UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MATHEUS CASSALI DA ROSA

REDES NEURAIS CONVOLUTIVAS APLICADAS À DETECÇÃO DE ERVAS DANINHAS

Porto Alegre
2019

MATHEUS CASSALI DA ROSA

REDES NEURAIS CONVOLUTIVAS APLICADAS À DETECÇÃO DE ERVAS DANINHAS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.
Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Valner Brusamarello

Porto Alegre
2019

MATHEUS CASSALI DA ROSA

REDES NEURAIS CONVOLUTIVAS APLICADAS À DETECÇÃO DE ERVAS DANINHAS

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Valner Brusamarello, UFRGS
Doutor pela UFSC – Florianópolis, Brasil

Banca Examinadora:

Prof. Dr. Alexandre Balbinot, UFRGS
Doutor pela UFRGS – Porto Alegre, Brasil

Profa. Dra. Adriane Parraga, UERGS
Doutora pela UFRGS – Porto Alegre, Brasil)

Prof. Dr. Ricardo de Azambuja,
Doutor pela Plymouth University – Plymouth, Reino Unido)

Coordenador do PPGEE: _____
Prof. Dr. João Manoel Gomes da Silva Junior

Porto Alegre, abril de 2019.

DEDICATÓRIA

Dedico este trabalho aos meus pais, em especial pela dedicação e apoio em todos os momentos difíceis.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Aos colegas do PPGEE pelo seu auxílio nas tarefas desenvolvidas durante o curso e apoio na revisão deste trabalho.

Ao CNPq pela provisão da bolsa de mestrado.

RESUMO

A discriminação entre plantas de cultura e erva daninha é um passo muito importante para os sistemas de pulverização seletiva, cuja aplicação é feita apenas onde for necessário. Tais sistemas são essenciais para evitar o desperdício de agroquímicos e reduzir os impactos econômicos e ambientais. Várias técnicas de visão computacional foram desenvolvidas para abordar o problema, no entanto, existem poucos trabalhos utilizando *deep learning* para essa finalidade. Neste trabalho é analisado o desempenho de segmentação de ervas daninhas e plantas através de duas arquiteturas diferentes de aprendizagem profunda para a segmentação semântica: Rede Totalmente Convolucionais (*Fully Convolutional Network*) e SegNet. Um banco de dados aberto com 39 imagens de plantas e ervas daninhas foi usado para estudo de caso. Os resultados mostraram uma precisão global maior que 90% no conjunto de validação para ambas as arquiteturas. Num segundo experimento, novas redes FCN foram treinadas com diferente pré-processamento das imagens e diferentes proporções treino/teste do conjunto de dados para avaliar o impacto dessas ações no desempenho de segmentação.

Palavras-chave: Detecção de ervas daninhas, *deep learning*, FCN, SegNet, segmentação de imagem.

ABSTRACT

The discrimination between crop and weed is a very important step for selective spraying systems which the application is made only where is necessary. Such systems are essential to avoid waste of agrochemicals and reduce economic and environmental impacts. Several computer vision techniques were developed to solve this problem, however there are few works using deep learning for this purpose. In this work the precision for weed and crop segmentation is analyzed using two different architectures of deep learning for semantic segmentation: Fully Convolutional Network and SegNet. An open database with 39 plant and weeds images was used for case study. The results showed global accuracy higher than 90% on the validation set for both architectures. In a second experiment, new FCN networks were trained with different image preprocessing and different training/test ratios of the dataset to evaluate the impact of these actions on segmentation performance.

Keywords: weed detection, deep learning, FCN, SegNet, image segmentation.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

CNN	*Convolutional Neural Network* - Rede Neural Convolutiva
FCN	*Fully Convolutional Network* - Rede Totalmente Convolutiva
HSV	*Hue, Saturation, Value* - Sistema de cores
Lab	Sistema de cores
RGB	*Red, Green, Blue* - Sistema de cores
RNA	Rede Neural Artificial
VGG	*Visual Geometry Group*

LISTA DE SÍMBOLOS

CA	*Class accuracy*
fn	*False negative* - Falso negativo
fp	*False positive* - Falso positivo
GA	*Global accuracy*
IoU	*Intersection over Union*
P	*Precision*
R	*Recall*
tp	*True positive* - Verdadeiro positivo

SUMÁRIO

1 INTRODUÇÃO

A alta demanda global por alimentos exige do ambiente rural larga produção agrícola. O avanço da tecnologia tem auxiliado no sentido de aumentar e otimizar a agroindústria. O processo de mecanização do campo ocorrido nas últimas décadas é exemplo do esforço para produção em larga escala. No entanto, são vários os problemas a serem abordados para que essa produção seja ecologicamente sustentável.

O surgimento de plantas invasoras é considerado natural em ambiente rural. Entretanto, é necessário agir rapidamente para erradicar sua presença antes que as ervas daninhas consumam os recursos provenientes da plantação, prevenindo a deterioração das plantas cultivadas (HAMUDA; GLAVIN; JONES, 2016). Geralmente, plantas invasoras ou ervas daninhas são espécies que nascem sem assistência humana, isto é, elas crescem esporadicamente no solo e começam a competir por luz e nutrientes, dificultando o desenvolvimento do cultivo. Além disso, as ervas daninhas se espalham rapidamente e em grande quantidade, competindo por espaço de outras plantas (TANG et al., 2016). Ervas daninhas podem representar uma perda de até 70% da plantação original, dependendo de algumas condições, tais como clima, tipo de solo e grau de infestação (DYRMANN; KARSTOFT; MIDTIBY, 2016). Elas podem trazer perdas significativas na produção global de alimentos, que podem ser maiores do que os danos causados por animais nocivos (artrópodes, nematoides, roedores, pássaros, lesmas e caramujos), patógenos (fungos e bactérias) e vírus (MEULEN; CHAUHAN, 2017).

O método mais utilizado para remoção de ervas daninhas em grandes áreas é o controle químico, que consiste em pulverização de herbicidas por toda a área para assegurar que todas as ervas daninhas sejam retiradas (CORDEAU et al., 2016). A seleção de um herbicida deve ser baseada na espécie de plantas presente na área a ser tratada, assim como nas propriedades físico-químicas dos produtos. Entretanto, métodos tradicionais de pulverização causam grande desperdício destes químicos porque não há qualquer tipo de seletividade no momento da aplicação (AHMAD et al., 2018), o que é economicamente e ecologicamente danoso. Uso frequente e incorreto de herbicidas podem causar contaminação de solos e, consequentemente, de águas superficiais e de subsolos, assim como trazer riscos aos trabalhadores em contato com estes componentes químicos (MAJEED, 2018). Por esta razão, métodos mais inteligentes de pulverização devem ser desenvolvidos e aplicados em áreas rurais.

Métodos computacionais para identificação de plantas são importantes para sistemas embarcados em drones e veículos terrestres para pulverização localizada. Algoritmos para separação entre plantas e solo tem sido muito estudados e são utilizados por alguns agricultores na primeira fase do cultivo, fase esta que consiste em remover todas as ervas daninhas antes de começar a plantação. Um desses métodos é o *Excess Green* (MEYER; NETO, 2008), no qual a detecção da planta é baseada na cor verde caracte-

rística da clorofila, porém é eficiente apenas para separação entre solo e vegetação. No entanto, quando as plantas estão crescendo, ervas daninhas crescem juntas e prejudicam a produção. Neste passo, um método para discriminação entre espécies é necessário. Diversos estudos tem sido feitos para desenvolver técnicas para automaticamente diferenciar erva daninha da plantação, com o intuito de realizar pulverização localizada de herbicidas. Umas dessas técnicas se baseia no cálculo de índices de vegetação, como o NDVI (CARLSON; RIPLEY, 1997), que faz uso da refletância de comprimentos de onda na faixa entre luz vermelha e infravermelha. Outras técnicas consistem de análise de imagens onde a cor e o formato são levados em consideração para a detecção de ervas daninhas (BOSILJ; DUCKETT; CIELNIAK, 2018). Métodos mais recentes utilizam inteligência artificial, onde uma rede neural é treinada para identificar certos tipos de espécies (TANG et al., 2017). Entretanto, há diversos métodos diferentes de inteligência artificial a serem estudados e aplicados para detecção de ervas daninhas. Cada um desses métodos tem vantagens e desvantagens com respeito a precisão e desempenho computacional. Assim sendo, duas diferentes técnicas de aprendizado de máquina serão exploradas nesse trabalho: *Fully Convolutional Network* (FCN) (LONG; SHELHAMER; DARRELL, 2015) e SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2017).

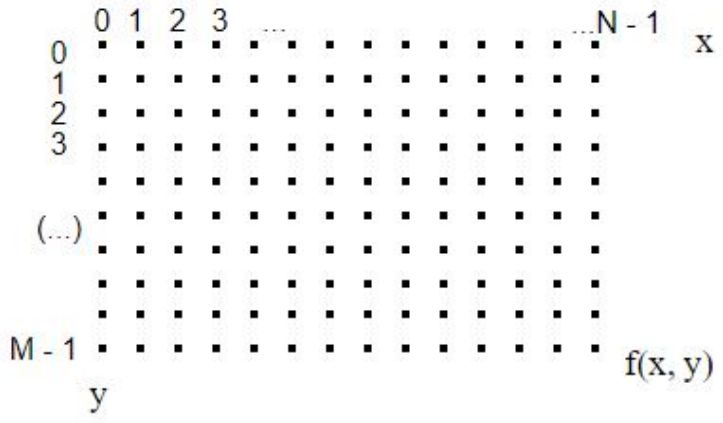
O principal objetivo do trabalho é a avaliação de algoritmos de aprendizado profundo de máquina para segmentação de imagem aplicada à distinção entre ervas daninhas e plantas de cultivo utilizando um banco de dados aberto de imagens de uma plantação de cenouras. O banco de dados é apresentado em (LAMESKI et al., 2017) e inclui etiquetas (*labels*) para a avaliação do desempenho de segmentação do algoritmo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Imagens como Matrizes

Em processamento digital de imagem, a imagem é definida como uma função de duas variáveis $f(x, y)$, em que x e y são as coordenadas espaciais, definidas em um plano discreto. A amplitude de f é proporcional a intensidade luminosa, também uma quantidade discreta. Cada par (x, y) é chamado de *pixel*. Os valores máximos que x e y podem tomar são as dimensões da imagem, M e N, dados em número de pixels. A Figura 1 mostra uma representação geral de uma imagem no plano (x, y). Cada ponto da Figura 1 representa um pixel.

Figura 1: Representação de uma imagem no plano (x, y).



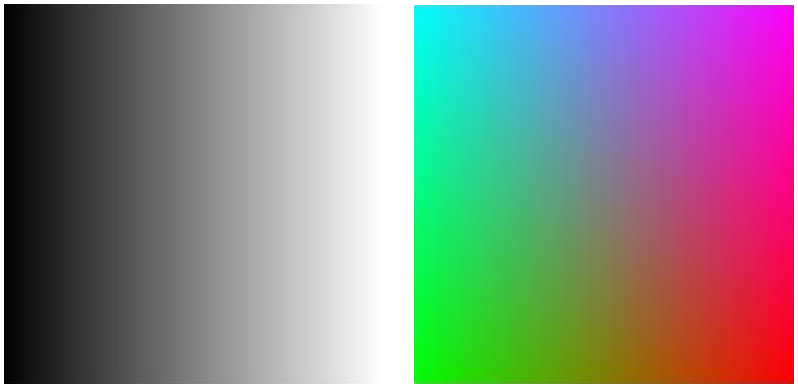
A função bidimensional $f(x, y)$ pode ser presentada também na forma matricial:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(1,0) & \cdots & f(N-1,0) \\ f(0,1) & f(1,1) & \cdots & f(N-1,1) \\ \vdots & \vdots & \ddots & \vdots \\ f(0,M-1) & f(1,M-1) & \cdots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

Cada pixel pode tomar um valor inteiro positivo entre 0 e $L-1$, sendo L o número de diferentes tons de luminosidade. Neste trabalho, apenas imagens com $L = 256$ foram utilizadas, de modo que cada pixel de uma imagem monocromática seja representado por um *byte*. A Figura 3(a) mostra uma imagem de uma função $f(x, y) = x$, sendo $M = N = 256$.

Uma representação possível para uma imagem colorida é o sistema RGB. Nesse sistema, uma imagem é representada por três matrizes, sendo cada matriz referente às cores primárias: vermelho, verde e azul. A sobreposição das três matrizes gera a imagem colorida. A maioria dos televisores e monitores funciona com esse sistema (TRUSSELL; SABER; VRHEL, 2005). Para o caso da imagem colorida, a saída de $f(x, y)$ não é mais um escalar que diz respeito à intensidade luminosa monocromática, mas sim um vetor de três elementos: $[R, G, B]$, sendo R a intensidade luminosa da cor vermelha, G a intensidade luminosa da cor verde e B a intensidade luminosa da cor azul. A Figura 3(b) mostra uma imagem em que $f_r(x, y) = x$, $f_g(x, y) = 255 - x$ e $f_b(x, y) = 255 - y$, sendo f_r, f_g e f_b são as funções que descrevem a intensidade luminosa das cores vermelha, verde e azul, respectivamente; e $M = N = 256$.

Figura 2: Exemplo de imagens.



(a) Imagem monocromática (b) Imagem colorida

2.2 Segmentação de Imagem

A segmentação de imagens subdivide uma imagem em suas partes ou objetos constituintes. O nível de subdivisão a ser realizado depende da tarefa a ser realizada. Em outras palavras, a segmentação termina quando os objetos de interesse tiverem sido devidamente destacados. Para dar um exemplo, em aplicações de aquisição aérea de alvos terrestres, o objetivo é, entre outras coisas, a identificação dos elementos constituintes da estrada em objetos que possuam tamanho pertencente a uma escala de tamanhos correspondentes a potenciais veículos. Não há razão pela qual realizar a segmentação da imagem abaixo dessa escala, assim como não há por que segmentar objetos fora dos limites da estrada (GONZALEZ; WOODS, 2000, p. 295). Importante destacar que cada pixel da imagem constitui uma unidade básica que nunca pertence a mais de um objeto ou parte, de modo que as regiões em que se divide a imagem não se sobrepõem.

Os métodos de segmentação de imagens monocromáticas geralmente são baseados numa das seguintes propriedades básicas: descontinuidade e similaridade. Na primeira categoria, a abordagem é dividir a imagem de acordo com as mudanças bruscas nos níveis de cor. Nesse quesito, objetivos comuns são detecção de pontos isolados e detecção de linhas e bordas na imagem. Com relação à segunda categoria, os métodos são baseados em limiarização, crescimento de regiões e divisão e fusão de regiões (GONZALEZ; WOODS, 2000, p. 295).

Além dos métodos monocromáticos, existem os métodos de segmentação baseados

em cores, nos quais os objetos a serem segmentados tem como característica essencial a sua cor. Existem também métodos treináveis, em que redes neurais artificiais são responsáveis por realizar a segmentação a partir de um aprendizado autônomo dos padrões dos objetos a serem segmentados; padrões estes que são aprendidos pela RNA a partir de um banco de dados suficientemente vasto.

2.2.1 Detecção de Verde

Um dos desafios abordados por processamento digital de imagem é o de detecção automática de verde com o intuito de separar as plantas do solo para aplicações de agricultura de precisão. Detecção de verde é um tipo específico de segmentação de imagem, baseado unicamente na intensidade das cores. Diversos índices de vegetação foram desenvolvidos da década de 1990 para cá. Em (WOEBBECKE et al., 1995) os autores testaram 5 diferentes índices de cores de vegetação utilizando coordenadas cromáticas e matiz modificado (*modified hue*) para distinguir plantas vivas do solo e de resíduos secos de outras plantas. Tais índices são (sem indicar as coordenadas x e y):

$$r - g,\ g - b,\ \frac{g-b}{r-g} \text{ e } 2g - r - b \tag{2}$$

onde r, g e b são as coordenadas cromáticas:

$$r = \frac{R^*}{R^* + G^* + B^*},\ g = \frac{G^*}{R^* + G^* + B^*} \text{ e } b = \frac{B^*}{R^* + G^* + B^*} \tag{3}$$

e R^*, G^* e B^* são os valores RGB normalizados, variando de 0 a 1, definidos como:

$$R^* = \frac{R}{R_m},\ G^* = \frac{G}{G_m} \text{ e } B^* = \frac{B}{B_m}, \tag{4}$$

onde R, G e B são os valores de cada pixel para cada canal RGB; e $R_m = G_m = B_m = 255$, valor máximo para um pixel. Matiz modificado se adquire dos canais RGB:

$$Matiz = cos^{-1}\frac{2R - G - B}{2[R^2 + G^2 + B^2 - RG - GB - RB]^{1/2}}. \tag{5}$$

Em (WOEBBECKE et al., 1995) o autor conclui que o índice de vegetação de excesso de verde ($ExG = 2g - r - b$) fornece uma imagem monocromática quase binária, onde fica destacada uma região de interesse correspondente às plantas verdes.

Outro exemplo de índice é o NDI, proposto por (PEREZ et al., 2000). O NDI é a diferença normalizada entre os canais verde e vermelho:

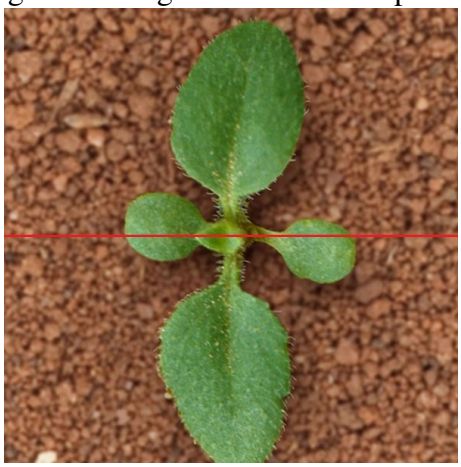
$$NDI = \frac{G - R}{G + R}. \tag{6}$$

Este índice fornece valores que variam de -1 a +1. No entanto, para exibir a imagem, cujos valores de pixel variam entre 0 e 255, ao NDI anterior é somado 1, e então é multiplicado por 128 para fornecer um intervalo de 256 níveis de cinza. À imagem resultante desses índices, geralmente se aplica o método de Otsu (OTSU, 1979) para realizar a binarização da imagem.

Em (MEYER; NETO, 2008), os autores propõem o índice de excesso de verde menos excesso de vermelho:

$$ExGExR = ExG - ExR, \tag{7}$$

Figura 3: Imagem RGB de uma planta.



sendo $ExR = 1,4r - b$, utilizando o valor 0 para a binarização da imagem. Os autores compararam o desempenho dos três métodos (ExG + Otsu, NDI + Otsu e ExGExR) e relataram um melhor desempenho utilizando ExGExR dentro dos casos estudados.

A Figura 3 mostra uma planta verde utilizada como referência. As figuras 5(a), 5(b) e 5(c) mostram a planta da Figura 3 sob o efeitos dos índices NDI, ExG e ExGExR, respectivamente. As figuras 5(d), 5(e) e 5(f) mostram as imagens após a binarização. O branco em destaque em cada imagem é a região de interesse relativa à planta. A Figura 5 mostra os histogramas dos índices relativos à linha vermelha na Figura 3. Para efeito de comparação, os índices ExG e ExGExR foram multiplicados por 255. É interessante notar o "degrau"contido entre $x = 100$ e $x = 500$, onde a linha vermelha atravessa o verde da planta. O objetivo aqui não é comparar o desempenho dos índices, e sim apresentar o efeito de cada um deles para facilitar o entendimento de cada método.

2.3 Redes Neurais

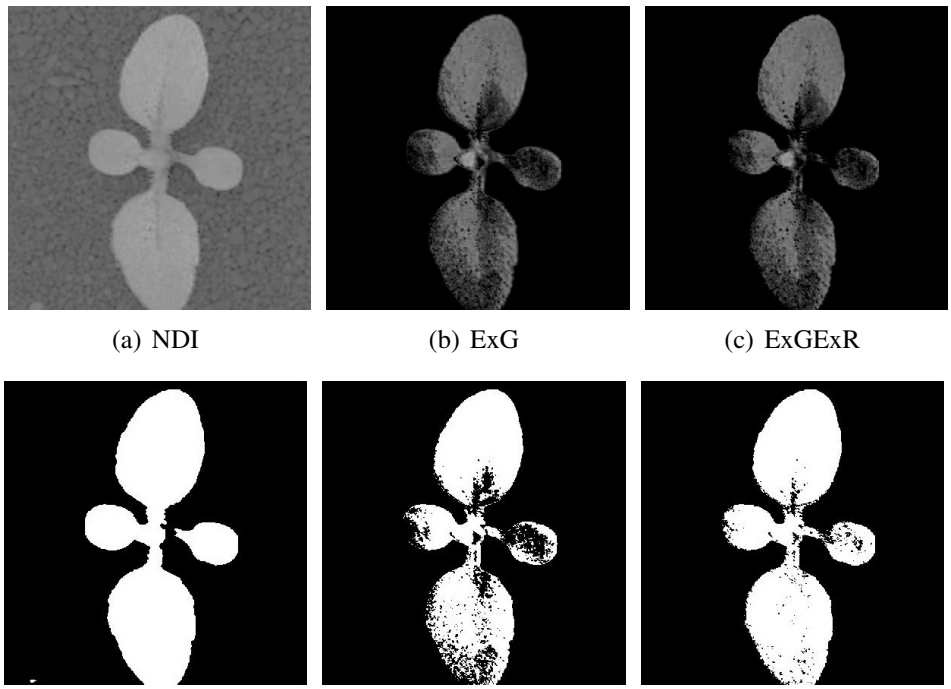
As redes neurais artificiais, usualmente chamadas apenas de "redes neurais", começaram a serem estudadas na década de 40 do século XX tendo como inspiração o funcionamento do cérebro humano ou animal. Entende-se o cérebro como um computador, tendo em vista que ele é capaz de processar informações de alta complexidade, de maneira não-linear e em paralelo (HAYKIN, 2001, p. 27).

O cérebro possui diversos mecanismos para reconhecer o ambiente externo, como a visão, a audição, etc. A partir desses estímulos externos, ele é capaz de processar informações e armazená-las para uso imediato ou futuro. Como um exemplo, pode-se considerar o sonar do morcego. Além de fornecer a distância do alvo, o sonar de um morcego fornece informações de velocidade, tamanho, entre outros. De posse dessas informações, o morcego pode montar a melhor estratégia para atacar a sua presa. O que torna isso possível é que, desde o início da vida, o cérebro recebe estímulos externos, além de interagir com o ambiente. Através do acúmulo de experiência, de estímulos e da resposta externa às interações, o cérebro é capaz do que pode-se chamar de aprendizado.

Diversas são as capacidades do cérebro que tornam a simulação por uma rede neural artificial algo atrativo, além da capacidade de aprendizagem (RAUBER, 2005):

- *robustez e tolerância a falhas*: a exclusão ou o mau funcionamento de alguns neurô-

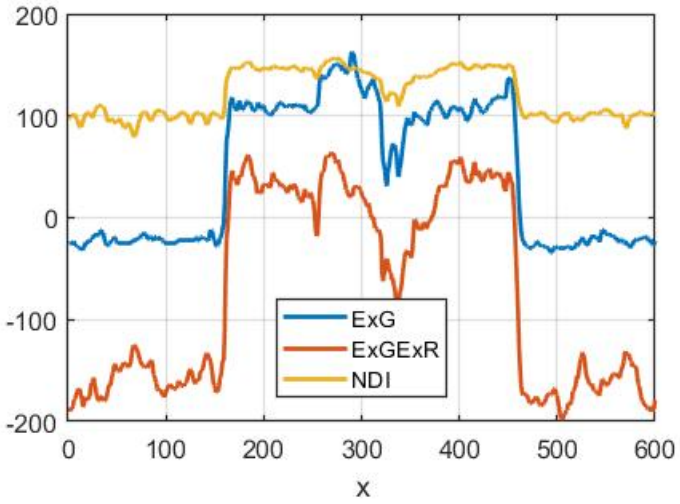
Figura 4: Comparação entre os diferentes índices e as binarizações resultantes.



(a) NDI (b) ExG (c) ExGExR

(d) NDI + Otsu (e) ExG + Otsu (f) ExGExR + *Threshold* positivo

Figura 5: Comparação entre os histogramas dos índices.



nios não interfere no seu funcionamento global,

- *processamento de informação incerta*: mesmo que a informação processada esteja incompleta ou afetada por ruído, um raciocínio considerado correto ainda é possível,
- *paralelismo*: um enorme número de neurônios funcionam ao mesmo tempo, de modo que diversas informações são processadas ao mesmo tempo.

De um modo geral, uma rede neural é uma máquina projetada para modelar como

o cérebro realiza uma determinada tarefa. Normalmente, a rede neural é implementada utilizando-se componentes eletrônicos, ou programada em um computador digital. Objetivamente, ela se assemelha ao cérebro em dois aspectos (HAYKIN, 2001, p. 28): o conhecimento é adquirido e armazenado através de um processo de aprendizagem, e as forças de conexão entre os neurônios, chamadas de pesos sinápticos, são utilizadas para armazenar o conhecimento.

Nesse capítulo pretende-se colocar de maneira concisa, objetiva e suficiente os principais conceitos para a compreensão do trabalho proposto.

2.3.1 *Deep Learning*

Deep Learning, ou "Aprendizagem Profunda", é um tipo de aprendizagem de máquina e se baseia na tecnologia de Redes Neurais Artificiais de múltiplas camadas. Aprendizagem profunda foi desenvolvida no sentido de atacar complexos problemas, tais como reconhecimento de imagem e voz.

O primeiro trabalho mais importante na área de *deep learning* foi o algoritmo de Ivakhnenko e Lapa (IVAKHNENKO; LAPA, 1965). O modelo combinava várias camadas neurais com características não-lineares, incluindo funções de ativação polinomiais, analisadas por eles com métodos estatísticos. Importante destacar que esse modelo ainda não incluía o uso do algoritmo de retropropagação: os pesos das redes eram calculados pelo método de mínimos quadrados, e o cálculo para cada camada era realizado isoladamente.

As primeiras redes convolutivas foram utilizadas por Fukushima (FUKUSHIMA, 1979). As redes de Fukushima tinham múltiplas camadas convolutivas e de subamostragem, tais como as redes mais recentes, mas a rede foi treinada fazendo uso de aprendizagem por reforço. Além disso, era necessário atribuir manualmente características de cada imagem aumentando o peso de certas conexões.

Até este ponto, algoritmo de *backpropagation* ainda não havia sido utilizado para redes neurais. O primeiro verdadeiro uso de *backpropagation* foi em (LECUN et al., 1989). LeCun utilizou redes convolutivas com algoritmo de retropropagação para classificação de dígitos escritos à mão.

Por cerca de duas décadas poucos trabalhos foram realizados utilizando redes neurais. Os avanços mais significativos em inteligência artificial nesse período aconteceram com o uso de máquinas de vetor de suporte (SVM), com destaque para (CORTES; VAPNIK, 1995). Novos avanços em *deep learning* só vieram a ocorrer quando os computadores se tornaram máquinas mais rápidas, em particular com a introdução das unidades de produção gráfica (GPUs). A partir desse ponto, importantes trabalhos com redes neurais foram realizados, vide Seção 3. Embora redes neurais profundas são mais lentas quando comparadas às SVMs, elas tem desempenho melhor com a mesma quantidade de dados de treinamento.

Nas próximas seções serão expostos alguns conceitos básicos de redes convolutivas, o tipo de rede neural profunda utilizada neste trabalho.

2.3.1.1 Redes Convolutivas

Redes convolutivas tem sido bastante explorada em trabalhos envolvendo reconhecimento de padrões por imagem (DYRMANN; KARSTOFT; MIDTIBY, 2016; KAMILARIS; PRENAFETA-BOLDÚ, 2018; MILIOTO; LOTTES; STACHNISS, 2017; TANG et al., 2017; VOULODIMOS et al., 2018) e constituem base para diversas arquiteturas que foram desenvolvidas nos últimos anos com a mesma finalidade (BADRINARAYANAN;

KENDALL; CIPOLLA, 2017; CHEN et al., 2018; KRIZHEVSKY; SUTSKEVER; HINTON, 2012; LONG; SHELHAMER; DARRELL, 2015; HE et al., 2017).

Uma desvantagem do uso de redes multi-camadas para reconhecimento de padrões em imagens ou fala (objetos multidimensionais) é a alta demanda computacional. Algumas vezes, uma imagem chega a possuir alguns milhões de pixels. Apenas com uma primeira camada totalmente conectada com 100 neurônios ocultos já haveria 10.000 sinapses. Problemas de sobre-ajuste podem ocorrer caso a quantidade de dados para treinamento seja escassa. Um modelo tem problema de sobre-ajuste quando ele se ajusta muito bem a um conjunto de dados já observado, mas é ineficaz para prever novos resultados. No entanto, a maior desvantagem da rede multi-camadas para aplicações em imagem ou fala é que ela não possui invariância intrínseca quanto a translações e distorções locais das entradas (LECUN; BENGIO et al., 1995). Isso significa que o aprendizado de características para esse tipo de rede é localizado, de modo que o aprendizado realizado numa área da rede não é generalizado para o resto da rede devido a não existência de compartilhamento de pesos.

A rede convolutiva é uma rede de múltiplas camadas projetada para reconhecer formas bidimensionais com alto grau de invariância a translação, escalamento, inclinação e outras formas de distorção (HAYKIN, 2001, p. 271). A rede convolutiva possui as seguintes formas de restrições (LECUN; BENGIO et al., 1995),(HAYKIN, 2001, p. 271):

1. *Extração de características*. Cada neurônio recebe seus sinais de entrada de um campo receptivo local na camada anterior, o que mantém extraindo características locais. Uma vez que a característica local seja extraída, sua localização exata já não importa tanto, desde que sua posição relativa às outras características seja preservada.

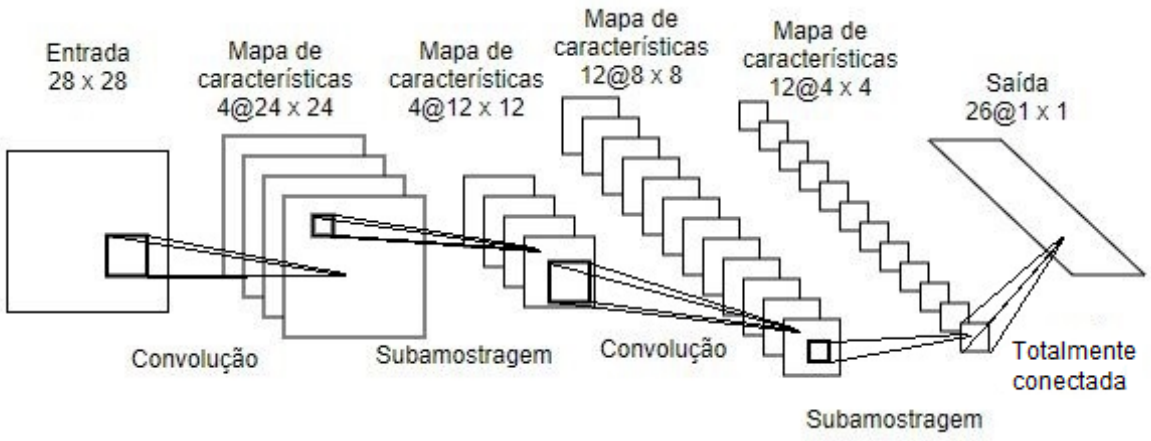
2. *Mapeamento de características*. Cada camada é composta de vários mapas de características, sendo cada um deles um plano em que os neurônios individuais compartilham o mesmo conjunto de pesos sinápticos, o que produz as seguintes vantagens:

    - *Invariância a deslocamento*, introduzida na operação de um mapa de características através do uso de uma operação de convolução com um núcleo (*kernel*) de tamanho pequeno, seguida por uma função ativadora.
    - *Redução do número de parâmetros livres*, devido ao compartilhamento de pesos sinápticos

3. *Subamostragem*. Após cada camada convolutiva, uma camada calcula a média local e realiza uma subamostragem, reduzindo a resolução do mapa de características. Isso faz com que haja uma diminuição da sensibilidade da saída do mapa frente a deslocamentos e outras distorções.

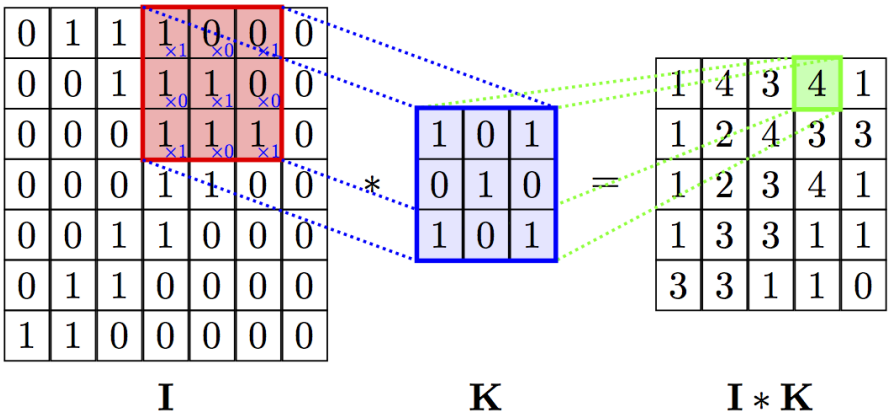
Todos os pesos de todas as camadas da rede são ensinados a rede por aprendizagem supervisionada, treinados por retropropagação. A extração de características da rede neural é realizada automaticamente (HAYKIN, 2001, p. 272). A Figura 6 mostra uma topologia de rede neural convolutiva para processamento de imagem de caractere manuscrito. A rede possui uma camada de entrada, quatro camadas ocultas e uma camada de saída. A camada de entrada recebe a imagem monocromática, já centrada e normalizada em tamanho, contendo um único caractere. Após isso, as operações computacionais se alternam entre convolução e subamostragem, como descrito a seguir:

Figura 6: Rede convolutiva para reconhecimento de caractere manuscrito por imagem.



Adaptado de (LECUN; BENGIO et al., 1995).

Figura 7: Exemplo de convolução bidimensional.



Retirado de (VELIKOVI, 2017).

A operação de convolução bidimensional consiste de multiplicar e somar os elementos respectivos de uma matriz bidimensional a outra, como mostra o exemplo da Figura 7. O valor em verde na Figura 7 é o resultado da soma das multiplicações entre os valores do *kernel* **K** com os valores em vermelho da matriz **I**. Os valores em vermelho são o campo receptivo do neurônio verde. A matriz **I*K** é o resultado da convolução entre **I** e **K**.

- A primeira camada oculta realiza convolução com quatro *kernels*. Resulta em quatro mapas de características, cada mapa contendo 24 x 24 neurônios. Cada neurônio recebe um campo de tamanho 5 x 5, dimensão dos *kernels*.

- A segunda camada oculta realiza subamostragem e calcula a média local. Possui quatro mapas de características de dimensão 12 x 12, devido a subamostragem de proporção 2 para 1. Cada neurônio tem um campo receptivo de 2 x 2, um coeficiente treinável, um bias treinável e uma função de ativação sigmoide.

- A terceira camada oculta faz outra convolução, dessa vez com 3 *kernels* de 5 x 5,

resultando em 12 mapas de características com 8 x 8 neurônios. Cada neurônio nesta camada pode ter conexões sinápticas com vários mapas da camada anterior.

- Quarta camada oculta realiza segunda subamostragem e cálculo da média local. 12 mapas de características de 4 x 4 neurônios. Opera similarmente à primeira camada de subamostragem.

- A camada de saída é do tipo totalmente conectada, logo não há compartilhamento de pesos. 26 neurônios de saída, sendo que cada neurônio é relacionado a um caractere entre 26 possíveis. Cada neurônio é atribuído a um campo receptivo de 4 x 4 neurônios, do tamanho dos mapas de características da camada anterior.

Com as sucessivas operações de convolução e subamostragem, obtém-se um aumento do número de mapas de características e uma diminuição da resolução em relação à camada anterior. A rede neural da Figura 6 contém aproximadamente 100.000 conexões sinápticas, mas apenas 2.600 parâmetros livres. Essa redução de parâmetros acontece devido ao compartilhamento de pesos (HAYKIN, 2001, p. 273).

2.3.1.2 AlexNet

A arquitetura AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) foi desenvolvida com o objetivo de abordar o problema de classificação de imagens. O banco de dados utilizados pelos autores para validação da arquitetura consiste de imagens pertencentes a 1000 classes possíveis. A entrada da rede é uma imagem RGB de dimensão 256 $\times$ 256. A saída é um vetor em que o enésimo elemento é interpretado como a probabilidade de que a imagem de entrada pertence à enésima classe.

A AlexNet consideravelmente maior que as redes convolutivas anteriores, como por exemplo a proposta em (LECUN et al., 1998). Possui cerca de 60 milhões de parâmetros e 650 mil neurônios e levou mais de 5 dias para ser treinada em duas GPUs GTX 580 de 3 GB em processamento paralelo.

(inserir figura)

A rede consiste de 5 camadas convolutivas e 3 camadas totalmente conectadas. Os *kernels*, ou filtros, extraem as características da imagem. Numa única camada convolutiva há muitos filtros de mesmas dimensões. A primeira camada possui 96 filtros de $11\times11\times3$, sendo 11×11 a dimensão do filtro para a operação de convolução bidimensional e 3 o número de canais das imagens de entrada (RGB).

As duas primeiras camadas convolutivas são seguidas de camadas sobrepostas de *max pooling*. *Max pooling* é a operação de subamostrar uma matriz, utilizando o valor mais alto da região de amostragem. Quando as regiões sobre as quais se faz subamostragem possuem intersecções, diz-se que são camadas sobrepostas de *max pooling* (*overlapping max pooling layers*). Esse tipo de subamostragem ajudou a reduzir erros de tipo *top-1* (calculado em cima do percentual de acertos em que a classe predita é igual à classe real do objeto analisado pelo classificador) e de tipo *top-5* (calculado checando se a classe real do objeto analisado está entre as 5 classes mais prováveis apontadas pelo classificador) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

(inserir figura sobre max pooling)

Não linearidades do tipo ReLU são aplicadas na saída de todas as camadas convolutivas e das camadas totalmente conectadas. ReLU (*Rectified Linear Unit*) é um operador não linear, em que a saída é igual à entrada para valores iguais ou maiores que zero e igual a zero para valores negativos. Os autores introduziram o uso de ReLU como função

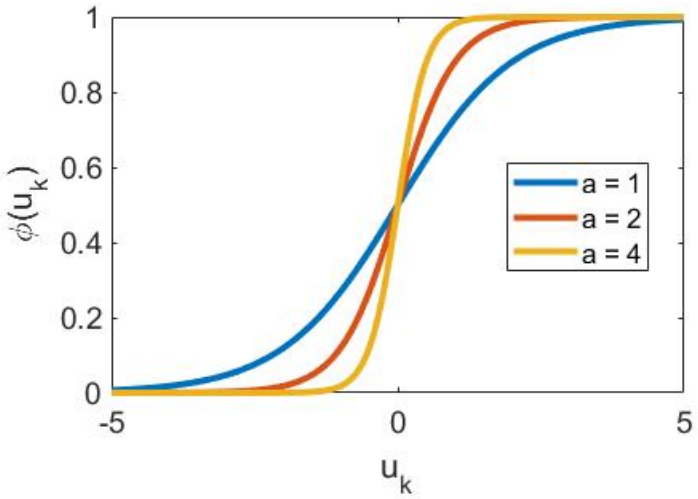
de ativação. Até então, era mais comum o uso de funções sigmoidais e tangentes hiperbólicas. A vantagem do uso da ReLU é que, para redes muito profundas, o tempo de treinamento cai drasticamente.

A função sigmoide, por exemplo, satura para valores de entrada muito altos ou muito baixos. A função sigmoide é definida por:

$$\phi(u_k) = \frac{1}{1 + exp(-au_k)} \tag{8}$$

onde a é o parâmetro de inclinação da função sigmoide. Em contrapartida, a inclinação da função ReLU nunca se aproxima a zero para altos valores de entrada. Isso faz com que a convergência seja mais rápida. Para valores negativos de entrada a inclinação é zero, porém o mais comum é que a grande maioria dos neurônios acabem possuindo valores positivos.

Figura 8: Função sigmoide para $a = 1$, $a = 2$ e $a = 4$.



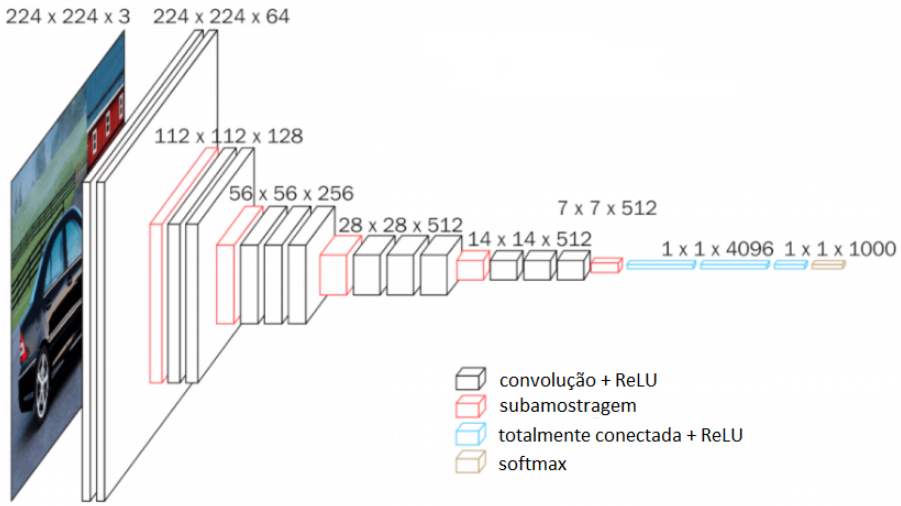
Para redução atacar o problema de sobreajuste, em que a rede treinada demonstra alto desempenho de predição no conjunto de treinamento e baixo desempenho com o conjunto de teste, os autores utilizaram a técnica de *dropout*, apresentada em (HINTON et al., 2012). *Dropout* consiste em desligar um ou mais neurônios da rede, de modo que eles não contribuem com a propagação da informação ao longo da rede. Isso faz com que os parâmetros treinados sejam mais robustos e não sofram de sobrejuste tão facilmente.

2.3.1.3 VGGNet

A VGGNet (SIMONYAN; ZISSERMAN, 2014) consiste de 16 camadas convolutivas e é caracterizada pela sua arquitetura uniforme. A VGGNet realiza uma melhoria em relação à AlexNet ao substituir *kernels* de dimensões 11×11 e 5×5 nas primeira e segunda camadas, respectivamente, por múltiplos *kernels* de dimensões 3×3 sucessivos.

A entrada da rede é uma imagem RGB de dimensões 224×224. A imagem passa por uma série de camadas convolutivas sucessivas seguidas de operações de *max pooling*, conforme a figura acima. Três camadas totalmente conectadas seguem a rede, a primeira e segunda camadas de 4096 canais, a terceira de 1000 canais (um para cada classe). A camada final é uma *softmax*. Todas as camadas ocultas possuem ReLU.

Figura 9: Arquitetura da VGG-16.



Retirado de (HASSAN, 2018).

A VGGNet possui diversas variações na sua arquitetura com relação ao número de camadas convolutivas. A mais utilizada é a VGG16, retratada na figura. A VGG16 supera consideravelmente modelos antecessores, diminuindo erros de tipo *top-1* e *top-5*. Os resultados conseguidos com a arquitetura demonstram que a profundidade da representação é benéfica para a precisão da classificação e que um aumento do desempenho de classificação pode ser conseguido aumentando-se a profundidade da rede.

2.3.1.4 ResNet

Embora resultados melhores têm sido conseguidos por redes cada vez mais profundas, apenas adicionar sucessivas camadas à uma rede não basta para melhorar o seu desempenho. Redes profundas são mais difíceis de treinar em função do problema de dissipação do gradiente, em que o gradiente é retropropagado por muitas camadas. As sucessivas multiplicações fazem com que o gradiente fique demasiado pequeno, dissipando a informação de treinamento. A Figura 10 apresenta a arquitetura da ResNet.

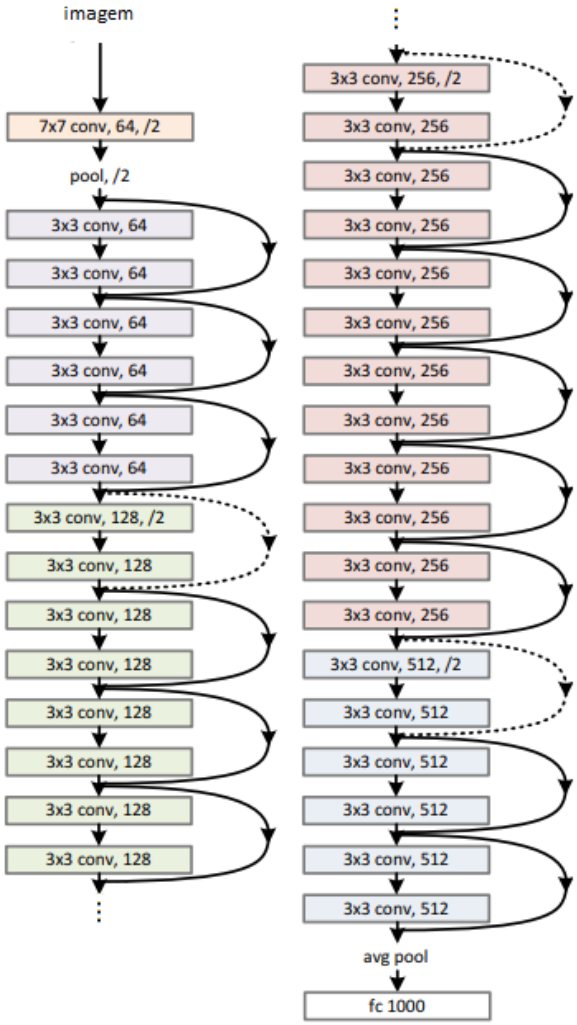
A ResNet (HE et al., 2016) ataca o problema de dissipação de gradiente introduzindo um atalho, ou uma "conexão de salto", mostrada na Figura 11, permitindo que o gradiente seja diretamente retropropagado às camadas anteriores.

Os autores afirmam que, utilizando a conexão de salto, enfileirar sucessivas camadas não degrada o desempenho de uma rede, fazendo que a arquitetura resultante se comporte de maneira similar. Assim, modelos mais profundos ao menos não produzem maiores taxas de erro de treinamento do que suas versões menos profundas. Segundo os autores, a inserção de blocos de conexão de salto permite utilizar diversas camadas para um mapeamento residual, o que é mais eficiente do que utilizar modelos com menos camadas. Com profundidade consideravelmente maior, a ResNet é capaz de aprender características mais complexas, o que a tornou muito popular para aplicações de visão computacional.

2.3.1.5 Redes Convolutivas para Segmentação de Imagem

Atualmente, abordagens utilizando aprendizado profundo, do inglês *deep learning*, tem gerado melhores resultados do que métodos previamente utilizados em tarefas de vi-

Figura 10: Arquitetura ResNet de 34 camadas. As camadas *conv* são convolutivas, *fc 1000* é a camada final totalmente conectada com mil classes de saída.

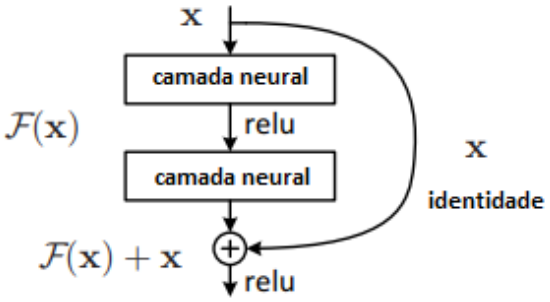


Adaptado de (HE et al., 2016).

são computacional (VOULODIMOS et al., 2018). Redes convolutivas, ou *convolutional neural networks* (CNN), são agora estado-da-arte para tarefas de classificação e detecção de objetos (HE et al., 2017; CHEN et al., 2018). Redes convolutivas são modelos de redes baseadas nas diferentes áreas do córtex visual e suas relações com regiões específicas do campo de visão (LECUN; BENGIO et al., 1995). Essas regiões, denominadas campos receptivos, são responsáveis pela ativação de diferentes neurônios, apresentando um nível de sobreposição entre os campos receptivos dos neurônios próximos. Esse comportamento cerebral inspirou a criação de diferentes sistemas para extrair características específicas dos dados.

As CNNs são compostas por várias camadas que usam a operação de convolução para executar a extração de recursos. Essa operação é realizada a partir de uma janela de dados deslizante, chamada filtro de convolução, que percorre toda a entrada da rede e opera de maneira análoga à sobreposição dos campos receptivos. O modelo pode conter vários filtros, que são ajustados durante o processo de treinamento para obter características distintas da entrada. No final, essas características extraídas tornam-se entrada

Figura 11: Bloco de conexão de salto.



Adaptado de (HE et al., 2016).

de um algoritmo de aprendizagem aplicado à classificação ou regressão, de acordo com o tipo de problema. Comumente as camadas convolucionais realizam outras operações para reduzir o espaço de características (subamostragem), além de normalização e zero-padding. No final das camadas convolucionais, uma sequência de camadas de neurônios conectadas a todas as ativações das camadas anteriores (totalmente conectada, ou *fully-connected*) é usualmente usada, analogamente às camadas das redes neurais tradicionais. Estes modelos convolucionais têm obtido bons resultados em diferentes problemas de visão computacional relacionados à classificação e detecção de imagens (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; HE et al., 2017).

Recentemente, as redes profundas estenderam suas habilidades à segmentação semântica. Redes convolutivas para previsão densa (pixel a pixel) e diferentes arquiteturas, tais como redes totalmente convolucionais – *fully convolutional networks*, (FCN) – Segnet e outras foram propostas para segmentação semântica usando redes convolutivas como ponto base (LONG; SHELHAMER; DARRELL, 2015; BADRINARAYANAN; KENDALL; CIPOLLA, 2017; CHEN et al., 2018). A Figura 12 mostra um exemplo de segmentação de imagem.

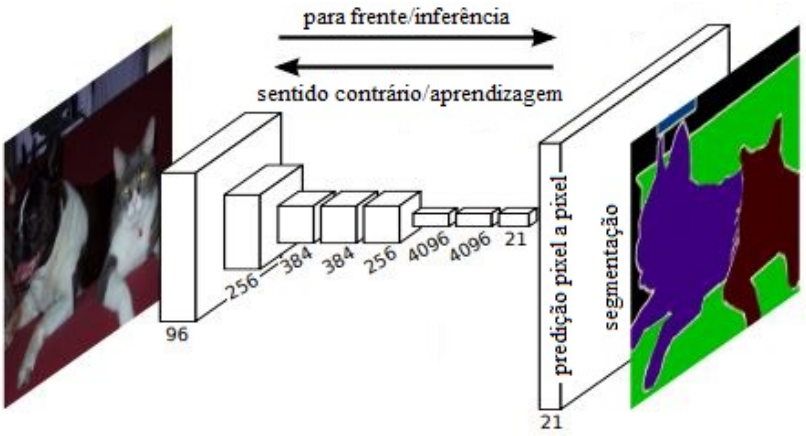
Figura 12: Exemplo de segmentação realizada por RNA treinada. Adaptado de (BADRINARAYANAN; KENDALL; CIPOLLA, 2017).

2.3.1.6 Fully Convolutional Network

Redes convolutivas atingiram grande desempenho para a tarefa de classificação de imagens com (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Em (LONG; SHELHA-

MER; DARRELL, 2015), os autores atacam o problema de segmentação de imagem através da adaptação de arquiteturas de rede que realizam classificação. Assim, a rede realiza uma classificação pixel a pixel ao invés de realizar uma classificação de uma imagem inteira. A Figura 13 mostra um exemplo de como funciona a rede proposta.

Figura 13: Redes totalmente convolutivas podem realizar predições pixel a pixel.



Adaptado de (LONG; SHELHAMER; DARRELL, 2015).

Para que a rede realize segmentação, substitui-se a camada final totalmente conectada por operações convolutivas, como mostra a Figura 14. Ao invés da rede dar como resultado a classe "gato malhado", com essa substituição a rede produz um "mapa de calor"do mesmo tamanho que a imagem de entrada, em que a classificação pixel a pixel produz uma região onde encontra-se o gato malhado. A rede, por possuir apenas camadas convolutivas, se chama rede totalmente convolutiva, ou *fully convolutional network* (FCN). Os autores realizaram adaptações às arquiteturas AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), VGG 16 (SIMONYAN; ZISSERMAN, 2014) e GoogLeNet (SZEGEDY et al., 2015); e realizaram ensaios em que a adaptação de VGG 16 (FCN-VGG16) teve melhor desempenho segundo os critérios utilizados.

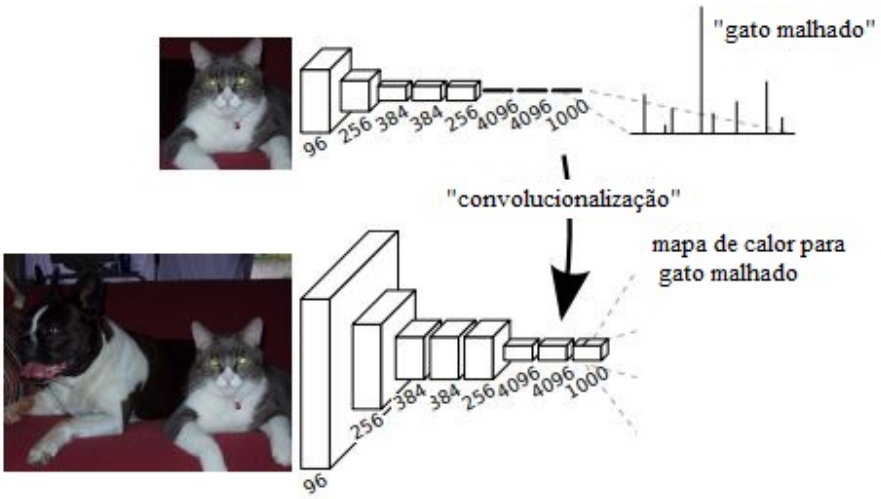
A partir de então, os autores definiram a arquitetura da rede FCN, mostrada na Figura 15. As operações de convolução são mostradas como *conv*, e as operações de subamostragem são descritas como *pool* (diminutivo de *pooling*). A rede combina predições realizadas a partir de uma sobreamostragem final de 32x, 16x e 8x.

As saídas de FCN-32s,-16s e -8s são mostradas na Figura 16. Nota-se a diferença de refinamento entre cada saída. O trecho da rede que vai da imagem até FCN-8s é capaz de produzir segmentação refinada localmente, mas é incapaz de aprender aspectos mais globais. FCN-32s e -16s são mais capazes de aprender características globais que -8s. Somar as diferentes predições permite que a rede faça segmentações com precisão local que respeitem a estrutura global.

2.3.1.7 SegNet

SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2017), assim como FCN, é uma arquitetura de rede que produz segmentação de imagem. SegNet tem uma rede codificadora e uma rede decodificadora correspondente, seguidas por uma camada que realiza classificação pixel a pixel. A Figura 17 mostra a arquitetura de uma rede SegNet. As camadas em azul representam operações de convolução, sucedidas por normalização em lote (*batch normalization*) e uma operação ReLU. As camadas em verde representam

Figura 14: Trocando a camada totalmente conectada pela camada convolutiva permite que a rede produza um "mapa de calor".



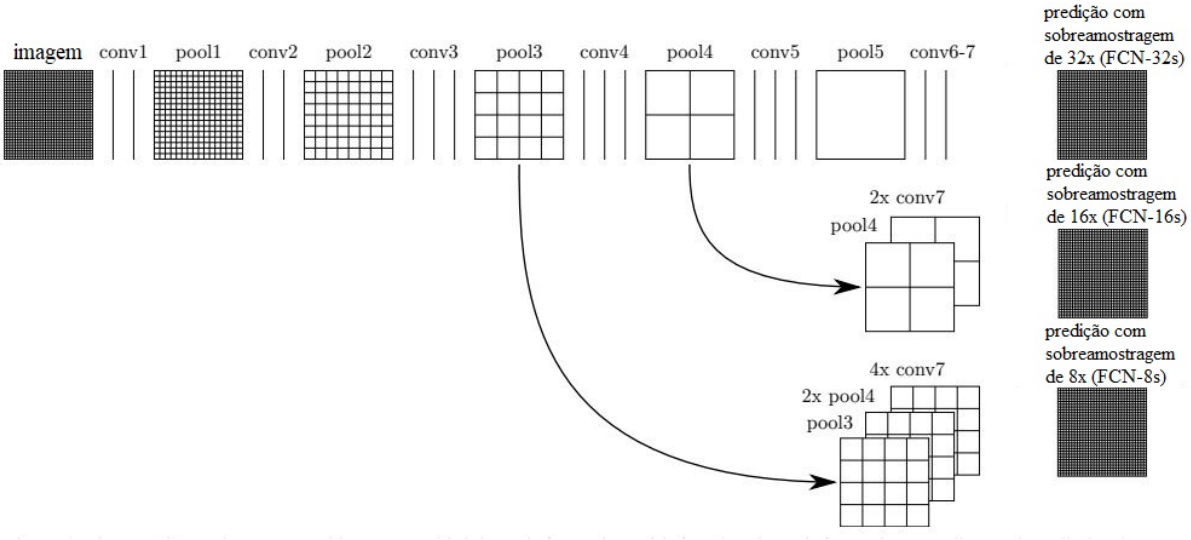
Adaptado de (LONG; SHELHAMER; DARRELL, 2015).

operações de subamostragem, as em vermelho representam sobreamostragens. Assim como FCN, SegNet não possui na ponta final da rete uma camada totalmente conectada, o que diminui drasticamente o número de parâmetros. A rede codificadora consiste de 13 camadas convolutivas, correspondentes às 13 primeiras camadas da rede VGG 16 (SIMONYAN; ZISSERMAN, 2014), projetadas para classificação de objetos. Como cada camada codificadora possui uma camada decodificadora correspondente, logo a rede decodificadora possui 13 camadas. Mapas de características são produzidos pela última camada decodificadora, que alimenta um classificador multi-classe soft-max.

A função softmax é um tipo de função sigmoide, útil para lidar com problemas de classificação. A função sigmoide é capaz de lidar com apenas duas classes. A função softmax transforma as saídas para cada classe para valores entre 0 e 1 e também divide pela soma das saídas. Isso essencialmente dá a probabilidade de a entrada estar em uma determinada classe. É definida como:

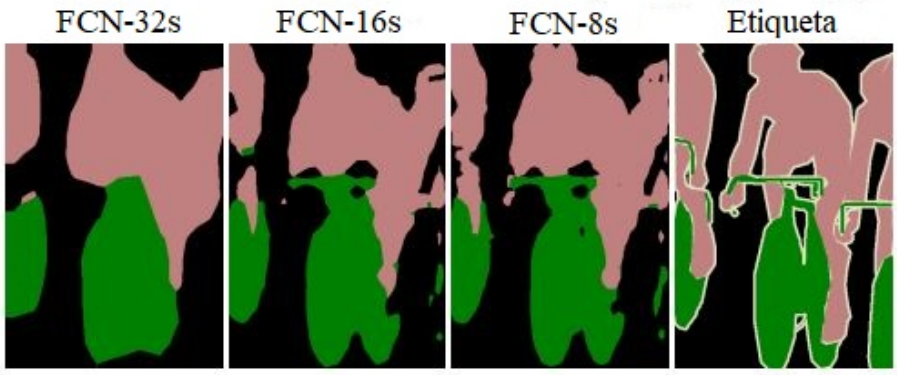
$$\sigma(z_j) = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_k}}, \text{ sendo } j = 1, ..., K. \tag{9}$$

Figura 15: Arquitetura da rede totalmente convolutiva.



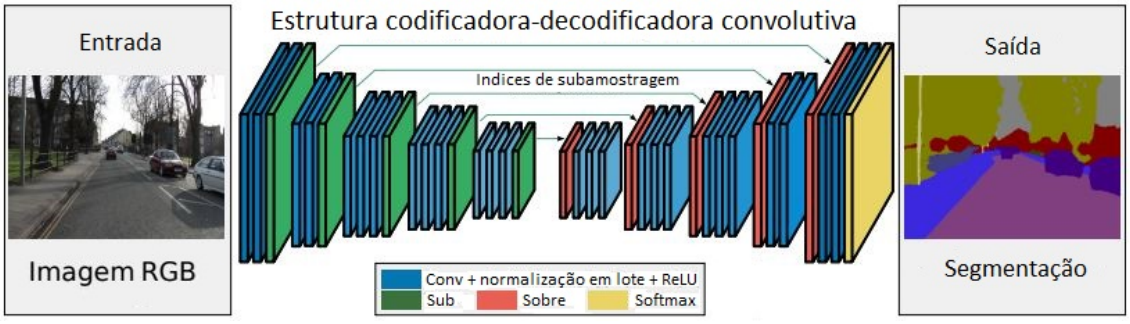
Adaptado de (LONG; SHELHAMER; DARRELL, 2015).

Figura 16: Predições com diferentes níveis de sobreamostragem.



Adaptado de (LONG; SHELHAMER; DARRELL, 2015).

Figura 17: Estrutura da rede SegNet.



Adaptado de (BADRINARAYANAN; KENDALL; CIPOLLA, 2017).

3 ESTADO DA ARTE

Para a elaboração desta dissertação, foi revisada uma série de trabalhos nas áreas de redes neurais e inteligência artificial e na área de agricultura de precisão.

Em 2010, Dan Claudiu Ciresan e Jurgen Schmidhuber realizaram uma das primeiras implementações de redes neurais em GPU (*Graphics Processing Unit*) (CIREŞAN et al., 2010). Essa implementação consiste de uma rede neural de nove camadas projetada para reconhecimento de caracteres escritos a mão.

Em 2012, Alex Krizhevsky publicou a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), uma versão mais profunda e ampla da LeNet e venceu por larga margem a competição conhecida como ImageNet. AlexNet aplicou as características da LeNet numa rede neural muito mais ampla que pode ser utilizada para aprender objetos e hierarquia de objetos muito mais complexos. As principais contribuições da AlexNet foram o uso das unidades lineares retificadas (*Rectified Linear Units* - ReLU) para introdução de não-linearidades e o uso de técnica *dropout* para intencionalmente ignorar alguns neurônios durante o treinamento, uma maneira de evitar sobreajuste. O sucesso da AlexNet deu início a uma pequena revolução: as redes convolutivas tornaram-se aptas a realizar tarefas mais úteis e complexas.

Alguns novos conceitos são propostos na rede Network-in-Network (NiN) (LIN; CHEN; YAN, 2013). Um desses conceitos é a introdução do uso de convoluções com *multilayer perception* (MLP), em que convoluções são realizadas com filtros 1×1, que inserem mais elementos não-lineares ao modelo. Isso ajuda a aumentar a profundidade da rede, que pode ser regularizada com *dropout*. Essa técnica é frequentemente utilizada como gargalo (*bottleneck layer*) de um modelo de *deep learning*. Outro conceito é o uso de *Global Avarage Pooling* (GAP) como alternativa às camadas totalmente conectadas, reduzindo significantemente o número de parâmetros da rede. Aplicando GAP num grande mapa de características, é possível gerar um vetor de características final de baixa dimensão sem reduzir as dimensões dos mapas de características.

Em (SIMONYAN; ZISSERMAN, 2014), os autores propõem a rede VGG (*Visual Geometry Group*). A maior contribuição deste trabalho é que mostra que a profundidade de uma rede é um componente fundamental para atingir maior precisão em reconhecimento ou classificação em redes convolutivas. A arquitetura da VGG consiste de duas camadas convolutivas, ambas com a função de ativação ReLU. Seguindo a função de ativação, tem-se uma camada única de subamostragem (*pooling*) e algumas camadas totalmente conectadas. Estas últimas também seguidas da função ReLU. A camada final do modelo é uma Softmax para classificação.

GoogLeNet (SZEGEDY et al., 2015) foi um modelo proposto pela Google com o objetivo de reduzir a complexidade computacional tradicional das redes convolutivas até então. O método proposto era de introduzir *Inception Layers* que contém vários cam-

pos receptivos, criados por filtros de diferentes tamanhos. Esses campos receptivos dão origem a operações que capturam escassas correlações no novo conjunto de mapas de características. GoogLeNet melhorou o estado da arte da precisão de reconhecimento a partir da inserção das *Inception Layers*. GoogLeNet consistia de 22 camadas no total, consideravelmente maior neste sentido do que suas predecessoras. No entanto, o número de parâmetros de GoogLeNet é muito menor do que seus predecessores AlexNet e VGG. GoogleLeNet tem 7M de parâmetros enquanto AlexNet possui 60M e VGG-19 possui 138M.

ResNet (HE et al., 2016) foi desenvolvida com o intuito de projetar uma rede ultra profunda que não sofresse do problema de desaparecimento de gradiente, presente em seus antecessores. ResNet foi desenvolvida com diversos números de camadas: 34, 50, 101, 152 e, inclusive, 1202. Entre essas, mais utilizada é a de 50 camadas, ResNet50, que possui 49 camadas convolutivas e uma camada totalmente conectada no final. Uma das ideias principais da ResNet foi passar a entrada por duas camadas convolutivas sucessivas e somar a saída dessa operação com a própria entrada. A arquitetura da ResNet permitiu que a rede tivesse centenas, até milhares de camadas sem exigir uma capacidade computacional maior, além de conseguir fornecer uma rica combinação de características.

DenseNet (HUANG et al., 2017), desenvolvida em 2017, consiste de camadas convolutivas densamente conectadas. As saídas de cada camada são conectadas com todas as camadas sucessoras, formando um bloco denso. Esse modelo de rede é eficiente para o reuso de mapas de características, o que reduz bastante o número de parâmetros da rede. A rede DenseNet possui vários blocos densos e blocos de transição, estes últimos são colocados entre dois blocos densos adjacentes. Este modelo apresentou precisão de estado da arte com um número razoável de parâmetros para tarefas de reconhecimento de objetos.

As redes citadas até aqui são redes projetadas para classificação de imagens. Os próximos trabalhos citados são redes feitas para segmentação de imagens.

Em (GIRSHICK, 2015), os autores propõe a rede R-CNN (rede convolutiva baseada em regiões). Os métodos baseados em regiões fazem "segmentação utilizando reconhecimento", o que significa que primeiro se extrai regiões de formas livres de uma imagem e descreve-as, seguido de classificação baseada em regiões. A predições baseadas em regiões são transformadas em predições por pixels, através da classificação do pixel de acordo com a região de maior escore que o contém. A R-CNN primeiro faz uso de uma procura seletiva para extrair uma grande quantidade de propostas de objetos e então realiza operações de CNN. Por último, classifica cada região através de SVMs (*Support Vector Machine*). R-CNN pode ser construída a partir de arquiteturas conhecidas, como AlexNet, VGG, GoogLeNet e ResNet.

A FCN (LONG; SHELHAMER; DARRELL, 2015), *Fully Convolutional Network*, realiza o aprendizado de pixel a pixel, sem extrair regiões. A FCN é também uma extensão das CNNs. A ideia das FCN é fazer com que a rede convolutiva aceite imagens de qualquer dimensão. As CNNs aceitam apenas imagens de tamanho fixo devido a restrição produzida pela camada totalmente conectada. No entanto, FCNs apenas possuem camadas convolutivas e operações de subamostragem.

SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2017) consiste de camadas chamadas de codificadoras e decodificadores (*encoders* e *decoders*). Cada *encoder* aplica convolução, normalização de lote e uma não-linearidade, além de aplicar subamostragem ao resultado. *Decoders* são estruturas similares, a diferença é que não possuem não-linearidades e aplicam sobreamostragem.

DeepLab (CHEN et al., 2018) é uma rede que surgiu como resposta ao problema de redes convolutivas profundas anteriores que apresentavam certa imprecisão para segmentação de objetos. DeepLab é uma combinação de uma rede convolutiva profunda (*deep convolutional neural network*) com uma camada final totalmente conectada de tipo *Conditional Random Field* (CRF). DeepLab consegue produzir segmentação de tal qualidade que melhorou o estado da arte da época.

Na área de agricultura de precisão, os trabalhos a seguir devem ser destacados.

O trabalho realizado em (THOMPSON; STAFFORD; MILLER, 1991) é um dos primeiros a demonstrar interesse em reduzir o uso de herbicida à uma aplicação localizada, diminuindo custos financeiros e ambientais. À época de sua publicação, veículos com detectores não eram viáveis, de modo que o controle em tempo real de ervas daninhas não era possível. Foi mostrado, no entanto, que a aplicação de herbicida espacialmente variável utilizando um sistema de localização de pulverizador e um mapa de campo de ervas daninhas tem potencial. O texto fala sobre o problema de distinção entre planta de colheita e erva daninha, destacando dois tipos de abordagem: por fatores geométricos e por fatores cromáticos. O mapa de dados seria construído a partir de várias técnicas de localização de plantas daninhas com base na análise de imagens, como câmeras de vídeo montadas em tratores, fotografias aéreas e observação manual de campo.

O artigo (PEREZ et al., 2000) trata do desenvolvimento de técnicas de captura e processamento de imagens próximas ao solo para detectar ervas daninhas de folhas largas em plantações de cereais em condições reais de campo. Os métodos propostos utilizam informações de cor para discriminar entre vegetação e fundo, enquanto técnicas de análise de forma são aplicadas para distinguir entre a cultura e as ervas daninhas. O desempenho dos algoritmos foi avaliado comparando os resultados com uma classificação humana, fornecendo uma taxa de sucesso aceitável. O estudo mostrou que, apesar das dificuldades em determinar com precisão o número de mudas (como em pesquisas visuais), é possível usar técnicas de processamento de imagens para estimar a área foliar relativa de plantas daninhas (área foliar/área foliar total da cultura e plantas daninhas) enquanto se desloca pelo campo e usar esses dados em uma procura manual por ervas daninhas em campo.

Em (VRINDTS; DE BAERDEMAEKER; RAMON, 2002), os autores tem por objetivo estudar a viabilidade de distinção entre cultura e erva daninha através da refletância de luz (desde a visível até luz infravermelha) das folhas das plantas. Espectros de refletância de copas de plantas e ervas daninhas foram utilizados para avaliar as possibilidades de detecção de plantas daninhas com medidas de reflexão em condições de laboratório. A classificação em cultura e ervas daninhas foi possível em testes de laboratório, usando um número limitado de razões de bandas de comprimento de onda. Espectros de culturas e ervas daninhas puderam ser separados com mais de 97% de classificação correta. Mais de 90% dos espectros de culturas e ervas daninhas puderam ser identificados corretamente.

Em (ALCHANATIS et al., 2005) os autores descreveram um sistema para detecção automática e avaliação de ervas daninhas baseado em sensor hiperspectral acústico-óptico e algoritmo de detecção. O algoritmo utilizou propriedades de refletância e de estatística para a detecção. A segmentação da cultura em relação ao solo foi feita utilizando alguns canais do sensor, enquanto a detecção de ervas daninhas foi realizada baseando-se em características de textura, extraídas das imagens segmentadas. O algoritmo foi aplicado a um dataset de imagens de plantas de algodão. Os resultados mostraram boa capacidade de detecção.

Em (OKAMOTO et al., 2007) os autores desenvolveram um método de detecção de ervas daninhas que pudesse ser aplicado ao controle automático de ervas daninhas apli-

cando transformadas de wavelet em informação espectral de imagens. Na etapa inicial deste estudo, a imagem das plantas (cultura e ervas daninhas) foi separada do solo usando distância euclidiana como função discriminante. Na etapa seguinte, a imagem da cultura e das plantas daninhas foram classificadas usando a diferença nas características espectrais das espécies vegetais. Neste processo, as variáveis de classificação foram geradas usando a transformada de wavelet para compressão de dados, redução de ruído e extração de características, e então a análise linear discriminante foi aplicada. Os resultados da validação indicaram que o método de classificação desenvolvido tinha potencial para uso prático.

Em (TELLAECHE et al., 2008) os autores focam no problema de detecção de *Avena sterilis*, uma erva daninha nociva que cresce em plantações de cereais. O método proposto, baseado em visão computacional, determina a quantidade e a distribuição de ervas daninhas nos campos de cultivo e aplica uma estratégia de tomada de decisão para pulverização seletiva. O método consiste em dois estágios: segmentação de imagem e tomada de decisão, sem utilização de IA. O processo de segmentação de imagens extrai células da imagem como unidades de baixo nível. A quantidade e a distribuição de ervas daninhas na célula são mapeadas como atributos de área e estruturais, respectivamente. A partir desses atributos, uma abordagem de tomada de decisão permite decidir se uma determinada célula precisa ser pulverizada.

O artigo (BURGOS-ARTIZZU et al., 2011) apresenta um sistema de visão computacional capaz de discriminar entre plantas de erva daninha e plantas de cultura sob luminosidade não controlada em tempo real. O sistema é composto por duas partes: um rápido processador de imagens capaz de entregar resultados em tempo real (*Fast Image Processing*) e um processador mais lento e preciso (*Robust Crop Row Detection*) utilizado para corrigir os erros do primeiro. Esse sistema alcança ótimos resultados sob uma variedade de condições. Foi testado em filmagens de plantações de milho; o sistema detecta com sucesso uma média de 95% de ervas daninhas e 80% de plantas de cultivo sob diferentes condições de luminosidade, umidade do solo e estágio de desenvolvimento das plantas.

Um banco de dados para discriminação entre plantas de cultivo e ervas daninha é proposto em (HAUG; OSTERMANN, 2014). O banco de dados consiste de 60 imagens com etiquetas. As imagens foram adquiridas com o robô autônomo Bonirob numa plantação de cenouras em estágio de crescimento. A câmera utilizada captura luz visível e luz infravermelha. Em (HAUG et al., 2014) é proposto uma abordagem com visão computacional para classificação de plantas sem segmentação. Ao invés de segmentar a imagem em folhas individuais ou plantas, foi utilizado um classificador por *Random Forest* para estimar as posições de pixels de plantas e ervas daninhas baseado em características extraídas da vizinhança. Os resultados são suavizados no espaço utilizando campo aleatório de Markov e regiões contínuas de plantas e ervas daninhas são inferidas em imagens de resolução plena através de interpolação.

Em (KOUNALAKIS; TRIANTAFYLLIDIS; NALPANTIDIS, 2016), os autores introduzem um então novo método que aplica características conhecidas por imagem combinadas com representações lineares avançadas de imagens para reconhecimento de ervas daninhas. O método proposto é baseado no que havia de estado da arte então para categorização de objetos e imagens, explorando desempenho aprimorado utilizando algoritmos de aprendizado de máquina. O sistema resultante pode ser aplicado numa variedade de ambientes, plantações e tipos de ervas daninhas.

Em (TANG et al., 2017) as mudas de soja e suas ervas daninhas associadas são objeto de pesquisa. Neste trabalho citado, é construído um modelo para identificar ervas

daninhas baseado em aprendizado de características utilizando k-means com redes convolutivas. O método proposto k-means para realizar aprendizado de características como processo de pré-treinamento, substituindo o método de inicialização aleatória dos parâmetros das redes convolutivas. Com esse método é possível inicializar os parâmetros com valores mais razoáveis, fazendo com que a rede treinada tenha maior precisão para identificação de ervas daninhas.

Em (DI CICCO et al., 2017), os autores atacam o problema da falta de *datasets* de imagens de agricultura, devida à dificuldade de construir um *dataset*, propondo um sistema que minimiza a intervenção humana necessária para treinar algoritmos de classificação e detecção. A abordagem é gerar *datasets* sintéticos generalizando as principais características do ambiente alvo (espécies de plantas e ervas daninhas, tipos de solo, condições de luminosidade). Mais especificamente, através do ajuste de parâmetros de modelo e explorando algumas texturas reais, foi possível renderizar uma grande quantidade de vistas realistas de um cenário agrícola artificial com pouco esforço.

Em (LOTTES et al., 2018), é proposto um sistema de classificação entre planta e erva daninha através de *fully convolutional network* com uma estrutura codificadora-decodificadora que incorpora informação espacial considerando sequências de imagens. Os autores proveram uma avaliação experimental, mostrando que o sistema proposto é capaz de generalizar habilmente para campos previamente não vistos sob condições de ambiente variáveis, uma característica fundamental para que o sistema seja realmente utilizável em agricultura de precisão.

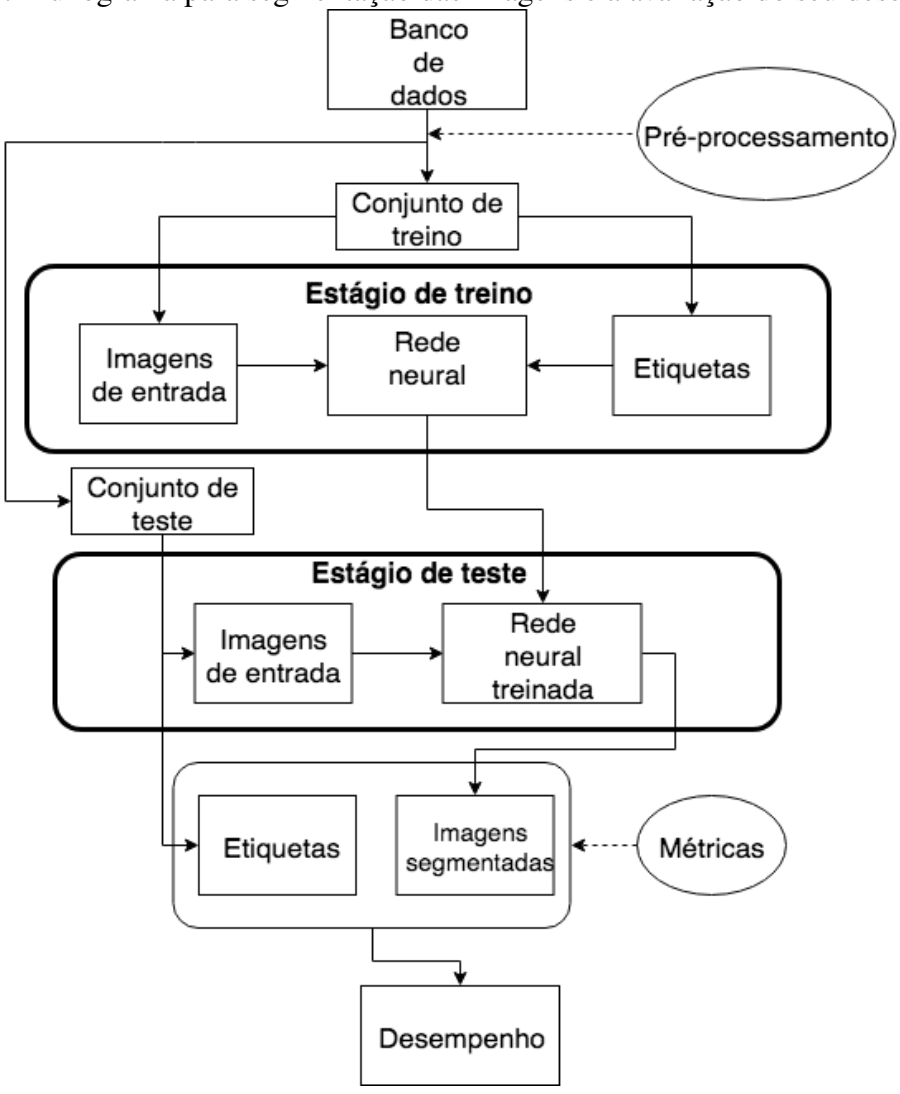
Em (MILIOTO; LOTTES; STACHNISS, 2018) os autores propõem uma solução baseada em redes convolutivas para o problema de segmentação semântica de campos de plantio entre plantas de beterraba, ervas daninha e solo baseada em imagens em RGB. A rede proposta explora índices de vegetação existentes e proveem classificação em tempo real.

Em (UTSTUMO et al., 2018) é proposto um robô capaz de detectar ervas daninhas e realizar tratamento em tempo real através de herbicidas localmente aplicados, sem trazer prejuízo às plantas de cultivo. O algoritmo do robô utiliza SVM (*Support Vector Machine*) para realizar a classificação e faz uso do sistema *Drop on Demand* (DoD) para a aplicação do herbicida.

4 METODOLOGIA

Para construir um detector automático de ervas daninhas utilizando redes neurais, o primeiro passo é escolher o banco de dados a ser utilizado para treinar a rede. Como se trata de um detector por segmentação de imagem, o banco de dados deve ser constituído de imagens e suas respectivas etiquetas. A etiqueta de cada imagem é uma matriz que relaciona cada pixel a uma classe.

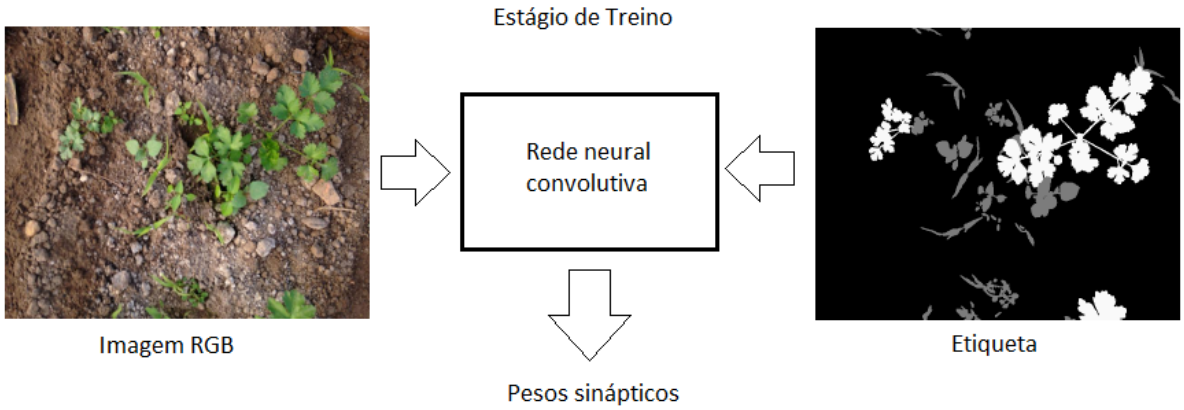
Figura 18: Fluxograma para segmentação das imagens e a avaliação do seu desempenho.



O banco de dados selecionado para essa tarefa é descrito na Seção 4.1. O treinamento das redes neurais para segmentação de imagem é por aprendizagem supervisionada, em que as etiquetas cumprem o papel do professor. A Figura 18 resume o processo de treino e teste da uma rede neural para essa tarefa.

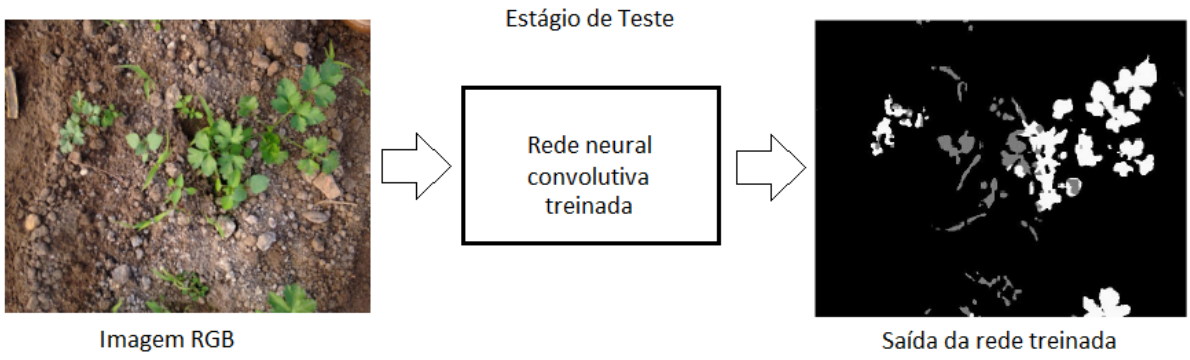
A partir de um banco de dados pré-processado, separamos os dados em dois conjuntos: o conjunto de treino e o conjunto de teste. O conjunto de treino é a parte do banco de dados que é utilizada para treinar realizar o aprendizado supervisionado da rede. No estágio de treino, a rede tem como entrada imagens RGB e suas respectivas etiquetas, e como saída os pesos sinápticos, conforme a Figura 19.

Figura 19: Treinamento de uma rede neural.



As imagens do conjunto de teste são processadas pela rede treinada para que ela realize a segmentação, conforme a Figura 20. A partir das imagens segmentadas pela rede e das etiquetas do conjunto de teste, se faz a comparação entre ambas utilizando métricas para mensurar o desempenho da rede neural treinada.

Figura 20: Estágio de teste de uma rede neural.



O uso de abordagens de aprendizagem profunda para segmentação de imagens aumentou nos últimos anos. O processo de segmentação de imagens consiste em uma classificação pixel a pixel de uma imagem, detectando e localizando objetos de uma ou mais categorias. Desde *AlexNet* (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), as redes neurais convolucionais (CNN) têm sido amplamente utilizadas para tarefas de classificação

e segmentação de imagens. Neste trabalho, serão apresentados resultados do uso da rede completamente convolucional, *fully convolutional networks* (FCN), (LONG; SHELHAMER; DARRELL, 2015) e de SegNet (BADRINARAYANAN; KENDALL; CIPOLLA, 2017) para executar a segmentação de planta/erva daninha/solo para o conjunto de dados criado por (LAMESKI et al., 2017).

4.1 Banco de Dados (*Dataset*)

O banco de dados utilizado nesse trabalho é de uma plantação de cenouras na região de Negotino, República da Macedônia. Há 39 imagens no banco de dados, todas adquiridas por uma câmera de celular de 10 megapixels (3264 $\times$ 2448 pixels), proposto em (LAMESKI et al., 2017). O banco de dados consiste de 311.620.608 pixels no total, dos quais 26.616.081 são respectivos a planta de cenoura, 18.503.308 são pixels de erva daninha e 266.501.219 são pixels de solo. As imagens foram geradas a uma distância de 1 metro do solo. As etiquetas produzidas pelos autores foram feitas a partir da segmentação das imagens, utilizando de base o índice ExGExR para separar os vegetais do solo. A separação entre planta e erva daninha foi realizada com inspeção manual. O número inteiro ”0” foi atribuído a pixels de solo, ”1” para pixels referentes a erva daninha e ”2” para pixels de plantas de cenoura. A Figura 21 mostra um par imagem/etiqueta. Como cada pixel é representado por um *byte*, cujo valor varia de 0 (preto mais escuro) a 255 (branco mais claro), os pixels relativos a cada classe são indistinguíveis entre si ao olhar. Para fins de visualização, a etiqueta foi multiplicada por um fator de 125, de modo que o solo apareça em preto (0), ervas daninha em cinza (125) e as plantas de cenoura em branco (250).

Figura 21: Um exemplo de imagem do banco de dados com sua respectiva etiqueta.

(a) Imagem (b) Etiqueta

Antes que seja feito o treinamento de fato da rede neural, é necessário realizar um pré-processamento das imagens do banco de dados. Na seção 4.2 é descrito o pré-processamento feito para cada ensaio realizado.

4.2 Ensaios

Para a validação da metodologia, dois ensaios foram realizados. O primeiro ensaio foi realizado para comparar o desempenho entre as arquiteturas FCN e SegNet para o

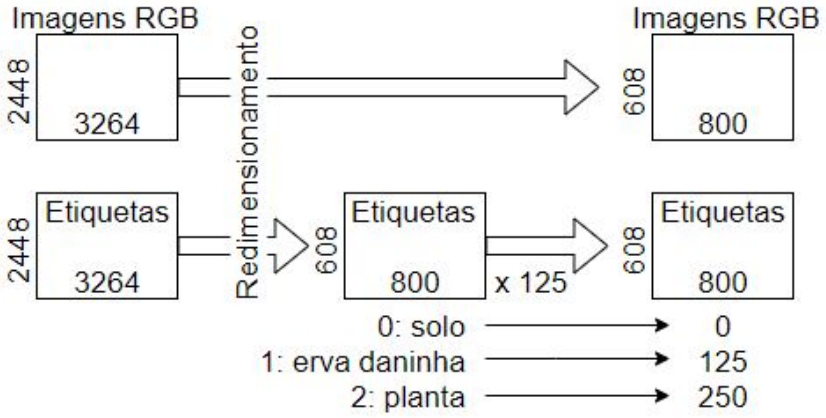
banco de dados explicado na seção 4.1. No segundo, algumas modificações são propostas e implementadas para a arquitetura FCN. Os resultados destas práticas são discutidos no Capítulo 5.

4.2.1 Ensaio para comparação das arquiteturas de rede propostas

Neste primeiro ensaio, é realizada a comparação do desempenho entre as arquiteturas SegNet e FCN para segmentação de imagens aplicadas ao bando de dados de (LAMESKI et al., 2017).

O banco de dados foi inicialmente dividido em dois conjuntos de treinamento e teste. A seleção foi feita aleatoriamente, onde 38,4% das imagens formaram o conjunto de treinamento, enquanto 61,5% formaram o conjunto de teste. As proporções geralmente usadas são 80/20 ou 70/30, no entanto proporções semelhantes não devem ter um grande impacto no modelo. Portanto, 15 imagens e suas respectivas etiquetas foram selecionadas para o estágio de treinamento, enquanto as 24 imagens restantes foram usadas para testar o desempenho de segmentação da rede treinada. A Figura 22 resume o pré processamento realizado para este ensaio. As imagens e suas etiquetas foram redimensionadas de 3264 $\times$ 2448 para 800 $\times$ 608. Com a diminuição das imagens, o treinamento das redes demanda muito menos tempo, devido à considerável diminuição do número de parâmetros a treinar. Houve uma pequena alteração na razão entre as dimensões (largura/altura) de 1,3333 para 1,3158. Isso foi feito porque ambas as dimensões precisam ser divisíveis por 32 em função das operações de sobreamostragem da arquitetura FCN (LONG; SHELHAMER; DARRELL, 2015).

Figura 22: Processamento realizado sobre o banco de dados para a realização do treino.



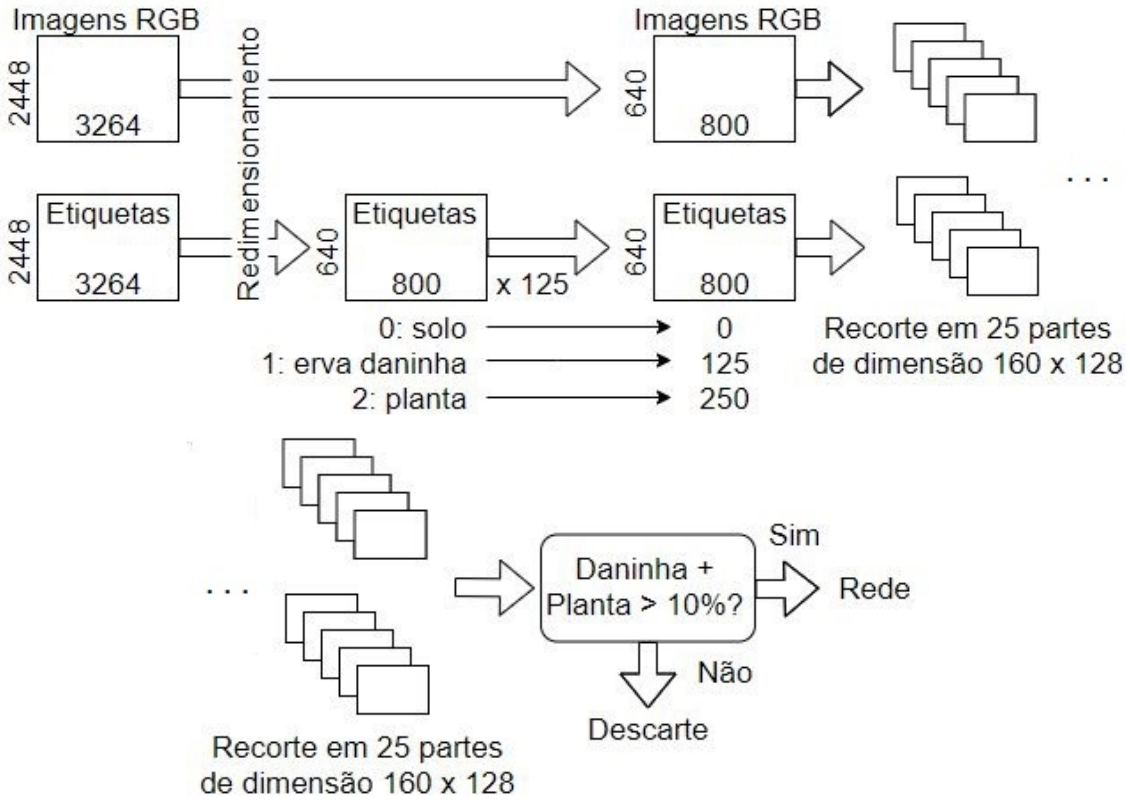
A rede FCN implementada é treinada em transferência com base no modelo VGG-16 usando a abordagem descrita em (LONG; SHELHAMER; DARRELL, 2015), ou seja, utilizando 16 camadas com convolução 3x3 e subamostragem de 2x2 em toda a rede, implementada utilizando a biblioteca TensorFlow, em linguagem Python. O modelo baseado em SegNet foi implementado em MATLAB. Ambas as redes foram treinadas em 600 épocas (*epochs*) usando tamanho de lote (*batch size*) de uma imagem. O treinamento foi realizado em um processador i7 557 com 24 gigabytes de RAM.

Após o treinamento, uma inferência sobre o conjunto de teste é realizado sobre o conjunto de teste de 24 imagens utilizando ambas as redes treinadas. As imagens segmentadas são comparadas entre si, tendo as etiquetas como referência, utilizando as métricas apropriadas, conforme a Figura 18. As métricas são explicadas no Capítulo 5.

4.2.2 Rede FCN com novo pré processamento

Neste segundo ensaio, dois treinamentos são realizados utilizando a mesma arquitetura de rede FCN, aplicada ao banco de dados de (LAMESKI et al., 2017). O pré processamento realizado nesse ensaio é semelhante ao do ensaio anterior, porém com alguns ajustes. A dimensão utilizada para as imagens e respectivas etiquetas é de 800 $\times$ 640. Isso porque elas serão recortadas em 25 partes iguais de dimensão 160 $\times$ 128 (valores divisíveis por 32). Posteriormente, os recortes de imagens passam por uma seleção: se o recorte possui mais de 10% de pixels de vegetação (planta ou erva daninha), ele é utilizado para o treinamento; caso contrário, o recorte é descartado. Essa etapa é descrita na Figura 23. Importante destacar que o recorte e seleção só foi feito para os conjuntos de treino, e não de teste.

Figura 23: Processamento realizado sobre o banco de dados para a realização do treino.



Isso foi realizado para analisar se há impacto positivo no desempenho de segmentação para os pixels de erva daninha e plantas, tendo em vista que o foco é detectar ervas daninhas. A Tabela 1 mostra os percentuais de pixels no conjunto de cada treinamento realizado. Para fins de simplificação, os ensaios das seções 4.2.1 e 4.2.2 são chamados de Ensaio 1 e Ensaio 2, respectivamente.

No primeiro treinamento deste ensaio, 15 imagens são utilizadas para treinamento e 24 para teste, utilizando os mesmos conjuntos de treino e teste do primeiro ensaio. A partir dos resultados desse treinamento, é possível analisar o impacto da seleção realizada, comparando com os resultados da FCN do primeiro ensaio. No segundo treinamento, 27 imagens são utilizadas para treino e 12 para teste. Importante destacar que as 12 imagens de teste deste segundo treinamento estão contidas no conjunto de teste de 24 imagens dos treinamentos anteriores. Com o segundo treinamento, é possível analisar se o desempenho

Tabela 1: % de pixels de cada classe para cada treinamento pós processamento das imagens

Ensaio	**Dimensões**	**Treino/Teste**	**Conjunto**	**% Solo**	**% Erva dan.**	**% Planta**
1	800×608	15/24	Treino	85,04	8,38	6,58
			Teste	85,39	5,25	9,36
2	800×640		Treino	74,90	13,00	12,10
			Teste	85,67	4,69	9,64
		27/12	Treino	74,67	11,37	13,96
			Teste	84,01	4,51	11,48

de segmentação melhora com o aumento do conjunto de treino. Assim como no primeiro ensaio, os treinamentos do segundo ensaio foram realizados em processador i7 557 com 24 gigabytes de RAM, sendo a rede FCN programada com a biblioteca TensorFlow, em linguagem Python.

5 RESULTADOS

5.1 Ensaio para comparação das arquiteturas propostas

A abordagem usada neste ensaio resultou em uma precisão global (soma de todas as predições corretas divididas por todas as previsões) de 94,1055% para a rede FCN treinada e 93,0964% para a rede SegNet treinada no conjunto de validação. A Figura 25(a) mostra uma das imagens do conjunto de testes, Figura 25(b) mostra sua respectiva etiqueta, figuras 25(c) e 25(d) mostram a segmentação feita pela FCN treinada e SegNet, respectivamente.

Tabela 2: Matriz de confusão (%). FCN à esquerda, SegNet à direita.

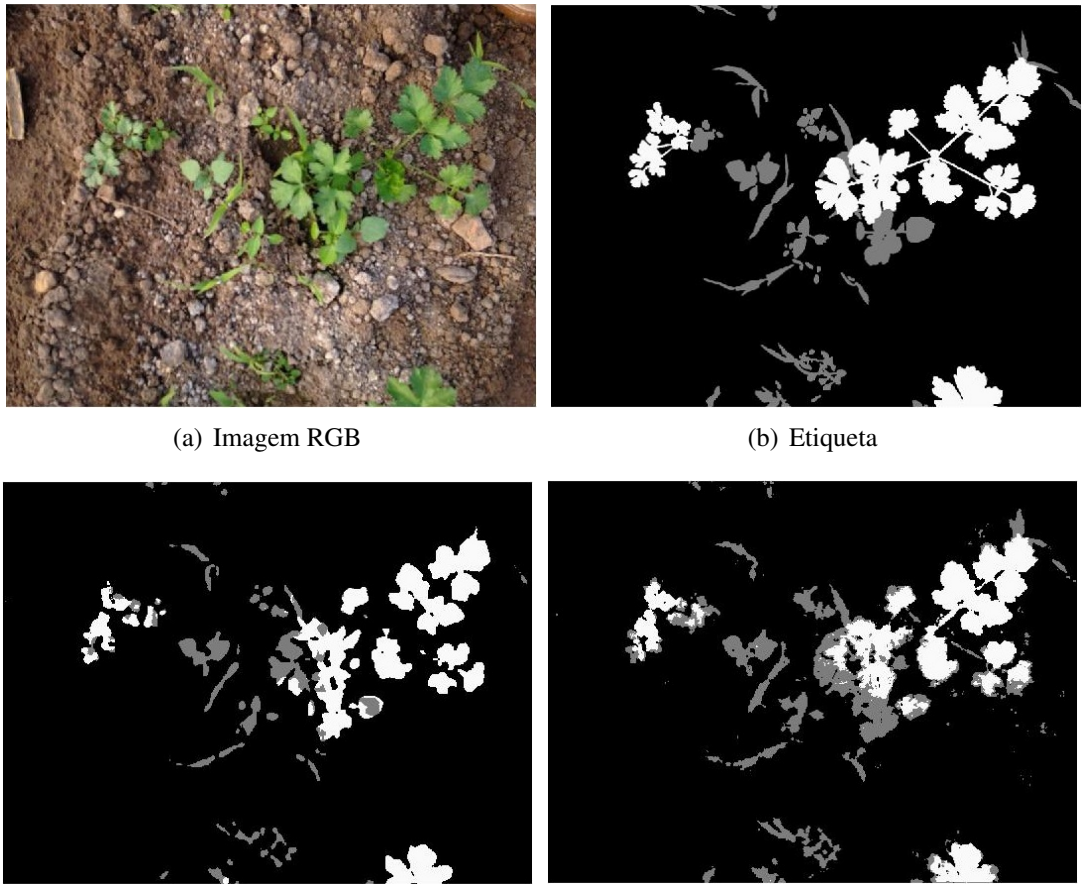
		Classe predita					
		Solo	**Erva**	**Planta**	**Solo**	**Erva**	**Planta**
Classe real	**Solo**	**85,2442**	0,0566	0,0866	**82,5711**	1,7239	1,0924
	Erva	2,6660	**1,8119**	0,7707	1,3126	**2,8765**	1,0595
	Planta	1,9386	0,3761	**7,0494**	0,5723	1,1430	**7,6488**

As matrizes de confusão são mostradas na Tabela 2. Estes resultados mostram a porcentagem de pixels inferida corretamente como uma classe *i* ou incorretamente inferida como outra classe *j*. Por exemplo: para a rede FCN: 85,2442 % dos pixels previstos foram corretamente classificados como pixels de solo, 0,0866 % preditos erroneamente como planta, etc. Os valores mostrados em negrito correspondem aos valores corretamente previstos.

Como se pode ver na Tabela 2, apesar do fato de que a rede FCN produziu uma precisão global um pouco maior, a SegNet inferiu corretamente uma porcentagem maior de pixels de ervas daninhas e plantas, enquanto o FCN inferiu corretamente mais pixels de solo. Como o conjunto de dados não é balanceado (os pixels do solo representam mais de 80 % do banco de dados), a precisão global por si só não fornece informações suficientes para fazer uma comparação entre os desempenhos das redes.

Assim, foram utilizadas as métricas de desempenho *Precision* (P), *Recall* (R) e *Intersection over Union* (IoU) para fazer uma comparação melhor. Essas métricas estão entre as mais usadas para medir o desempenho de segmentação de imagens (KAMILARIS; PRENAFETA-BOLDÚ, 2018).

Figura 24: Etiquetas geradas pelas redes e a etiqueta original para uma imagem do conjunto de teste.



(a) Imagem RGB

(b) Etiqueta

(c) Saída da FCN treinada

(d) Saída da SegNet treinada

$$P = \frac{tp}{tp + fp}, \tag{10}$$

$$R = \frac{tp}{tp + fn}, \tag{11}$$

$$IoU = \frac{tp}{tp + fn + fp}, \tag{12}$$

sendo tp, fp e fn o número de verdadeiros positivos, falsos positivos e falsos negativos, respectivamente (SOKOLOVA; LAPALME, 2009; KAMILARIS; PRENAFETA-BOLDÚ, 2018). Para um dado mais detalhado, esses parâmetros foram calculados por classe, porque seus valores médios seriam muito mais influenciados pela segmentação de pixels de solo, em função de estar em muito maior número no banco de dados, como dito anteriormente. Na Figura 25, os desempenhos das arquiteturas FCN e SegNet para segmentação de solo, erva daninha, planta são comparados.

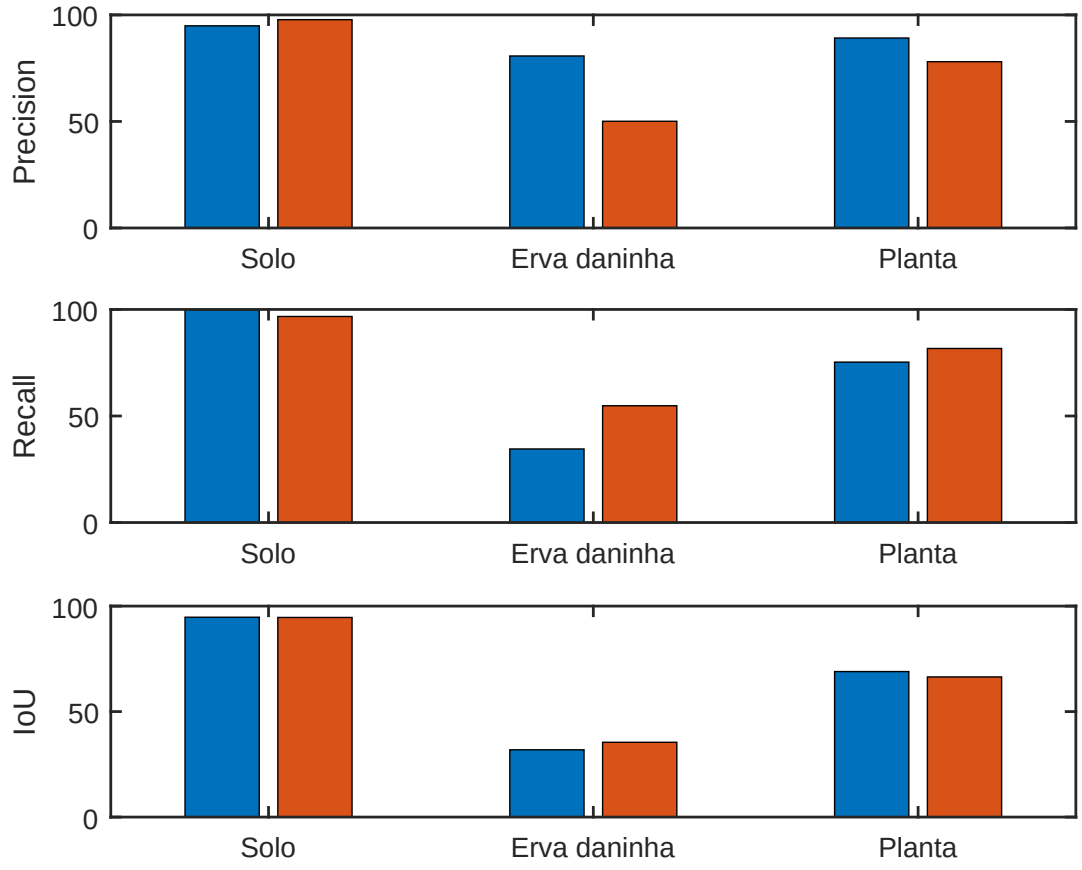


Figura 25: FCN em azul, SegNet em vermelho.

Conforme mostrado na Figura 25, P da rede FCN da segmentação de ervas daninhas e plantas é maior que P da rede SegNet (89,2 % vs. 78,0 % para pixels de plantas, 80,7 % vs. 50,1 % para erva daninha). Embora a rede SegNet tenha corretamente inferido

números maiores de pixels de ervas daninhas e plantas de cultivo (como mostrado na Tabelas 2), a rede FCN inferiu incorretamente menores números de pixels de ervas daninhas e de plantas, o que significa que a rede FCN produziu um número menor de falsos positivos para estas duas classes, o que explica P maior da rede FCN para a plantas e ervas daninhas.

Por outro lado, a segmentação realizada pela rede SegNet apresenta maior *Recall* para ervas daninhas e plantas (81,7 % vs. 75,3 % para a plantas, 54,8 % 34,5 % para ervas daninhas). *Recall* aumenta com o número de positivos verdadeiros, mas diminui com o número de falsos negativos. A rede FCN inferiu incorretamente um maior número de pixels de ervas daninhas e plantas como pixels de solo do que a rede SegNet, o que afetou *Recall* para ambas as classes.

Precision, *Recall* e *IoU* foram semelhantes para pixels de solo, para ambas as arquiteturas e acima de 94 %. Isso ocorre principalmente porque, conforme destacado anteriormente, mais de 80 % do conjunto de dados é composto por pixels no solo. A quantidade de falsos negativos e falsos positivos não é significativa quando comparada com a quantidade de verdadeiros positivos para ambos os casos.

Ambas as arquiteturas mostraram resultados semelhantes em termos de precisão global. No entanto, a principal diferença entre os resultados das duas abordagens é que o modelo baseado em FCN apresentou maior percentual de falsos negativos para as classes de culturas e plantas daninhas, e o modelo baseado em SegNet apresentou maior percentual de falsos positivos para essas duas classes.

5.2 Rede FCN com banco de dados (LAMESKI et al., 2017)

No ensaio 2, são comparados os desempenhos de segmentação das três redes treinadas de arquitetura FCN: a rede FCN treinada no ensaio 1; e as redes FCN do ensaio 2, uma treinada com 15 imagens e testada em 24, outra treinada em 27 imagens e testada em 12. A Tabela 3 mostra o percentual de segmentação referente a cada uma das três redes, de maneira semelhante à Tabela 2 na seção anterior.

Tabela 3: Matriz de confusão de FCN(15/24) do ensaio 1 no topo, FCN(15/24) do ensaio 2 no centro, FCN(27/12) do ensaio 2 mais abaixo.

		Classe predita		
		Solo	**Erva**	**Planta**
Classe real	**Solo**	**85,2442**	0,0566	0,0866
	Erva	2,6660	**1,8119**	0,7707
	Planta	1,9386	0,3761	**7,0494**
	Solo	**85,1127**	0,2344	0,3218
	Erva	1,3147	**2,7411**	0,6360
	Planta	1,1256	0,4840	**8,0297**
	Solo	**83,2772**	0,3134	0,4167
	Erva	1,2685	**2,2267**	1,0140
	Planta	1,0655	0,4939	**9,9241**

Comparando os números entre FCN(15/24) do ensaio 1 e FCN(15/24) do ensaio 2, podemos notar que a segunda produziu maiores números de predições corretas de pixels de ervas daninhas e de plantas: um salto de 1,8119% para 2,7411% relativo à classe de

erva daninha, e de 7,0494% para 8,0297% relativo à classe de plantas. A precisão global também aumentou: de 94,1055% para 95,8835%. Analisando FCN(27/12) do ensaio 2 tem-se um aumento do número de pixels corretamente preditos de plantas, de 8,0297% para 9,9241%. Porém, uma diminuição de 2,7411% para 2,2267% de pixels de ervas daninha, e de 95,8835% para 95,4280% da precisão global. A Figura 26 compara os desempenhos das três redes segundo as métricas *Precision*, *Recall* e *Intersection over Union*, semelhante à Figura 25.

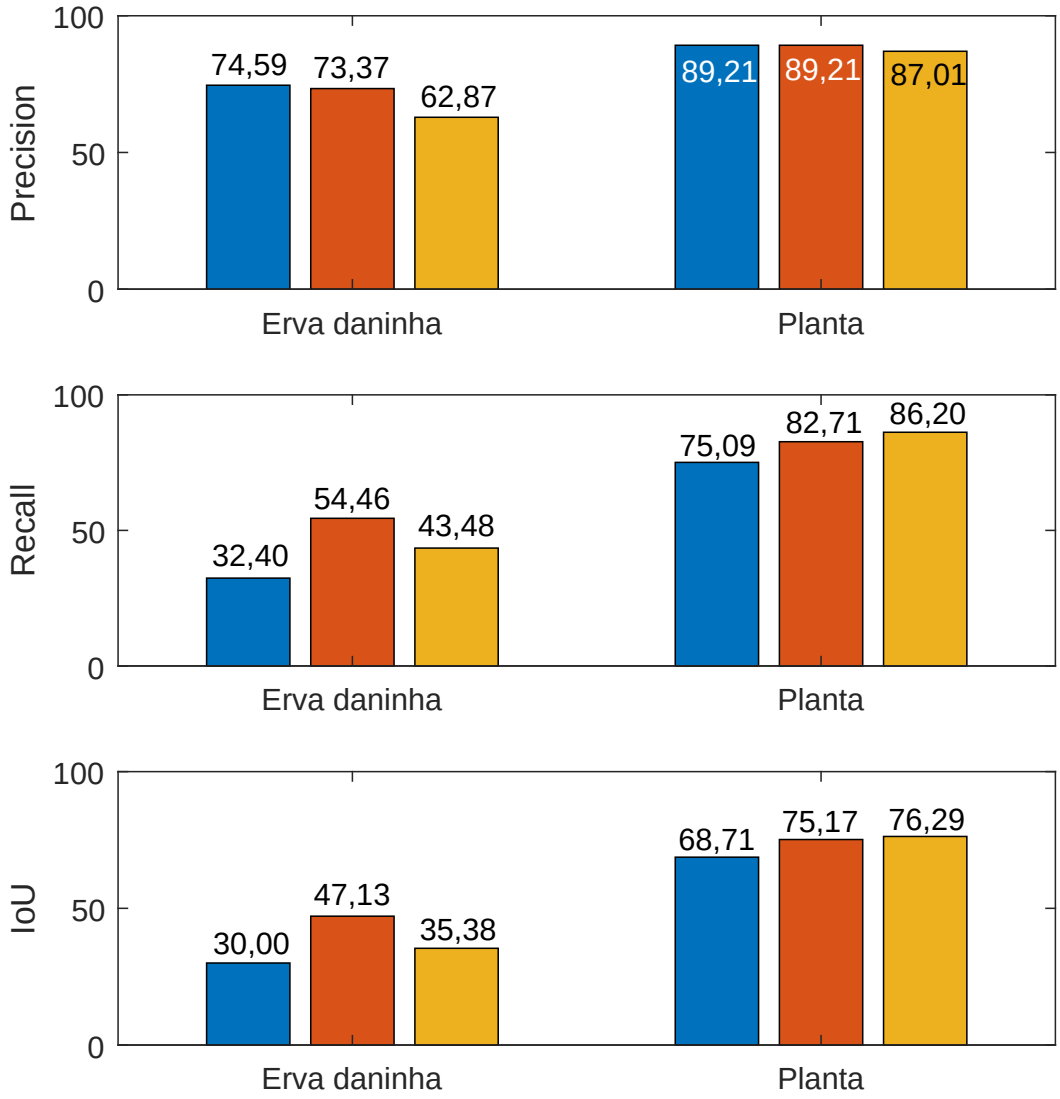


Figura 26: Em azul, FCN(15/24) do ensaio 1; em vermelho, FCN(15/24) do ensaio 2; em amarelo, FCN(27/12) do ensaio 2.

Nota-se a partir da Figura 26 uma melhora significativa das métricas *Recall* e *IoU* para os pixels de plantas e de ervas daninhas da rede FCN(15/24) do ensaio 2 em relação à rede FCN(15/24) do ensaio 1 (*Recall* aumentou de 32,40% para 54,46% para erva daninha, de

75,09% para 82,71% para a classe planta; *IoU* aumentou de 30,00% para 47,13% para a classe erva daninha, de 68,71% para 75,17% para a classe planta). Quanto à métrica *Precision*, houve uma pequena piora para a classe de erva daninha (de 74,59% para 73,37%) e a manutenção em 89,21% para a classe planta. As mudanças desses valores devem-se unicamente à diferença entre os pré-processamentos das duas redes. A rede FCN(27/12) do ensaio 2 demonstra piora em todas as métricas relativas aos pixels de erva daninha em relação à rede FCN(15/24) do mesmo ensaio (de 73,37% para 62,87% para a métrica *Precision*, de 54,46% para 43,48% para *Recall*, de 47,13% para 35,38% para *IoU*) e uma piora pequena de 89,21% para 87,01% de *Precision*, melhora de 82,71% para 86,20% de *Recall* e de 75,17% para 76,29% de *IoU* para a classe planta. Embora nota-se a piora de diversas métricas da rede FCN(27/12) em relação a FCN(15/24) do ensaio 2, é necessário destacar que o conjunto de teste de FCN(27/12) é diferente do conjunto de teste das outras duas redes (vide Seção 4.2.2).

As tabelas 4, 5, 6 e 7 mostram o desempenho de segmentação segundo às três métricas (*Precision*, *Recall* e *Intersection over Union*) relativo a cada imagem do conjunto de teste para as classes erva daninha e planta das três redes FCN treinadas (FCN(15/24) do ensaio 1, FCN(15/24) e FCN(27/12) do ensaio 2).

Tabela 4: Comparação entre FCN(15/24) do ensaio 1 e FCN(15/24) do ensaio 2 para pixels de plantas.

#	Planta (%)	P_{1PLA}	P_{2PLA}	R_{1PLA}	R_{2PLA}	IoU_{1PLA}	IoU_{2PLA}
1	8,17	87,33	95,22	72,33	62,91	65,46	60,98
2	10,07	90,00	92,78	80,22	84,91	73,65	79,64
3	3,83	76,48	79,23	73,54	66,29	59,97	56,48
4	8,69	94,69	93,07	68,14	84,32	65,63	79,34
5	8,17	86,67	96,24	78,17	84,63	69,78	81,93
6	10,31	96,68	95,40	70,41	79,22	68,75	76,31
7	3,90	91,42	88,86	76,43	88,41	71,31	79,59
8	7,36	93,21	94,55	68,18	83,09	64,96	79,29
9	4,24	81,28	88,57	78,76	84,31	66,66	76,04
10	8,76	95,48	92,19	74,10	85,11	71,59	79,39
11	9,59	92,10	86,28	79,51	89,31	74,43	78,21
12	7,94	88,54	92,35	72,52	84,81	66,30	79,24
13	5,14	83,03	65,58	71,20	76,20	62,16	54,43
14	11,98	92,14	92,82	78,28	86,97	73,38	81,49
15	7,06	96,76	95,37	68,71	80,48	67,17	77,46
16	14,12	89,48	91,40	71,35	77,77	65,83	72,47
17	15,13	94,22	91,13	70,99	78,82	68,03	73,21
18	12,03	94,18	87,86	80,99	88,11	77,13	78,54
19	10,55	90,49	89,61	78,30	86,94	72,35	78,98
20	11,73	95,40	91,65	77,49	86,26	74,70	79,98
21	12,48	96,52	93,67	81,15	90,71	78,84	85,47
22	10,24	87,20	86,29	79,53	81,50	71,22	72,16
23	10,46	75,33	78,02	80,55	88,20	63,74	70,64
24	12,79	72,30	82,81	71,20	85,82	55,94	72,84
$\bar{x}_{1-24}$	9,36	89,21	89,21	75,09	82,71	68,71	75,17

Os valores de índice 1, 2 e 3 são as métricas relativas às redes FCN(15/24) do ensaio 1, FCN(15/24) e FCN(27/12) do ensaio 2, respectivamente. $\bar{x}_{1-24}$ são as médias destes valores para as 24 imagens do conjunto de testes. $\bar{x}_{13-24}$ são as médias das últimas 12 imagens que são comuns aos dois conjuntos de testes das três redes FCN treinadas. As células em amarelo destacam o maior desempenho entre cada rede para cada uma das imagens e para cada uma das três métricas. As células em azul destacam o maior valor médio para cada métrica. A Tabela 5 mostra que, na média das 12 últimas imagens ($\bar{x}_{13-24}$), as métricas *Recall* e *Intersection over Union* apresentam melhora (de 83,98% para 86,20% para *R*, de 74,81% para 76,29% para *IoU*) e apenas uma pequena diminuição de *P* (de 87,18% para 87,01%) para a classe planta.

Tabela 5: Comparação entre FCN(15/24) e FCN(27/12) do ensaio 2 para pixels de plantas.

#	Planta (%)	P_{2PLA}	P_{3PLA}	R_{2PLA}	R_{3PLA}	IoU_{2PLA}	IoU_{3PLA}
1	8,17	95,22	-	62,91	-	60,98	-
2	10,07	92,78	-	84,91	-	79,64	-
3	3,83	79,23	-	66,29	-	56,48	-
4	8,69	93,07	-	84,32	-	79,34	-
5	8,17	96,24	-	84,63	-	81,93	-
6	10,31	95,40	-	79,22	-	76,31	-
7	3,90	88,86	-	88,41	-	79,59	-
8	7,36	94,55	-	83,09	-	79,29	-
9	4,24	88,57	-	84,31	-	76,04	-
10	8,76	92,19	-	85,11	-	79,39	-
11	9,59	86,28	-	89,31	-	78,21	-
12	7,94	92,35	-	84,81	-	79,24	-
13	5,14	65,58	68,69	76,20	82,93	54,43	60,18
14	11,98	92,82	91,49	86,97	89,86	81,49	82,93
15	7,06	95,37	93,49	80,48	82,53	77,46	78,04
16	14,12	91,40	92,42	77,77	84,21	72,47	78,77
17	15,13	91,13	92,56	78,82	84,07	73,21	78,75
18	12,03	87,86	87,82	88,11	90,17	78,54	80,15
19	10,55	89,61	89,72	86,94	87,75	78,98	79,73
20	11,73	91,65	92,07	86,26	86,94	79,98	80,89
21	12,48	93,67	94,74	90,71	86,65	85,47	82,67
22	10,24	86,29	85,80	81,50	81,86	72,16	72,10
23	10,46	78,02	77,02	88,20	89,81	70,64	70,83
24	12,79	82,81	78,31	85,82	87,60	72,84	70,50
$\bar{x}_{1-24}$	9,36	89,21	87,01	82,71	86,20	75,17	76,29
$\bar{x}_{13-24}$	11,14	87,18	87,01	83,98	86,20	74,81	76,29

As tabelas 6 e 7 comparam os desempenhos de segmentação das três redes para pixels de erva daninha. Observa-se nessas duas tabelas que as métricas para as imagens com menos de 3% de pixels de erva daninha (imagens 19, 20 e 21) apresentam os mais baixos valores de desempenho de segmentação para as três métricas, devido justamente ao baixo número de pixels de erva daninhas a serem segmentados pelas redes, fato que por si só aumenta a probabilidade de erro. A Figura 27 mostra as imagens geradas pelas três redes a etiqueta original da imagem 20 do conjunto de teste.

Nota-se também a partir das tabelas 4 e 6 que, apesar de FCN(15/24) do ensaio 2 apresentar em média melhor desempenho em relação a FCN(15/24) do ensaio 1, FCN(15/24) do ensaio 1 apresentou melhor desempenho segundo uma ou mais métricas para algumas das imagens do conjunto de teste. Isso denota a necessidade de se ter um conjunto de dados com alta variabilidade para uma avaliação fidedigna do desempenho de segmentação.

Tabela 6: Comparação entre FCN(15/24) do ensaio 1 e FCN(15/24) do ensaio 2 para pixels de erva daninha.

#	Erva (%)	P_{1ERV}	P_{2ERV}	R_{1ERV}	R_{2ERV}	IoU_{1ERV}	IoU_{2ERV}
1	9,79	69,96	57,51	38,47	76,60	33,01	48,91
2	8,04	81,69	76,84	32,80	70,61	30,55	58,22
3	6,61	91,64	72,73	31,80	60,67	30,90	49,43
4	6,62	74,81	86,58	48,18	71,02	41,46	63,98
5	7,95	94,04	89,30	41,30	81,98	40,25	74,64
6	8,62	88,20	85,45	50,56	74,96	47,35	66,48
7	4,82	87,34	90,51	38,89	63,09	36,82	59,17
8	4,79	73,42	83,46	45,93	73,29	39,38	64
9	5,51	98,25	92,93	39,30	69,42	39,02	65,94
10	5	88,54	88,04	43,55	66,92	41,23	61,34
11	6,63	95,86	93,08	44,68	57,91	43,84	55,52
12	4,58	84,78	92,78	39,53	63,87	36,91	60,85
13	5,70	95,40	81,50	38,40	35,29	37,70	32,67
14	3,95	87,06	86,42	30,86	58,37	29,51	53,47
15	3,17	80,50	81,65	42,96	70,50	38,91	60,86
16	3,71	51,32	51,50	18,44	42,39	15,70	30,30
17	3,60	46,64	38,23	14,11	22,98	12,15	16,76
18	3,10	85,30	59,94	25,03	20,95	23,99	18,37
19	1,63	25,80	57,36	3,54	20,81	3,21	18,02
20	0,68	10,99	4,22	4,94	4,95	3,53	2,33
21	0,95	34,99	48,67	17,40	40,45	13,15	28,35
22	4,10	75,13	63,80	26,89	51,44	24,69	39,82
23	10,75	92,38	89,87	30,14	50,16	29,41	47,47
24	12,77	76,13	88,42	30,02	58,39	27,43	54,25
$\bar{x}_{1-24}$	4,69	74,59	73,37	32,40	54,46	30,00	47,13

Como frisado anteriormente, a rede FCN(27/12) do ensaio 2 é inferior segundo às três métricas em relação a rede FCN(27/12) do mesmo ensaio quando analisa-se as redes na totalidade de seus conjuntos de teste. Em contrapartida, analisando a Tabela 5, observa-se que na média das últimas 12 imagens ($\bar{x}_{13-24}$) o desempenho de FCN(27/12) do ensaio 2 é maior segundo as três métricas em relação a FCN(15/24) do mesmo ensaio (*Precision*: de 62,63% para 62,87%, *Recall*: de 39,72% para 43,48%, *IoU*: de 33,56% para 35,38%).

Em (LAMESKI et al., 2017), o trabalho em que é apresentado o banco de dados abordado neste texto, os autores apresentam resultados preliminares de segmentação. Utilizam *Class Accuracy* (CA) – média entre todas as classes do número de predições corretas feitas para uma classe específica dividido pelo número real de amostras dessa classe – para mensurar os seus resultados. Foram treinadas 6 redes SegNet, com razão treino/teste 1/38 e 38/1; utilizando 3 sistemas de cores: RGB, HSV e Lab. A Tabela 8 traz uma comparação

entre resultados.

Tabela 7: Comparação entre FCN(15/24) e FCN(27/12) do ensaio 2 para pixels de erva daninha.

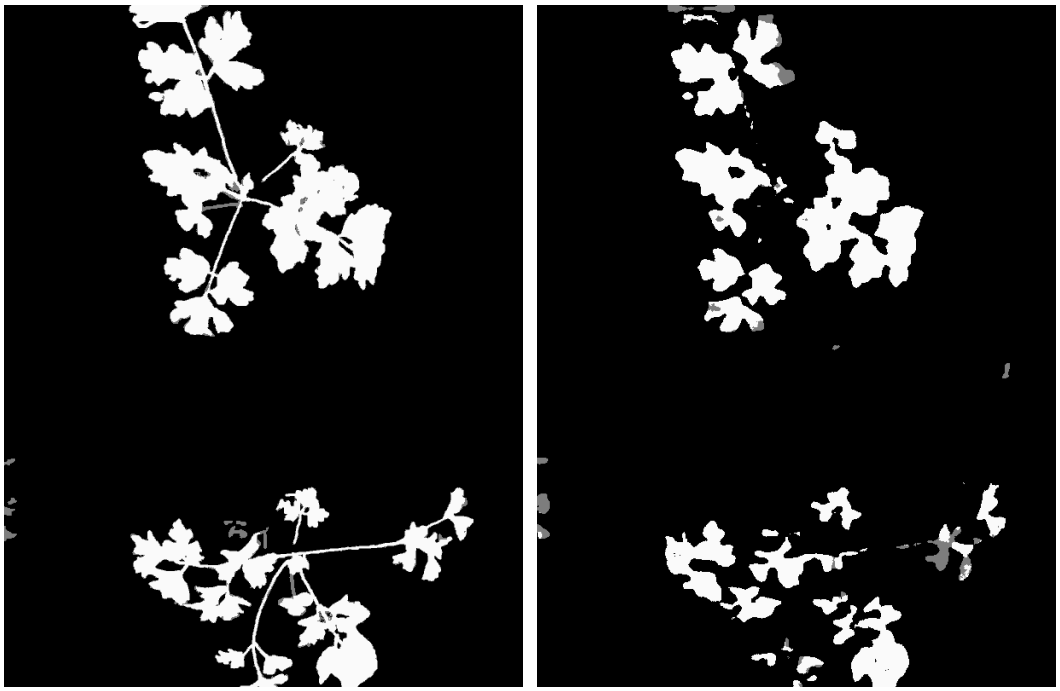
#	Erva (%)	P_{2ERV}	P_{3ERV}	R_{2ERV}	R_{3ERV}	IoU_{2ERV}	IoU_{3ERV}
1	9,79	57,51	-	76,60	-	48,91	-
2	8,04	76,84	-	70,61	-	58,22	-
3	6,61	72,73	-	60,67	-	49,43	-
4	6,62	86,58	-	71,02	-	63,98	-
5	7,95	89,30	-	81,98	-	74,64	-
6	8,62	85,45	-	74,96	-	66,48	-
7	4,82	90,51	-	63,09	-	59,17	-
8	4,79	83,46	-	73,29	-	64	-
9	5,51	92,93	-	69,42	-	65,94	-
10	5	88,04	-	66,92	-	61,34	-
11	6,63	93,08	-	57,91	-	55,52	-
12	4,58	92,78	-	63,87	-	60,85	-
13	5,70	81,50	87,74	35,29	43,14	32,67	40,69
14	3,95	86,42	86,59	58,37	58,27	53,47	53,45
15	3,17	81,65	80,56	70,50	68,14	60,86	58,52
16	3,71	51,50	64,28	42,39	53,06	30,30	40,98
17	3,60	38,23	50,33	22,98	31,75	16,76	24,18
18	3,10	59,94	67,18	20,95	22,95	18,37	20,64
19	1,63	57,36	48,66	20,81	26,70	18,02	20,83
20	0,68	4,22	8,30	4,95	12,33	2,33	5,22
21	0,95	48,67	27,81	40,45	41,89	28,35	20,07
22	4,10	63,80	57,54	51,44	54,81	39,82	39,03
23	10,75	89,87	89,09	50,16	52,70	47,47	49,50
24	12,77	88,42	86,39	58,39	55,95	54,25	51,42
$\bar{x}_{1-24}$	4,69	73,37	62,87	54,46	43,48	47,13	35,38
$\bar{x}_{13-24}$	4,51	62,63	62,87	39,72	43,48	33,56	35,38

Tabela 8: Comparação com (LAMESKI et al., 2017)

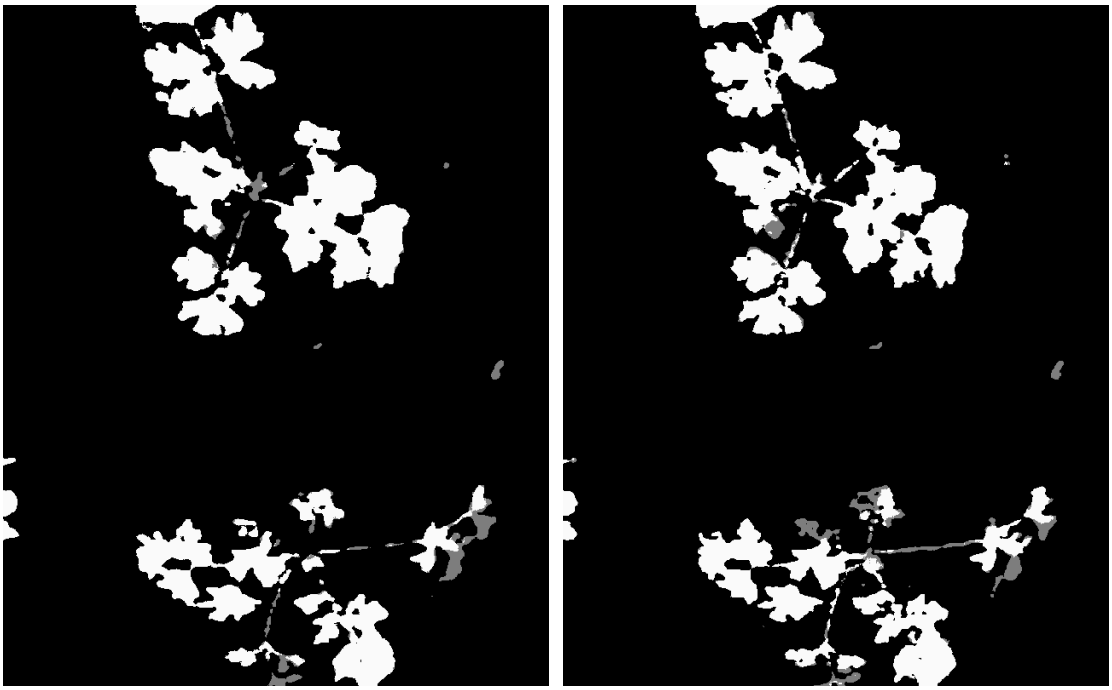
	Rede (treino/teste)	**CA (%)**
(LAMESKI et al., 2017)	$SegNet$ (1/38) RGB	59
	$SegNet$ (1/38) HSV	48,1
	$SegNet$ (1/38) Lab	52,1
	$SegNet$ (38/1) RGB	64,1
	$SegNet$ (38/1) HSV	60,9
	$SegNet$ (38/1) Lab	63
Nosso trabalho	FCN_2 (15/24)	80,5
	FCN_2 (27/12)	78,3

As redes FCN_2 são referentes às duas redes treinadas no segundo ensaio. Ambas as redes FCN_2 apresentam CA média superiores em relação às redes treinadas segundo a abordagem de (LAMESKI et al., 2017).

Figura 27: Etiquetas geradas pelas redes e a etiqueta original para a imagem 20 do conjunto de teste. As imagens foram rotacionadas em 90° para facilitar a visualização.



(a) Etiqueta (b) FCN ensaio 1 (15/24)



(c) FCN ensaio 2 (15/24) (d) FCN ensaio 2 (27/12)

6 CONCLUSÃO

Sistemas de agricultura de precisão estão cada vez mais presentes no ambiente agrícola e algoritmos inteligentes podem ser incluídos em drones e veículos terrestres para ação em tempo real. Neste trabalho, duas arquiteturas de aprendizagem profunda, FCN e SegNet, foram utilizadas e avaliadas para a segmentação de plantas daninhas e culturas. Um banco de dados aberto foi utilizado com 39 imagens e seus respectivos rótulos de uma plantação de cenoura. Todas as imagens foram adquiridas no espectro visível, tornando os métodos descritos neste trabalho úteis para sistemas que usam câmeras RGB tradicionais.

Em um primeiro ensaio, uma rede FCN e uma SegNet foram treinadas com o mesmo conjunto de treino e testadas sobre o mesmo conjunto de teste. Foram utilizadas 15 imagens para treino e 24 para teste. O pré-processamento realizado nas imagens foi no sentido de diminuir suas dimensões para exigir menor poder computacional. A quantidade de parâmetros das CNN, geralmente, é proporcional às dimensões das imagens de entrada, que constituem a primeira camada de uma rede neural artificial. Uma rede desnecessariamente grande não traz ganho em desempenho de segmentação, além de tomar maior tempo para realizar a inferência.

Os resultados de ambas as arquiteturas no primeiro ensaio mostraram precisão satisfatória para esta atividade e, portanto, podem ser úteis no desenvolvimento de sistemas de visão computacional para pulverização seletiva. O desempenho de ambos os algoritmos em relação a falsos negativos e falsos positivos deve ser considerado quando implementado em sistemas de pulverização seletiva. Falsos positivos indicam aplicações em locais não desejados, enquanto falsos negativos indicam falta de aplicação em locais onde deveria ser aplicado. A aplicação de herbicida na cultura pode ser indesejável, a menos que a cultura seja tolerante ao tipo de herbicida usado.

A arquitetura FCN foi novamente abordada noutro experimento com o objetivo de melhorar o desempenho da rede para detecção de ervas daninhas. Para isto, um novo pré-processamento foi realizado. Esse pré-processamento faz com que a quantidade de pixels de vegetação no conjunto de treinamento componha no mínimo 10% do total. Duas redes FCN foram treinadas desse modo: uma com conjunto de treino/teste igual ao do primeiro ensaio e outra com conjunto treino/teste de 27/12, com o intuito de observar se há algum ganho de desempenho com um maior conjunto de treino. O desempenho da FCN deste ensaio foi comparado com as redes treinadas pelos autores do banco de dados utilizado neste trabalho (LAMESKI et al., 2017).

A rede FCN(15/24) do segundo ensaio apresentou melhor desempenho nas métricas *Recall* e *Intersection over Union* e o mesmo valor de *Precision* para pixels de planta. Para a classe de pixels de erva daninha, houve crescimento significativo de *Recall* e *IoU* e uma leve diminuição de *Precision*. É possível concluir, nesse caso, que houve diminuição dos casos de falsos negativos para ambas as classes de pixels. Comparando FCN(15/24) com

FCN(27/12) do segundo ensaio, não há mudança a se destacar nas métricas para pixels de planta, mas há grande diminuição das três métricas para pixels de ervas daninhas. No entanto, quando se compara o desempenho de segmentação das duas redes nas 12 imagens em comum dos seus conjuntos de teste, nota-se que a rede FCN(27/12) apresenta melhor desempenho segundo às três métricas para a classe de pixels de ervas daninhas, e melhor *Recall* e *IoU* para pixels de plantas (*Precision* tem diferença desprezível de pouco mais de um décimo de %). Isso se deve à variabilidade das quantidades de pixels de cada classe de imagem a imagem dentro do conjunto de teste. As redes apresentaram baixo desempenho de segmentação quando o percentual de pixels a serem detectados era entre 3 e 0,5%, como mostram as tabelas 6 e 7. Finalizando, as redes do ensaio 2 demonstraram maior CA médio do que as redes treinadas em (LAMESKI et al., 2017).

Para futuros trabalhos nessa linha, seria interessante trabalhar em direção da viabilização do algoritmo para aplicação em campo. Além disso, pesquisar novas formas de melhorar ainda mais o desempenho de segmentação com técnicas de aprendizado de máquina, e trabalhar com novos bancos de dados.

REFERÊNCIAS

AHMAD, J. *et al* . Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems. **Computers in Industry**, Amsterdam, v.98, p.23–33, 2018.

ALCHANATIS, V. *et al* . Weed detection in multi-spectral images of cotton fields. **Computers and Electronics in Agriculture**, Amsterdam, v.47, n.3, p.243–260, 2005.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, New York, v.39, n.12, p.2481–2495, Dec 2017.

BOSILJ, P.; DUCKETT, T.; CIELNIAK, G. Analysis of morphology-based features for classification of crop and weeds in precision agriculture. **IEEE Robotics and Automation Letters**, Piscataway, v.3, n.4, p.2950–2956, 2018.

BURGOS-ARTIZZU, X. P. *et al* . Real-time image processing for crop/weed discrimination in maize fields. **Computers and Electronics in Agriculture**, Amsterdam, v.75, n.2, p.337–346, 2011.

CARLSON, T. N.; RIPLEY, D. A. On the relation between NDVI, fractional vegetation cover, and leaf area index. **Remote Sensing of Environment**, Amsterdam, v.62, n.3, p.241–252, 1997.

CHEN, L.-C. *et al* . Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Los Alamitos, v.40, n.4, p.834–848, 2018.

CIREŞAN, D. C. *et al* . Deep, big, simple neural nets for handwritten digit recognition. **Neural Computation**, Cambridge, v.22, n.12, p.3207–3220, 2010.

CORDEAU, S. *et al* . Bioherbicides: dead in the water? a review of the existing products for integrated weed management. **Crop Protection**, Amsterdam, v.87, p.44–49, 2016.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v.20, n.3, p.273–297, 1995.

DI CICCO, M. *et al*. Automatic model based dataset generation for fast and accurate crop and weeds detection. *In*: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), Vancouver. **Proceedings[. . .]**. IEEE. Piscataway, 2017. p.5188–5195. Disponível em: https://ieeexplore.ieee.org/document/8206408. Acesso em 10 out. 2017.

DYRMANN, M.; KARSTOFT, H.; MIDTIBY, H. S. Plant species classification using deep convolutional neural network. **Biosystems Engineering**, Amsterdam, v.151, p.72–80, 2016.

FUKUSHIMA, K. Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron. **IEICE Technical Report, A**, Tokyo, v.62, n.10, p.658–665, 1979.

GIRSHICK, R. Fast r-cnn. *In*: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION, 2015, Santiago. **Proceedings[...]** Piscataway: 2015. p.1440–1448. Disponível em: https://ieeexplore.ieee.org/document/7410526. Acesso em: 15 nov. 2017.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. São Paulo: Edgard Blucher, 2000.

HAMUDA, E.; GLAVIN, M.; JONES, E. A survey of image processing techniques for plant extraction and segmentation in the field. **Computers and Electronics in Agriculture**, Amsterdam, v.125, p.184–199, 2016.

HASSAN, M. **VGG16 - Convolutional Network for Classification and Detection**. 2018. Disponível em: https://neurohive.io/en/popular-networks/vgg16/. Acesso em: 09 jul. 2019.

HAUG, S.; OSTERMANN, J. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. *In*: EUROPEAN CONFERENCE ON COMPUTER VISION, **Proceedings[...]**. Springer, Cham, 2014. p.105–116.

HAUG, S. *et al*. Plant classification system for crop/weed discrimination without segmentation. *In*: APPLICATIONS OF COMPUTER VISION (WACV), IEEE WINTER CONFERENCE ON, Steamboat Springs, **Proceedings[...]**. IEEE, Piscataway. 2014. p.1142–1149. Disponível https://ieeexplore.ieee.org/document/6835733. Acesso em: 12 dez. 2017.

HAYKIN, S. **Redes neurais**: princípios e prática. Porto Alegre: Bookman, 2001.

HE, K. *et al* . Deep residual learning for image recognition. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, Las Vegas. **Proceedings[...]** Piscataway: IEEE, 2016. p.770–778.

HE, K. *et al* . Mask r-cnn. *In*: ICCV), IEEE INTERNATIONAL CONFERENCE ON, 2017, Veneza. **Proceedings[...]** Piscataway: IEEE , 2017. p.2980–2988. Disponível em: https://ieeexplore.ieee.org/document/8237584. Acesso em: 15 jan. 2018.

HINTON, G. E. *et al* . Improving neural networks by preventing co-adaptation of feature detectors. **Neural and Evolutionary Computing**, Toronto, 2012.

HUANG, G. *et al*. Densely connected convolutional networks. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR), Honolulu. **Proceedings[...]** Piscataway: IEEE, 2017. p.2261–2269.

IVAKHNENKO, A. G.; LAPA, V. G. **Cybernetic predicting devices**. New York: CCM Information Corporation, 1965.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: a survey. **Computers and Electronics in Agriculture**, Amsterdam, v.147, p.70–90, 2018.

KOUNALAKIS, T.; TRIANTAFYLLIDIS, G. A.; NALPANTIDIS, L. Weed recognition framework for robotic precision farming. *In*: IMAGING SYSTEMS AND TECHNIQUES (IST), 2016 IEEE INTERNATIONAL CONFERENCE ON, 2016, Chania. **Proceedings[...]** Piscataway: IEEE, 2016. p.466–471. Disponível em: https://ieeexplore.ieee.org/document/7738271. Acesso em: 18 mar. 2018.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *In*: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 2012, Stateline. **Proceedings[...]**, 2012. p.1097–1105.

LAMESKI, P. *et al* . Weed Detection Dataset with RGB Images Taken Under Variable Light Conditions. *In*: INTERNATIONAL CONFERENCE ON ICT INNOVATIONS, 2017, Skopje. **Proceedings[...]**, 2017. p.112–119.

LECUN, Y.; BENGIO, Y. *et al* . Convolutional networks for images, speech, and time series. **The Handbook of Brain Theory and Neural Networks**, Cambridge, v.3361, n.10, 1995.

LECUN, Y. *et al* . Backpropagation applied to handwritten zip code recognition. **Neural Computation**, Homdel, v.1, n.4, p.541–551, 1989.

LECUN, Y. *et al* . Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, Pasadena, v.86, n.11, p.2278–2324, 1998.

LIN, M.; CHEN, Q.; YAN, S. Network in network. **Neural and Evolutionary Computing**. Banff, 2013.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2015, Boston. **Proceedings[...]**, 2015. p.3431–3440.

LOTTES, P. *et al* . Fully Convolutional Networks with Sequential Information for Robust Crop and Weed Detection in Precision Farming. **IEEE Robotics and Automation Letters**. IEEE. Piscataway, 2018.

MAJEED, A. Application of Agrochemicals in Agriculture: benefits, risks and responsibility of stakeholders. **Food and Chemical Toxicology**, Amsterdam, v.2, n.1.3, 2018.

MEULEN, A. van der; CHAUHAN, B. S. A review of weed management in wheat using crop competition. **Crop Protection**, Amsterdam, v.95, p.38–44, 2017.

MEYER, G. E.; NETO, J. C. Verification of color vegetation indices for automated crop imaging applications. **Computers and Electronics in Agriculture**, Amsterdam, v.63, n.2, p.282–293, 2008.

MILIOTO, A.; LOTTES, P.; STACHNISS, C. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. *In*: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), **Proceedings[...]**. IEEE. Piscataway, 2018. p.2229–2235.

OKAMOTO, H. *et al* . Plant classification for weed detection using hyperspectral imaging with wavelet analysis. **Weed Biology and Management**, Tokyo, v.7, n.1, p.31–37, 2007.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v.9, n.1, p.62–66, 1979.

PEREZ, A. *et al* . Colour and shape analysis techniques for weed detection in cereal fields. **Computers and Electronics in Agriculture**, Amsterdam, v.25, n.3, p.197–212, 2000.

RAUBER, T. W. Redes neurais artificiais. **Universidade Federal do Espírito Santo**, Vitória, 2005.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **International Conference on Learning Representations**. Columbus, 2014.

SOKOLOVA, M.; LAPALME, G. A Systematic Analysis of Performance Measures for Classification Tasks. **Inf. Process. Manage.**, Tarrytown, v.45, n.4, p.427–437, July 2009.

SZEGEDY, C. *et al* . Going deeper with convolutions. *In*: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, **Proceedings[...]**. IEEE. Boston, 2015. p.1–9.

TANG, J.-L. *et al* . Weed detection using image processing under different illumination for site-specific areas spraying. **Computers and Electronics in Agriculture**, Amsterdam, v.122, p.103–111, 2016.

TANG, J. *et al* . Weed identification based on K-means feature learning combined with convolutional neural network. **Computers and Electronics in Agriculture**, Amsterdam, v.135, p.63 – 70, 2017.

TELLAECHE, A. *et al* . A new vision-based approach to differential spraying in precision agriculture. **Computers and Electronics in Agriculture**, Amsterdam, v.60, n.2, p.144–155, 2008.

THOMPSON, J.; STAFFORD, J.; MILLER, P. Potential for automatic weed detection and selective herbicide application. **Crop Protection**, Londres, v.10, n.4, p.254–259, 1991.

TRUSSELL, H. J.; SABER, E.; VRHEL, M. Color image processing: basics and special issue overview. **IEEE Signal Processing Magazine**, v.22, n.1, 2005.

UTSTUMO, T. *et al* . Robotic in-row weed control in vegetables. **Computers and Electronics in Agriculture**, Amsterdam, v.154, p.36–45, 2018.

VELIKOVI, P. **Deep learning for complete beginners: convolutional neural networks with keras**. 2017. Disponível em: https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html. Acessado em 03/10/2018.

VOULODIMOS, A. *et al* . Deep learning for computer vision: a brief review. **Computational Intelligence and Neuroscience**, Athens, v.2018, 2018.

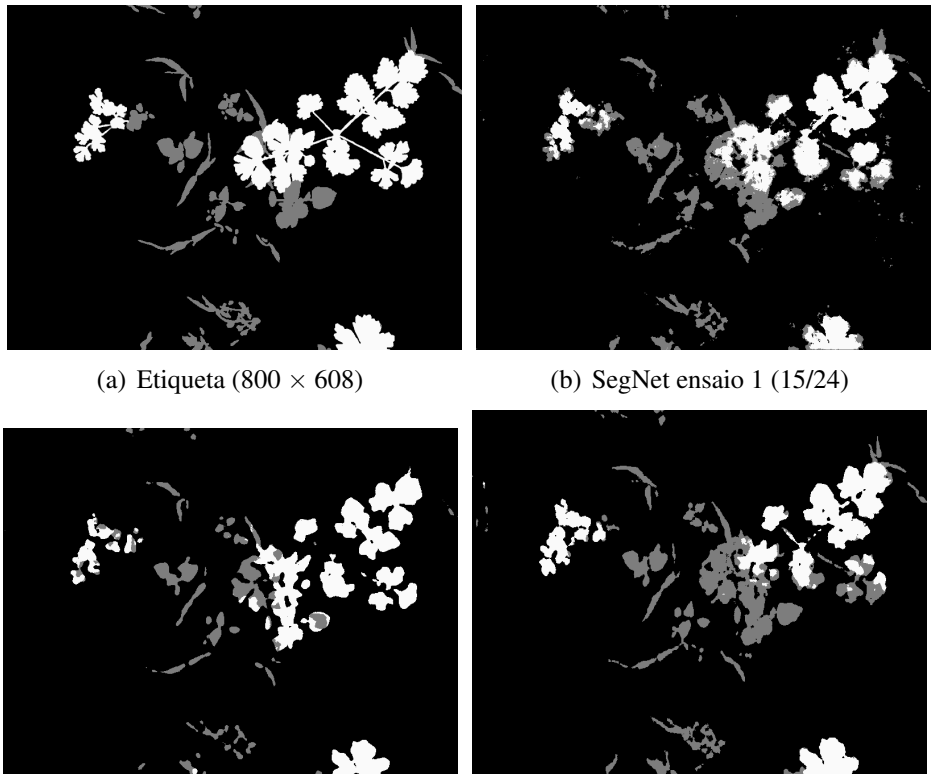
VRINDTS, E.; DE BAERDEMAEKER, J.; RAMON, H. Weed detection using canopy reflection. **Precision Agriculture**, Nova Iorque, v.3, n.1, p.63–80, 2002.

WOEBBECKE, D. M. *et al* . Color indices for weed identification under various soil, residue, and lighting conditions. **Transactions of the ASAE**, St. Joseph, v.38, n.1, p.259–269, 1995.

APÊNDICE A IMAGENS SEGMENTADAS

Neste apêndice são apresentadas as imagens geradas pelas redes que foram treinadas neste trabalho. Elas estão postas em contraste com as etiquetas originais do banco de dados (LAMESKI et al., 2017), conforme apontado nas legendas das figuras. Elas estão na mesma ordem utilizada pelas tabelas 4, 5, 6 e 7.

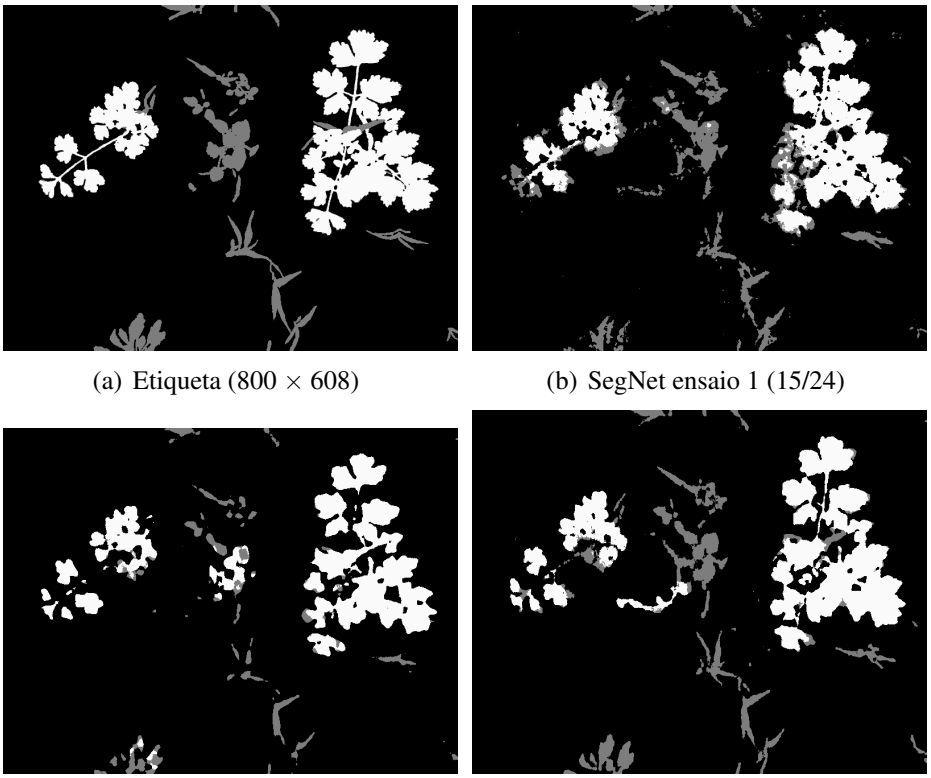
Figura 28: Conjunto de teste (01).



(a) Etiqueta (800 × 608) (b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24) (d) FCN ensaio 2 (15/24)

Figura 29: Conjunto de teste (02).



(a) Etiqueta (800 × 608)

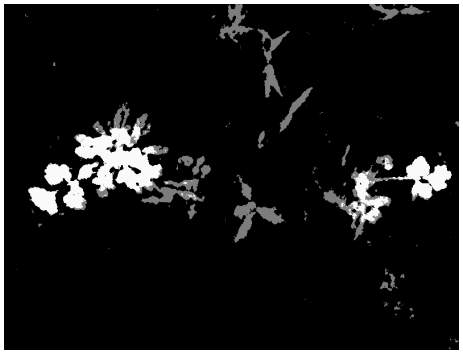
(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

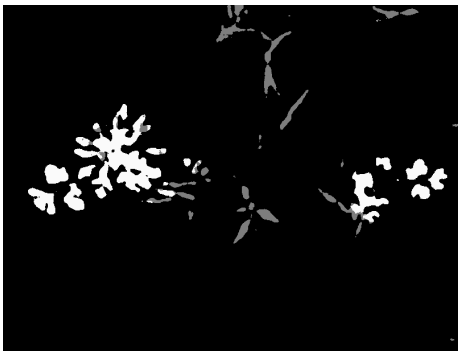
(d) FCN ensaio 2 (15/24)

Figura 30: Conjunto de teste (03).

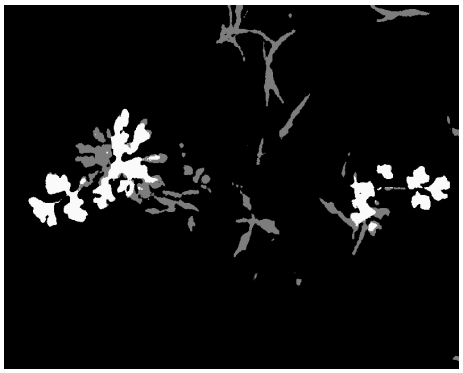
(a) Etiqueta (800 × 608)



(b) SegNet ensaio 1 (15/24)

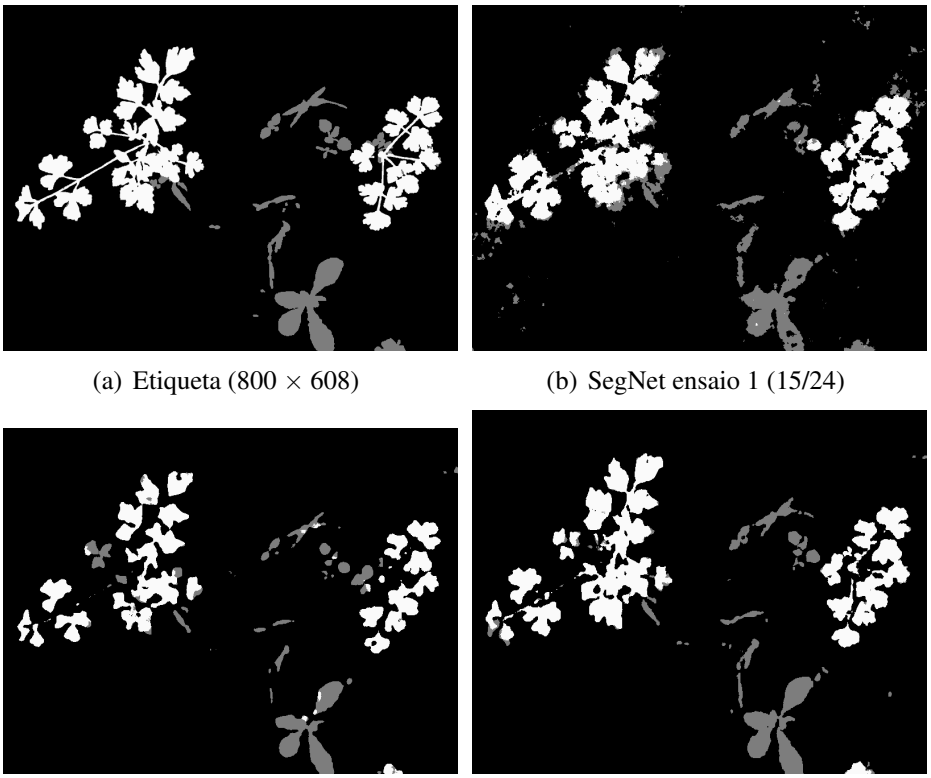


(c) FCN ensaio 1 (15/24)



(d) FCN ensaio 2 (15/24)

Figura 31: Conjunto de teste (04).



(a) Etiqueta (800 × 608)

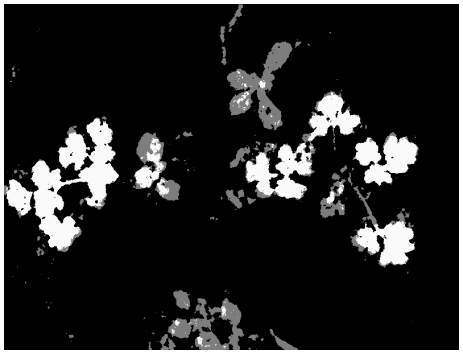
(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 32: Conjunto de teste (05).

(a) Etiqueta (800 × 608)

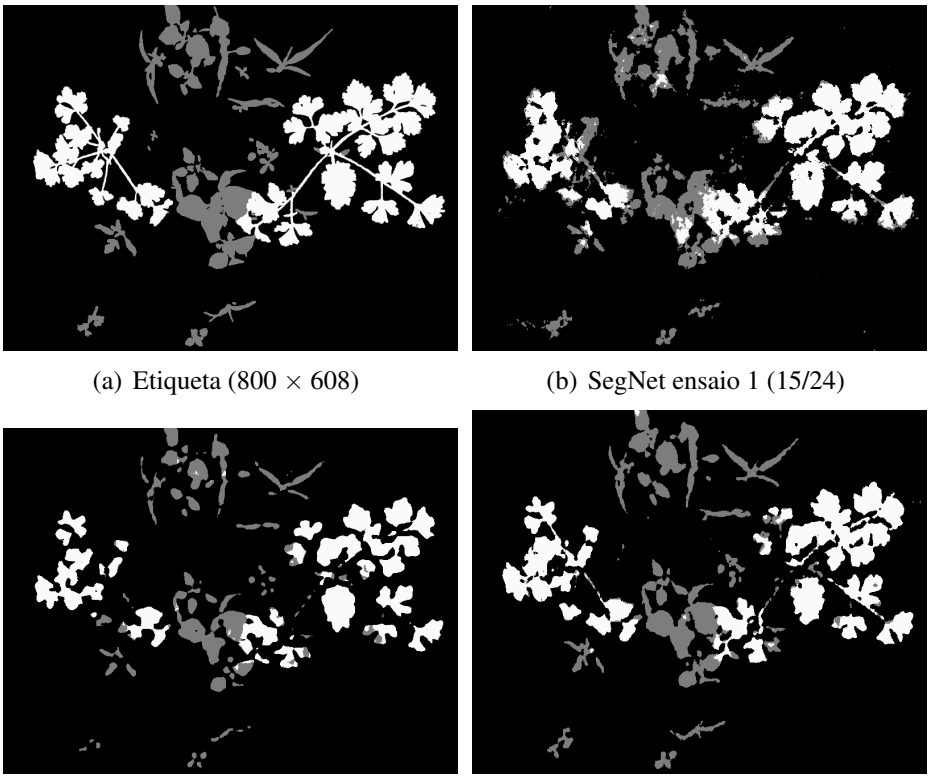


(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 33: Conjunto de teste (06).



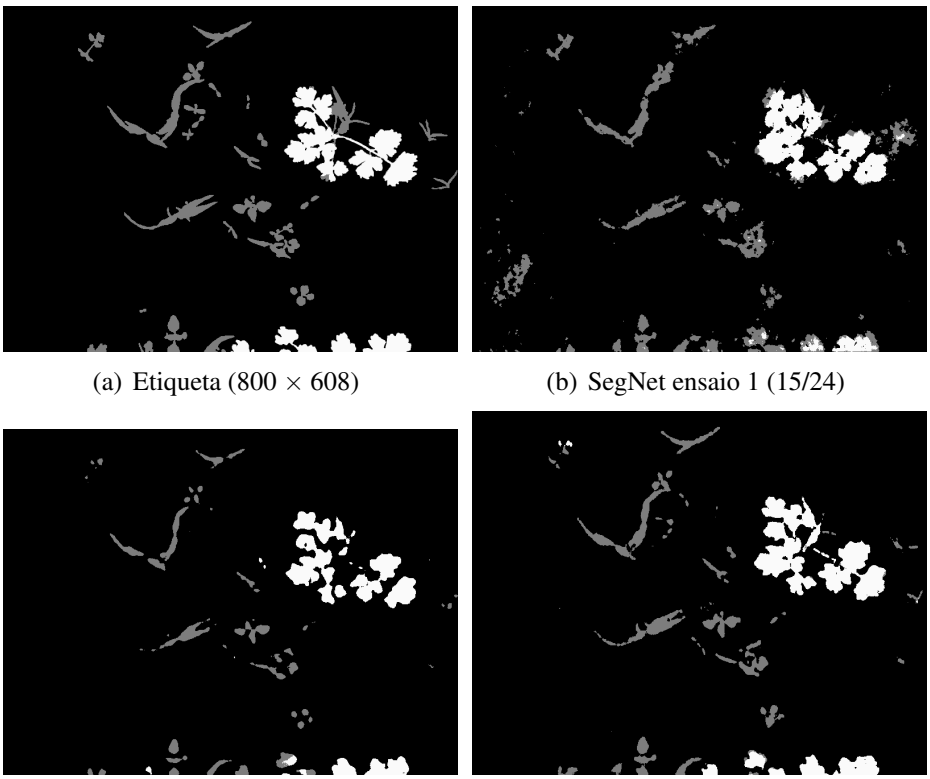
(a) Etiqueta (800 × 608)

(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 34: Conjunto de teste (07).



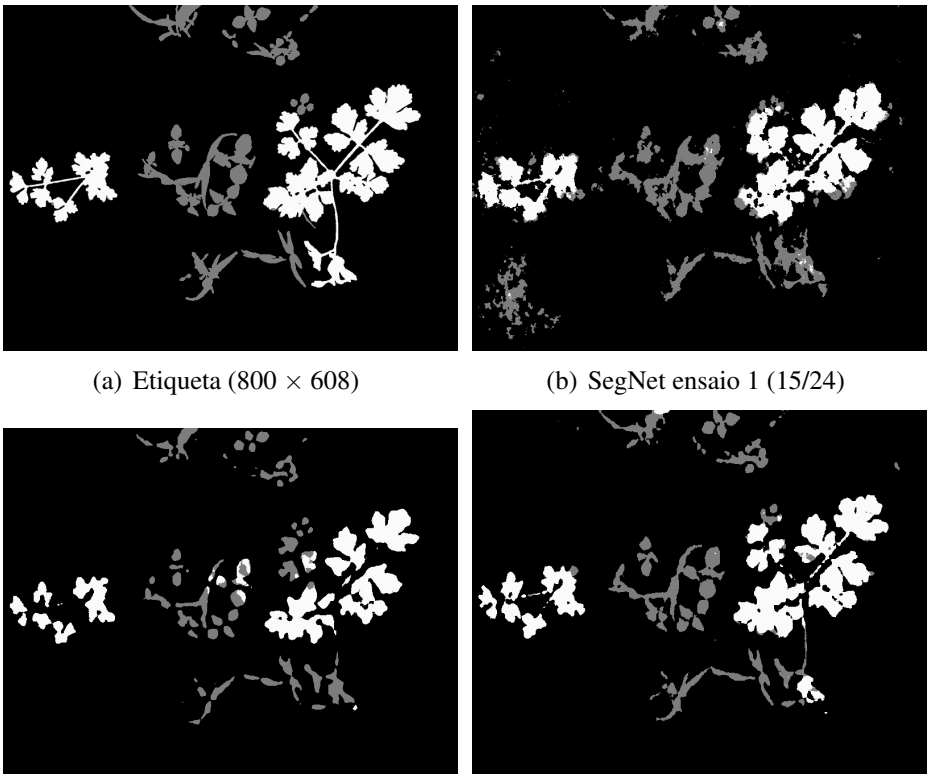
(a) Etiqueta (800 × 608)

(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 35: Conjunto de teste (08).



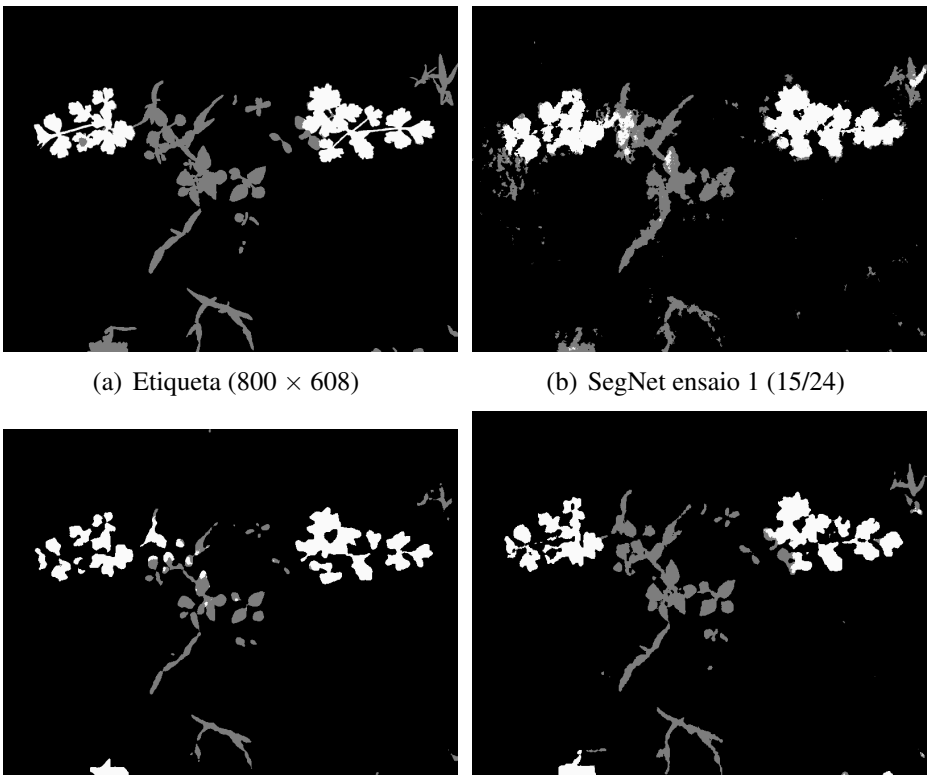
(a) Etiqueta (800 × 608)

(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 36: Conjunto de teste (09).



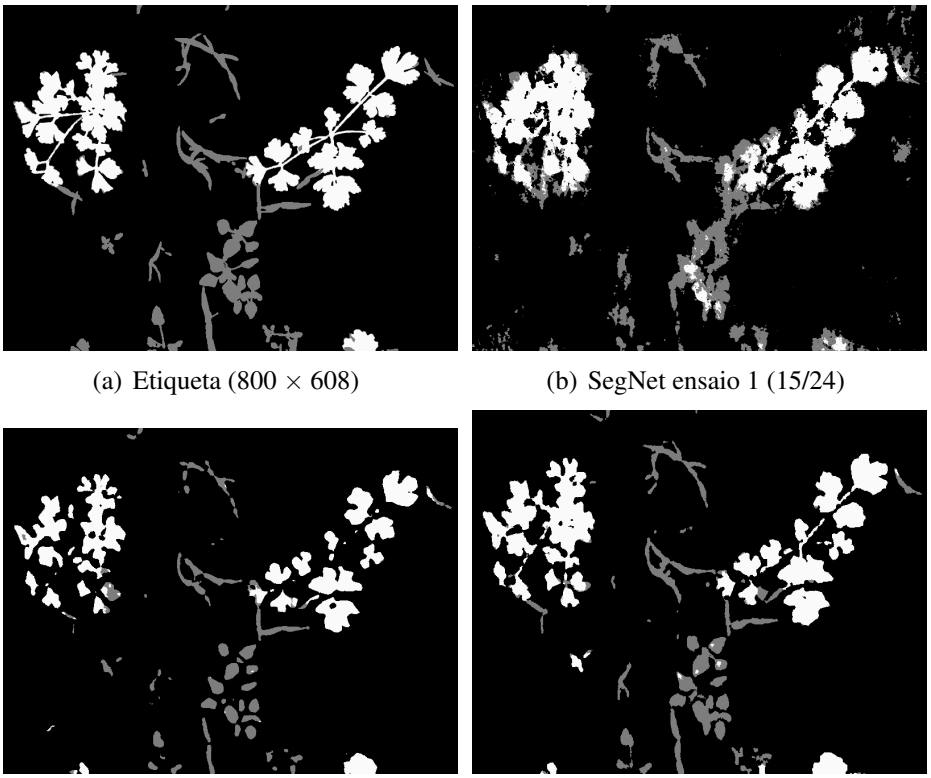
(a) Etiqueta (800 × 608)

(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 37: Conjunto de teste (10).



(a) Etiqueta (800 × 608)

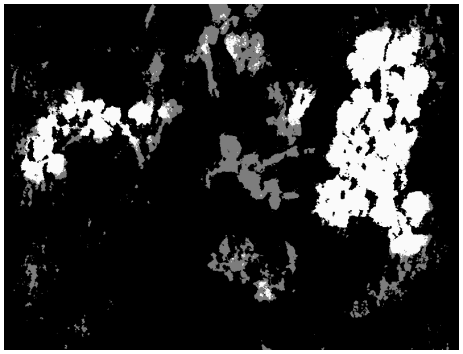
(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 38: Conjunto de teste (11).

(a) Etiqueta (800 × 608)

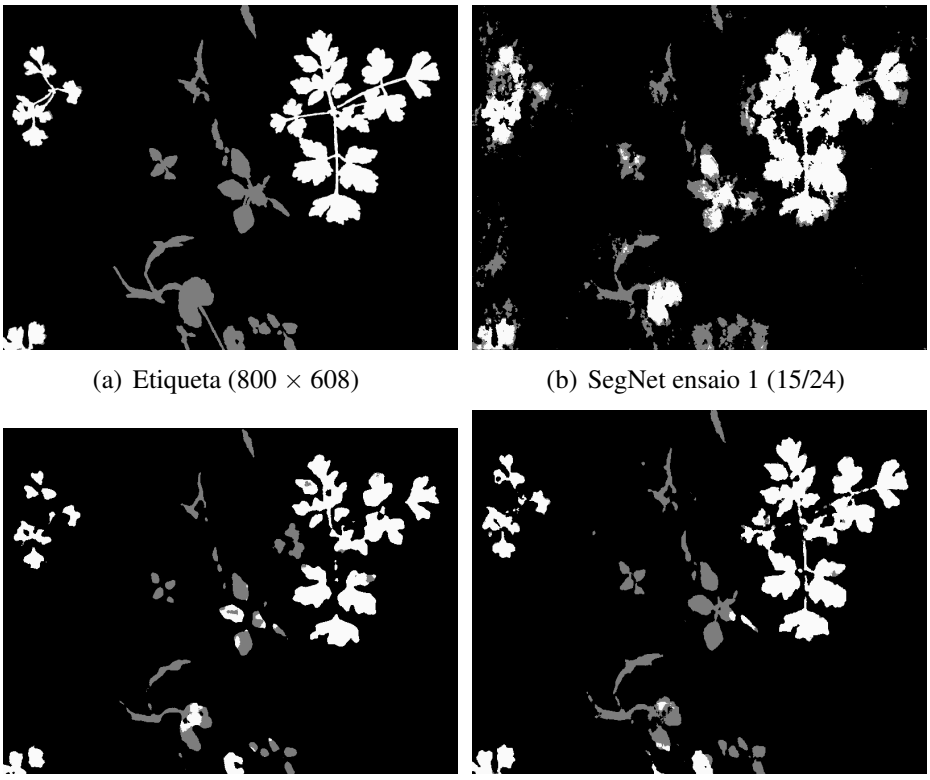


(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 39: Conjunto de teste (12).



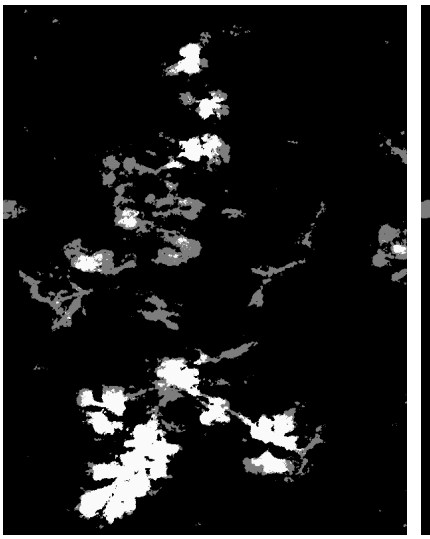
(a) Etiqueta (800 × 608)

(b) SegNet ensaio 1 (15/24)

(c) FCN ensaio 1 (15/24)

(d) FCN ensaio 2 (15/24)

Figura 40: Conjunto de teste (13).



(a) SegNet ensaio 1 (15/24)

(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 41: Conjunto de teste (14).

(a) SegNet ensaio 1 (15/24)

(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 42: Conjunto de teste (15).

(a) SegNet ensaio 1 (15/24)

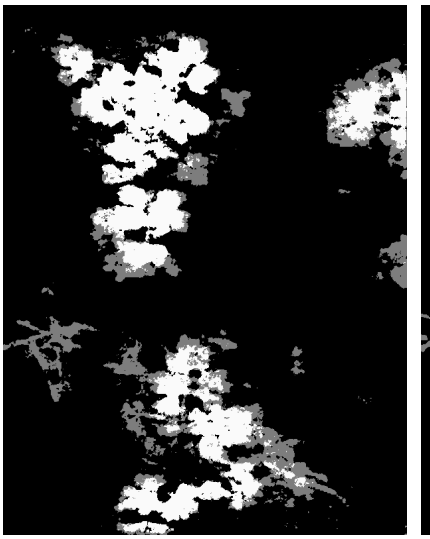
(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 43: Conjunto de teste (16).



(a) SegNet ensaio 1 (15/24)

(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 44: Conjunto de teste (17).

(a) SegNet ensaio 1 (15/24)

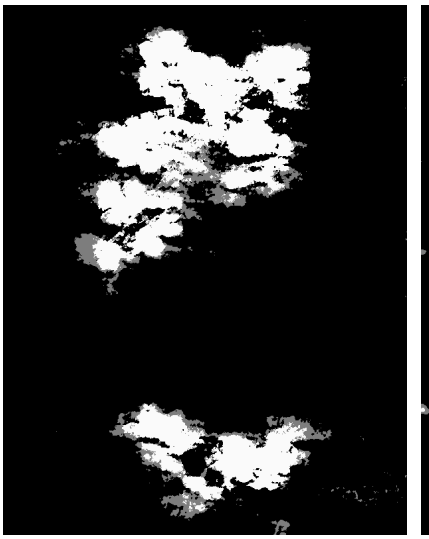
(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 45: Conjunto de teste (18).



(a) SegNet ensaio 1 (15/24)

(b) FCN ensaio 1 (15/24)

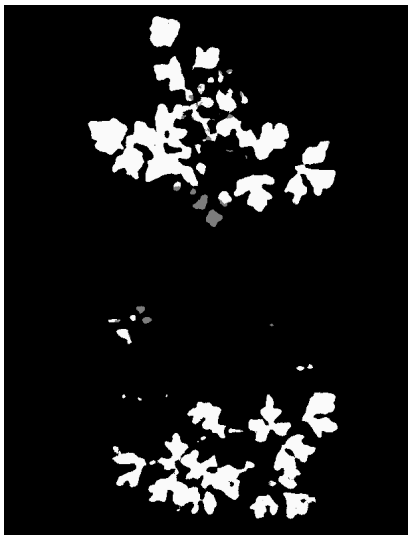
(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 46: Conjunto de teste (19).

(a) SegNet ensaio 1 (15/24)



(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 47: Conjunto de teste (20).

(a) SegNet ensaio 1 (15/24)

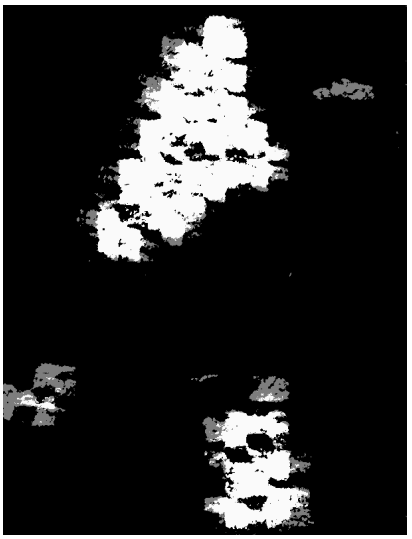
(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

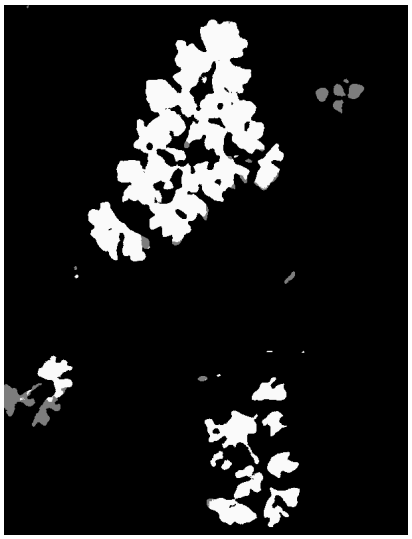
(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 48: Conjunto de teste (21).



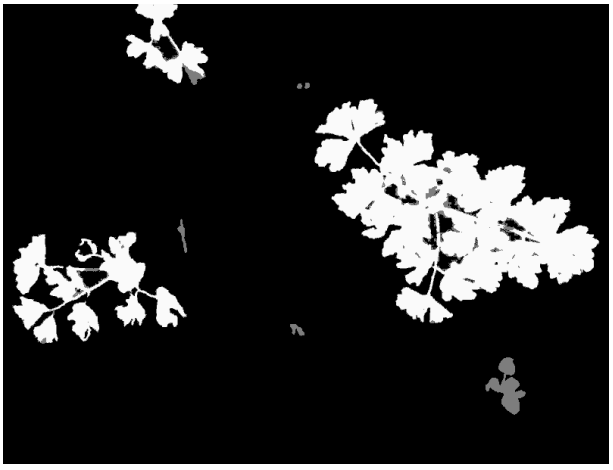
(a) SegNet ensaio 1 (15/24)



(b) FCN ensaio 1 (15/24)

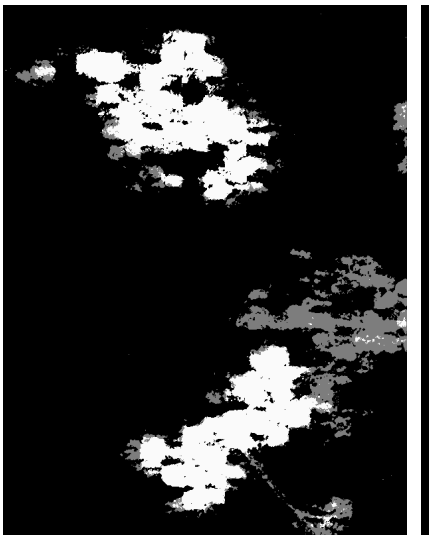
(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)



(e) Etiqueta (800 $\times$ 608)

Figura 49: Conjunto de teste (22).



(a) SegNet ensaio 1 (15/24)

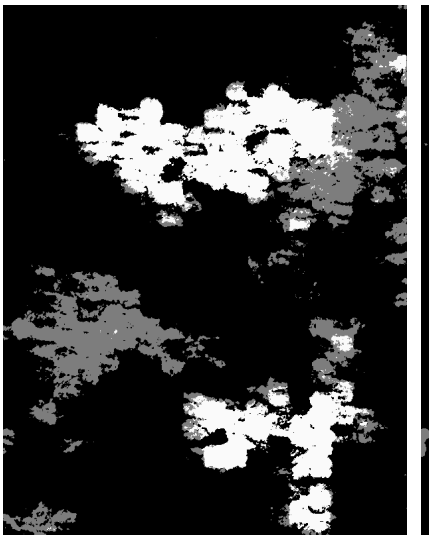
(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

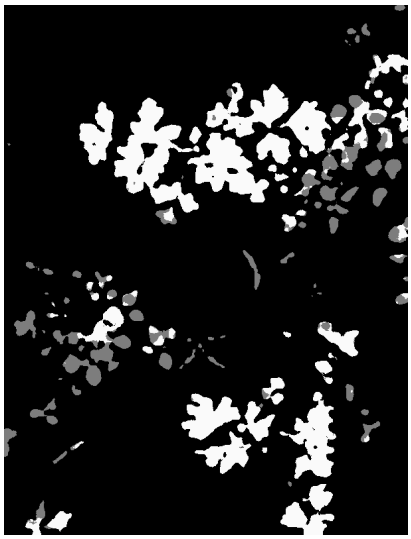
(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 50: Conjunto de teste (23).



(a) SegNet ensaio 1 (15/24)



(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)

Figura 51: Conjunto de teste (24).

(a) SegNet ensaio 1 (15/24)

(b) FCN ensaio 1 (15/24)

(c) FCN ensaio 2 (15/24)

(d) FCN ensaio 3 (27/12)

(e) Etiqueta (800 $\times$ 608)