÍTULO

7

Conclusão

Neste trabalho foi desenvolvido um software que realiza a detecção de ervas daninhas em imagens de lavouras de soja, além de discriminar as mesmas entre ervas daninhas de folhas largas e gramíneas. O algoritmo SLIC Superpixel se mostrou uma eficiente ferramenta de segmentação para imagens de plantações capturadas por VANTs, além de otimizar o tempo gasto na construção do banco de imagens. O banco construído neste trabalho é composto por mais de quinze mil imagens de solo, soja e ervas daninhas e será disponibilizado.

O uso de Redes Neurais Convolucionais alcançou excelentes resultados, com precisão superior a 98% na classificação de todas as classes. No conjunto com 15 mil imagens, foi obtido 99.5% de precisão média entre todas as imagens analisadas. Os algoritmos comparados também obtiveram bons resultados na classificação, mas as redes neurais apresentam a vantagem dos seus resultados não serem dependentes da escolha de bons extratores de atributos. Além disso, o uso de redes convolucionais podem contar com os benefícios recentes do rápido aumento do poder de processamento e memória, que viabilizam o treinamento de grandes conjuntos de imagens em um tempo viável.

Para trabalhos futuros seria interessante abordar a avaliação com um banco de imagens cobrindo uma maior gama de variáveis, como locais de plantio e altura da captura das imagens. Sem a dependência dos extratores, é provável que a metodologia utilizada neste trabalho possa ser estendida a problemas similares, envolvendo outros tipos de plantações, com poucas adequações. Dada a alta precisão alcançada neste trabalho na classificação de ervas daninhas por tipo de folhas, também seria interessante avaliar a

precisão atingida na classificação das ervas daninhas por espécies.

Este estudo também demonstrou que as redes neurais conseguiram alta precisão, sem a necessidade da utilização das quinze mil imagens do banco. Este fato sugere um estudo adicional para verificar o comportamento da rede, para este problema, sendo treinada com menos imagens, o que reduziria o tempo de treinamento e marcação manual das imagens. O uso de aprendizagem semi-supervisionada da rede é uma alternativa a ser avaliada para auxiliar a marcação manual das imagens.

Há também algumas questões que podem ser melhor exploradas como o efeito de desbalanceamento das classes na detecção de ervas daninhas e uma análise mais profunda na performance e erros dos extratores e classificadores comparados às redes neurais. Por fim, testes de avaliação do software Pynovisão utilizando imagens de referência pré-rotuladas por um conjunto de especialistas podem dar uma maior dimensão aos resultados obtidos nesse trabalho.

Referências Bibliográficas

[1] A. C. Silva, E. P. C. Lima and H.R. Batista. A Importância da Soja para o Agronegócio Brasileiro: uma Análise sob o Enfoque da Produção, Emprego e Exportação. 2011.

[2] M. A. Rizzardi and N. G. Fleck. Métodos de quantificação da cobertura foliar da infestação das plantas daninhas e da cultura de soja. *Ciência Rural,* v34(1), 2004.

[3] P. J. Herrera, J. Dorado and A. Ribeiro. A Novel Approach for Weed Type Classification Based on Shape Descriptors and a Fuzzy Decision-Making Method. *Sensors,* v14(8), p15304-15324, 2014.

[4] Y. LeCun, Y. Bengio and G. Hinton. Deep Learning. *Nature,* v521, p436-444, 2015.

[5] M. H. Siddiqi, S. Lee and A. M. Kwan. Weed Image Classification using Wavelet Transform, Stepwise Linear Discriminant Analysis and Support Vector Machines for Real-Time Selective Herbicide Applications. *Journal of Information Science and Engineering,* v30, p1253-1270, 2014.

[6] J. M. Peña Barragán, M. Kelly, A. I. D. Castro and F. López Granados. Object-base approach for crop row characterization in UAV images for site-specific weed management. *Proceedings of the 4th GEOBIA,* p426, 2012.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua e S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence,* IEEE Transactions, 34(11), p2274-2282, 2012.

[8] S. J. D. Prince. Computer Vision: Models, Learning and Inference. *Cambridge University Press,* 2012.

[9] L. Shapiro and G. Stockman. Computer Vision. *Prentice Hall,* 2000.

[10] R. Szeliski. Computer Vision: Algorithms and Applications. *Springer,* 2010.

[11] R. Jain, R. Kasturi and B. Schunck. Machine Vision. *McGraw-Hill,* 1995.

[12] V. Ugale and D. Gupta. A Comprehensive Survey on Agricultural Image Processing. *International Journal of Science and Research,* v5(1), p133-135, 2016.

[13] E. Voll, D. L. P. Gazziero, A. M. Brighenti, F. S. Adegas, C. A. Gaudencio and C. E. Voll. Dinâmica das Plantas Daninhas e Práticas de Manejo. *Embrapa,* Documentos 260, ISSN 1516-781X, 2005.

[14] J. R. B. Farias, A. L. Nepomuceno and N. Neumaler. Ecofisiologia da Soja. *Circular Técnica 48,* 2007.

[15] W. R. Fehr and C. E. Caviness. Stages of soybean development. *Ames: Iowa State University,* Special Report, 80, p12, 1977.

[16] G. R. Mohammadi and F. Amiri. Critical period of weed control in soybean (Glycine max) as influenced by starter fertilizer. *Australian Journal of Crop Science,* v5(11), p1350-1355, 2011.

[17] B. Hartzler. Protecting soybean yields from early-season competition. *IC-498,* v3, p76, 2007.

[18] R. C. Van Acker, C. J. Swanton and S. F. Weise. The Critical Period of Weed Control in Soybean. *Weed Science,* v41, p194-200, 1993.

[19] A. N. Chaves, P. S. Cugnasca and J. J. Neto Busca Adaptativa com Múltiplos Veículos Aéreos Não Tripulados. *Revista de Sistemas e Computação,* Salvador, v2(1), p53-59, 2012.

[20] D. Floreano and R. J. Wood. Science, technology and the future of small autonomous drones. *Nature,* v521, p460-466, 2015.

[21] G. A. Longhitano. VANTs para sensoriamento remoto: aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas. *Escola Politécnica da Universidade de São Paulo,* 2010.

[22] E. Marris. Drones in science: Fly, and bring me data. *Nature,* v498, p156-158, 2013.

[23] D. L. Pham, C. Xu and J. L. Prince. Current methods in medical image segmentation 1. *Annual review of biomedical engineering,* v2(1), p315-337, 2000.

[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua e S. Susstrunk. SLIC Superpixels. *EPFL Technical,* Report 149300, 2010.

[25] I. Guyon and A. Elisseeff. An introduction to feature extraction. *Feature extraction,* p1-25, 2006.

[26] L. K. Soh and C. Tsatsoulis. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on geoscience and remote sensing,* v37(2), p780-795, 1999.

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05),* v1, p886-893, 2005.

[28] T. Ahonen, A. Hadid and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence,* v28(12), p2037-2041, 2006.

[29] S .B .Kotsiantis, I. Zaharakis and P. Pintelas. Supervised machine learning: A review of classification techniques. 2007.

[30] J. R. Quinlan. C4.5: Programs for Machine Learning. *Morgan Kaufmann,* 1993.

[31] Y. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference,* 1996.

[32] L. Breiman. Random Forests. *Machine Learning,* v45, issue 1, p5-32, 2001.

[33] A. Liaw and M. Wiener. Classification and regression by randomForest. *R news,* v2(3), p18-22, 2002.

[34] J. C. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Microsoft Reasearch,* 1998.

[35] T. Brosnan and D. Sun. Inspection and grading of agricultural and foodproducts by computervision systems – a review. *Computers and Electronics in Agriculture,* p193-213, 2002.

[36] A. Arif and K. M. Butt. Computer vision based navigation module for sustainable broad-acre culture robots. *ISSN 1013-5316,* 2014.

[37] F. Ahmed, H. A .Al-Mamun, A. S. M. H. Bari, E. Hossain and P. Kwan. Classification of crops and weeds from digital images: A support vector machine approach. *Crop Protection,* v40, p98-104, 2012.

[38] A. Tellaeche, G. Pajares, X. P. Burgos-Artizzu and A. Ribeiro. A computer vision approach for weeds identification through Support Vector Machines. *Applied Soft Computing,* v11(1), p908-915, 2011.

[39] D. Saha, A. Hanson and S. Y. Shin. Development of Enhanced Weed Detection System with Adaptive Thresholding and Support Vector Machine. *Proceedings of the International Conference on Research in Adaptive and Convergent Systems,* ACM, p85-88, 2016.

[40] A. J. Siddiqi, A. Hussain e M. Mustafa. Weed image classification using Gabor wavelet and gradient field distribution. *Computers and Electronics in Agriculture,* v66, p53-61, 2009.

[41] C. Hung, Z. Xu e S. Sukkarieh. Feature Learning Based Approach for Weed Classification Using High Resolution Aerial Images from a Digital Camera Mounted on a UAV. *Remote Sensing,* v6(12), p12037-12054. 2014.

[42] I. Colomina and P. Molina. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing,* v92, p79-97, 2014.

[43] J. M. Peña, J. T. Torres-Sánchez, A. I. de Castro, M. Kelly and F. Lopez-Granados. Weed Mapping in Early-Season Maize Fields Using Object-Based Analysis of Unmanned Aerial Vehicle (UAV) Images. *PLoS One,* v8(10), e77151, 2013.

[44] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório and P. A. Vargas. The use of Unmanned and wireless sensor network in agricultural applications. *Geoscience and Remote Sensing Symposium (IGARSS),* 2012 IEEE International, p5045-5048, 2012.

[45] J. Primicerio, S. F. Gennaro and E. Fiorillo. A flexible unmanned aerial vehicle for precision agriculture. *Springer,* 2012.

[46] JJ. T. Torres-Sánchez, F. Lopez-Granados, A. I. de Castro and J. M. Peña Barragán. Configuration and Specifications of an Unmanned Aerial Vehicle (UAV) for Early Site Specific Weed Management. *PLoS One,* v8(3), e58210, 2013.

[47] D. Gomez-Candon, A. I. De Castro and F. Lopez-Granados. Assessing the accuracy of mosaics from unmanned aerial vehicle (UAV) imagery for precision agriculture purposes in wheat. *Precision Agric,* v15, p44-56, 2014.

[48] Lisa LAB. Deep Learning Tutorial. *University of Montreal,* 2015.

[49] L. Deng and D. Yu. Deep Learning Methods and Applications. *Foundations and Trends in Signal Processing,* v7, 2013.

[50] I. Arel, D. C. Rose and T. P. Karnowski. Deep Machine Learning – A New Frontier in Artificial Intelligence Research. *The University of Tennessee,* IEEE Computational Intelligence Magazine, 2010.

[51] D. Ciresan, U. Meier and J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. 2012.

[52] A. Krizhevsky, I. Sutskever and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of Neural Information Processing Systems,* p1097-1105, 2012.

[53] S. Haykin and N. Network. A comprehensive foundation. *Neural Networks,* v2(2004), p41, 2004.

[54] H. B. Demuth, M. H. Beale, O. De Jess and M. T. Hagan. Neural network design. *Martin Hagan,* 2014.

[55] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics,* v5(4), p115-133, 1943.

[56] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review,* v65(6), p386, 1958.

[57] J. R. Parker. Algorithms for image processing and computer vision. *John Wiley & Sons,* 2010.

[58] Y. Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning,* v2(1), p1-127, 2009

[59] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics,* v36(4), p193-202, 1980.

[60] Y. LeCun, J. S. Denker, D. Hederson, R. E. Howard, W. Hubbard and L. D. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems,* 1990.

[61] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv,* p1409.4842, 2014.

[62] N. Kalchbrenner, E. Grefenstette and P. A. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv,* p1404.2188, 2014.

[63] C. N. Dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. *In Proceedings of the 25th International Conference on Computational Linguistics (COLING),* Dublin, Ireland, 2014.

[64] I. Wallach, M. Dzamba and A. Heifets. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. *arXiv preprint arXiv,* p1510.02855, 2015.

[65] Van der Walt, S. J. G. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart and T. Yu scikit-image: image processing in Python. *PeerJ,* v2, p453, 2014.

[66] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools,* 2000.

[67] L. Zanni, T. Serafini and G. Zanghirati. Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems. *Journal of Machine Learning Research 7,* p1467-1492, 2006.

[68] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013.

[69] L. Brett.. Machine learning with R. *Packt Publishing Ltd,* 2013.

[70] "M. Hall, E. Frank, G. Holmes and Pfahringer The WEKA Data Mining Software: An Update. *SIGKDD Explorations,* v11(1), 2009.