Gabriel Lucas Silva Machado

# Limiarização em redes neurais convolucionais para classificação em cenário aberto

**Belo Horizonte**

**2018/2**

Gabriel Lucas Silva Machado

# Limiarização em redes neurais convolucionais para classificação em cenário aberto

Relatório final da disciplina Projeto Orientado em Computação II do curso de Bacharelado em Ciência da Computação da UFMG

Univerisdade Federal de Minas Gerais - UFMG

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Orientador: Jefersson Alex dos Santos

Belo Horizonte

2018/2

# Resumo

O crescimento da quantidade de dados gerados é evidente em nossa sociedade, e esses dados variam desde tabelas a até mesmo relatórios e imagens. Devido a essa grande quantidade de dados, o rumo natural e favorável em que a literatura decidiu seguir é o de modelos orientados a dados. Porém ao se tratar de imagens, utilizar modelos orientados a dados pode ser complicado, pois em uma imagem pode existir uma grande quantidade de classes e indentificar todas elas antes de treinar um determinado modelo pode ser uma tarefa inviável. Para contornar o problema descrito, surgiu uma abordagem conhecida como cenário aberto, que tenta abstrair a existência de múltiplas classes em um deteminado problema, tendo foco somente nas classes relevantes, sendo as outras ditas como 'desconhecidas'. Esse trabalho consiste na adaptação de um modelo de aprendizado profundo para limiarização de plantações de café para uma abordagem de cenário semi-aberto e completamente aberto, utilizando como discriminador de classes o hiperparâmetro *threshold*. E para definir esse parâmetro foi proposto um estimador. [1]

Palavras-chave: redes neurais convolucionais, cenário aberto, imagens, limiarização.

[1] Código disponível para *download* no repositório: <https://github.com/Gabriellm2003/Coffee-CNN>

# Lista de ilustrações

# Lista de tabelas

# Agradecimentos

Os agradecimentos principais são direcionados aos alunos Caio Cesar Viana da Silva e Keiller Nogueira, que juntamente ao professor Jefersson Alex dos Santos me orientaram na elaboração desse projeto.

# Sumário

# 1 Introdução

Os constantes avanços tecnológicos que ocorreram nos últimos anos proporcionaram ferramentas capazes de captar imagens de alta resolução da superfície terrestre através do uso de sensores (CAMPBELL; WYNNE, 2011), além de tecnologias capazes de processar essas imagens. O estudo de como coletar, armazenar e analisar essas imagens ficou conhecido como sensoriamento remoto.

A quantidade de informação que imagens de sensoriamento remoto carregam é imensa, e por causa disso, estas vem sendo utilizadas para diversos tipos de aplicações, tais como planejamento urbano (CHEN et al., 2006), agricultura (NOGUEIRA et al., 2016), prevenção de desastres naturais (NOGUEIRA et al., 2018), entre outras. Por causa dessa importância, surgiram diversas técnicas com objetivo de analisar, entender, ou até mesmo extrair informações a partir dessas imagens. Nesse trabalho será usado uma dessas técnicas, sendo esta conhecida como redes neurais convolucionais (*CNN*).

As redes neurais convolucionais consistem em tipos especializados de redes neurais, caracterizadas por conterem camadas de convoluções. Esse modelo de arquitetura de rede é tipicamente utilizado para reconhecimento de padrões e processamento em imagens. A popularidade desse tipo de arquitetura para problemas que envolvem imagens, ocorre devido a artefatos contidos nesse tipo de rede , tais como as camadas de convolução, RELU e *pooling*. (GOODFELLOW; BENGIO; COURVILLE, 2016)

Os *datasets* de imagens utilizados para treinamento de algoritmos geralmente contêm anotações sobre o que cada elemento representa na imagem, e isso é denominado como a classe do elemento. Diferentemente de problemas *closed set*, nos quais todas as classes são conhecidas durante a fase de treinamento, problemas que utilizam imagens como *inputs*, muitas vezes são classificados como *open set*. Nessa abordagem, durante o treinamento são apresentadas informações incompletas das classes presentes no problema, e classes desconhecidas podem ser submetidas durante a fase de testes. (SCHEIRER et al., 2013) No caso desse trabalho, será utilizada uma abordagem simplificada do *open set*, em que todas as classes não relevantes ao problema serão colocadas em um único grupo.

As saídas geradas por redes neurais convolucionais(LECUN et al., 1998), tipicamente passam por um algoritmo classificador. No caso desse trabalho, será utilizado um algoritmo baseado em uma regressão logística, que antes de retornar a saída passa por uma função conhecida como *softmax*. Essa função retorna a probabilidade que uma dada entrada pertença a cada classe definida. Para aplicá-la em um problema de classificação binária (classe relevante e não relevante) é necessário definir um parâmetro conhecido como *threshold*, que representa a probabilidade que delimita a distinção dessas duas classes.

Tendo isso em mente, o objetivo desse trabalho consiste em avaliar experimentalmente alguma forma para estimação de *thresholds* que otimizem as classificações feitas por uma *CNN open set*. Para essa tarefa, será utilizada uma rede neural convolucional com arquitetura definida em (NOGUEIRA et al., 2016), aplicada ao *dataset* de imagens de sensoriamento remoto *AGRICULTURE* (PENATTI; NOGUEIRA; SANTOS, 2015) [1], com o propósito de segmentar as regiões da imagem que contêm plantações de café.

[1] *Dataset* disponível para *download* em: <http://www.patreo.dcc.ufmg.br/2017/11/12/brazilian-coffee-scenes-dataset>

# 2 Referencial Teórico

Dois anos atrás, o trabalho conhecido como *"Learning to Semantically Segment High-Resolution Remote Sensing Images"* (NOGUEIRA et al., 2016) definiu duas arquiteturas para redes neurais convolucionais para realizar segmentação semântica em imagens de sensoriamento remoto, utilizando uma abordagem *pixelwise*, isto é, cada *pixel* da imagem é tratado individulamente. Esse artigo é importante para esse trabalho, pois ele descreve arquiteturas de *CNNs* e técnicas fundamentais para o desenvolvimento desse projeto.

Também na mesma época, foi publicado o *paper "Towards Open Set Deep Networks"* (BENDALE; BOULT, 2015), que apresenta uma metodologia para adaptar redes neurais profundas à abordagem *open set*. Apesar da metodologia descrita no *paper* não ser utilizada nesse trabalho, ele é importante pois indtroduz diversos conceitos da abordagem *open set*, que foi utilizada neste trabalho.

Em 2014, Girshick (GIRSHICK et al., 2014) propôs um método (R-CNN) para detecção e segmentação de objetos. O funcionamento da R-CNN baseia-se na seleção e extração de *features* e regiões, utilizando redes neurais convolucionais pré-treinadas. Esse *paper* é relevante para esse trabalho, pois trata-se uma rede neural para classificação baseada em regiões.

Em relação à implementação desse trabalho, foi utilizado o *framework TensorFlow* (TENSORFLOW, 2018). Os motivos da escolha se devem principalmente pela excelente documentação e suporte da comunidade. Como guia de treinamento para a rede neural foram utilizados os anais de conferência (ORR; MüLLER, 1998), que contêm várias dicas de como descobrir bons parâmetros para redes neurais, e como melhorar resultados.

Por fim, foram utilizados os livros (GONZALEZ; WOODS, 2008) e (GOODFELLOW; BENGIO; COURVILLE, 2016), com o propósito de esclarecer conceitos das áreas de processamento digital de imagens e aprendizado de máquina recorrentes no trabalho.

# 3 Visão Geral

## 3.1 O *dataset*

O *dataset* utilizado é composto de 5 imagens retiradas sobre o município Monte Santos (MG) por um satélite SPOT em 2005. Cada uma das imagens contém 500x500 *pixels*, dos quais 51% representam áreas com plantações de café e os 49% restantes representam áreas com outros tipos de vegetações. (NOGUEIRA et al., 2016) O desafio do *dataset* é gerar a limiarização das plantações de café contindas nas imagens, a Figura 2 ilustra um exemplo de imagem e sua limiarização esperada, onde os *pixels* brancos representam as áreas de plantações de café e os pretos todo o restante.[1]

Figura 1: Exemplo de imagem do *dataset*

## 3.2 A abordagem *pixelwise*

Assim como foi proposto no artigo (NOGUEIRA et al., 2016), nesse trabalho foi utilizado uma abordagem *pixelwise* com o uso de janelas de contexto para o problema referente ao *dataset* *AGRICULTURE*(PENATTI; NOGUEIRA; SANTOS, 2015).

A abordagem *pixelwise* consiste no tratamento individual de todos os pixels presentes nas imagens. Já as janelas de contexto consistem em *crops* 25x25 com o *pixel* central representando a classe referente ao *crop*. Dessa forma, foram criadas janelas de contexto para todos os *pixels* da imagem, e cada uma dessas janelas é usada como entrada da rede. Assim, a rede neural proposta é capaz de classificar individualmente todos os *pixels* da imagem, levando em consideração os vizinhos mais próximos, e através disso, é possível obter limiarizações das regiões desejadas.

[1] *Dataset* disponível para *download* em: <http://www.patreo.dcc.ufmg.br/2017/11/12/brazilian-coffee-scenes-dataset>

Figura 2: Exemplos de janelas de contexto feitas a partir do *dataset*

## 3.3 Arquitetura da rede

A Figura 3 ilustra a arquitetura da rede neural convolucional utilizada neste trabalho. Essa rede recebe como entrada uma imagem 25x25. Essa imagem passa por 3 camadas de convoluções, 3 camadas *max pooling*, 2 camadas *fully conected*, para então chegar a camada de saída, que é responsável por classificar o *pixel* central da imagem de entrada. Diferentemente das redes neurais tradicionais, as CNNs possuem algumas peculiaridades. O objetivo e funcionamento dessas peculiaridades será explicado logo abaixo.

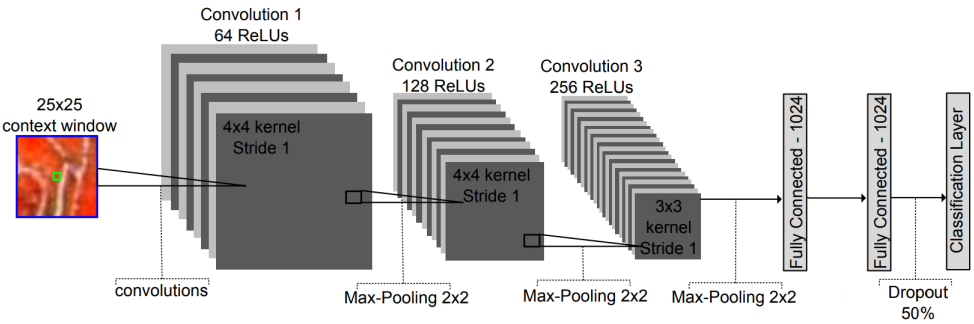


Figura 3: Arquitetura da *CNN* utilizada no trabalho. Retirada de (NOGUEIRA et al., 2016)

O objetivo principal das camadas de convolução é a extração de *features* relevantes contidos na imagem de entrada. Essa extração é feita aplicando-se filtros específicos, atualizados durante o treinamento, sobre pequenos *crops* dessas imagens. No caso da primeira camada de convolução (contida na Figura 3) são aplicados 64 filtros diferentes com dimensão 4x4 na imagem recebida como entrada com um *stride* de 1.

Imediatamente após as convoluções são aplicada funções de ativação (ReLuS) sobre as saídas geradas pela convolução. O objetivo dessa aplicação é introduzir não linearidade aos dados, visto que as convoluções aplicam operações exclusivamente lineares sobre eles. A introdução da não linearidade feita pelas ReLuS ocorre de acordo com a equação 3.1. (DESHPANDE, 2016)

$$ReLu(x) = \max(0, x) \tag{3.1}$$

Por fim, após a aplicação das ReLuS, os dados resultantes passam por uma camada de *pooling*. No caso da rede deste trabalho foi utilizado a *max pooling*. O objetivo principal das camadas de *pooling* é realizar uma redução espacial na matriz resultante. Na rede utilizada neste trabalho foram feitos 3 *downsamplings* utilizando 3 camadas de *max pooling* com filtros 2x2 e *stride* 2, da forma mostrada na Figura 4. (DESHPANDE, 2016)

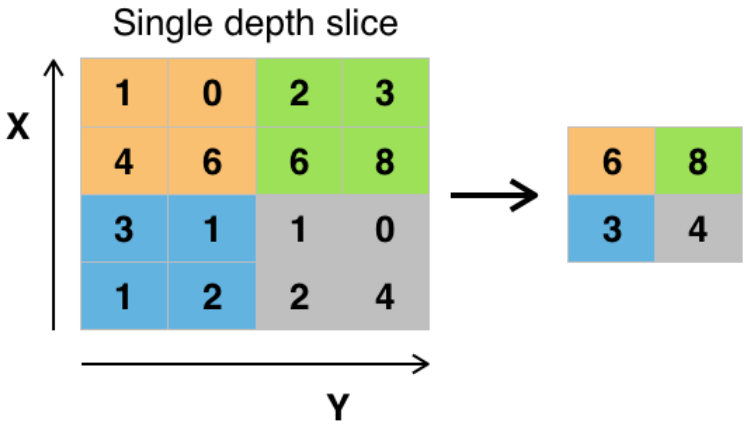


Figura 4: Exemplo de uma aplicação de *max pooling* com *stride*=2, utilizando um filtro 2x2.

## 3.4 Protocolo para treinamento e validação

Foi utilizado o protocolo *K-Fold Cross Validation* (BRONSHTEIN, 2017), com k = 5, para medir a capacidade de generalização do modelo. Como o *dataset* só possui 5 imagens, a divisão feita em cada *fold* utilizou 4 imagens para treinamento (80%) e 1 imagem para validação(20%). Ao utilizar 5 *folds*, foi possível avaliar o modelo com todas as combinações de sub-grupos existentes para a divisão 80%/20%.

## 3.5 O estimador

O estimador implementado utiliza o paradigma tentativa e erro em intervalos de números discretos. A primeira iteração do algoritmo é feita no intervalo pré-definido [ 0.01, 0.99 ] com saltos de 0.01. Para cada número dessa sequência, este é avaliado no modelo utilizado como *threshold*, e as estatísticas referentes a cada número testado na rede são salvas.

A partir da segunda iteração do algoritmo, o intervalo é definido pelas equações 3.2 e 3.3, onde o termo $melhorThreshold_{i-1}$ corresponde ao delimitador encontrado na

iteração anterior com maior acurácia. É recomendável rodar o algoritmo até que a variância das acurácias do intervalo testado esteja próxima de 0, isto é, a variação dos *thresholds* avaliados causou pouco impacto na acurácia.

$$intervalo_i = [\ melhorThreshold_{i-1} - 5 \times 10^{-2i+1},\ melhorThreshold_{i-1} + 5 \times 10^{-2i+1}\ ] \tag{3.2}$$

$$salto_i = 10^{-2i} \tag{3.3}$$

# 4 Resultados

O treinamento da rede foi feito utilizando uma máquina com as especificações descritas na tabela 1, utilizando os parâmetros descritos tabela 2 na *CNN* (figura 3). Como foi utilizado o protocolo *K-Fold Cross Validation*, foram treinados 5 modelos diferentes de redes, alternando os subconjuntos de imagens para treinamento e para validação. Foram feitas duas abordagens para o problema, que serão explicadas posteriomente neste documento.

Tabela 1: Especificações técnicas da máquina utilizada.

| | **Hardware** |
|---|---|
| **Placa Mãe** | ASUSTeK ROG RAMPAGE VI EXTREME |
| **CPU** | Intel(R) Core(TM) i9-7920X CPU @ 2.90GHz |
| **GPU** | GeForce GTX TITAN Xp 12Gb |
| **Memória** | 64Gb DDR4 Kingston HyperX Fury |

Tabela 2: Parâmetros utilizados na *CNN*.

| | |
|---|---|
| **Tamanho do *batch*** | 128 |
| **Decaimento de pesos** | 0.0005 |
| **Taxa de aprendizado inicial** | 0.01 |
| **Número de iterações** | 200000 |
| **Função de perda** | Média da entropia cruzada |

## 4.1 O cenário semi-aberto

Nessa abordagem, durante a fase de treino foram apresentadas duas classes para a rede neural. Esses subconjuntos são as plantações de café e o que não é plantação de café (o complemento do primeiro subconjunto).

A tabela 3 sumariza os resultados obtidos. Assim como foi descrito, o simulador foi rodado em duas etapas, sendo que na primeira foram testados *thresholds* no intervalo [0.01, 0.99], e na segunda etapa foram explorados intervalos específicos para cada *fold*, de forma a utilizar o melhor *threshold* encontrado na primeira iteração como referência.

Os Gráficos 5 e 6 sumarizam os resultados obtidos na primeira iteração do estimador. Através do Gráfico 6 é possível perceber a variância do hiperparâmetro em cada *fold*, sendo que para o terceiro *fold* essa variância foi a maior, e consequentemente o estimador obteve a maior melhoria de resultados nele. Já em relação à segunda iteração do estimador, o

Gráfico 7 demonstra o comportamento da variância para cada *fold*, e nele já é possível perceber que os valores estão mais estáveis, logo os valores testados já se aproximam do ótimo.

Em relação aos resultados finais, a tabela 3 sumariza os resultados obtidos, e nela é possível notar que houve melhoria para todos os casos. A Figura 8 mostra a evolução das predições para o resultado em que houve a maior melhoria (*fold* 3).

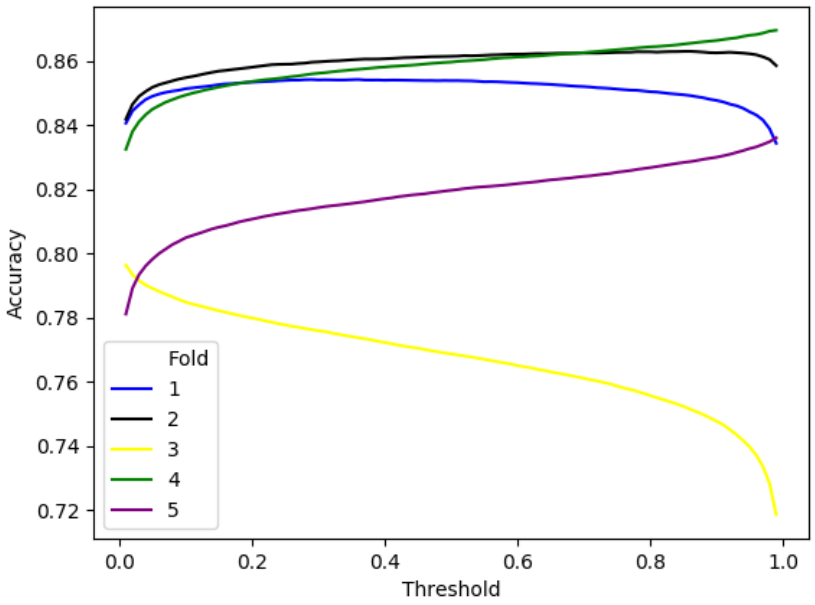


Figura 5: Gráfico de linhas com os resultados do primeiro *round* no estimador força bruta

Tabela 3: Tabela de resultados obtidos antes e depois de executar o estimador.

| | **Sem o estimador** | ***Round* 1** | ***Round* 2** |
|---|---|---|---|
| ***Fold* 1** | 0.853800 | 0.854164 | 0.854176 |
| ***Fold* 2** | 0.861376 | 0.862928 | 0.862964 |
| ***Fold* 3** | 0.768684 | 0.796316 | 0.800252 |
| ***Fold* 4** | 0.859576 | 0.869496 | 0.869568 |
| ***Fold* 5** | 0.819636 | 0.836012 | 0.837164 |
| **Média** | 0.8326144 | 0.8437832 | 0.8448248 |

## 4.2 O cenário aberto

Para essa abordagem, durante a fase de treinamento só foi apresentado à rede uma classe, que no contexto era a classe correspondente às plantações de café. Porém, as tentativas de usar essa abordagem não foram bem sucedidas, visto que durante o treino a

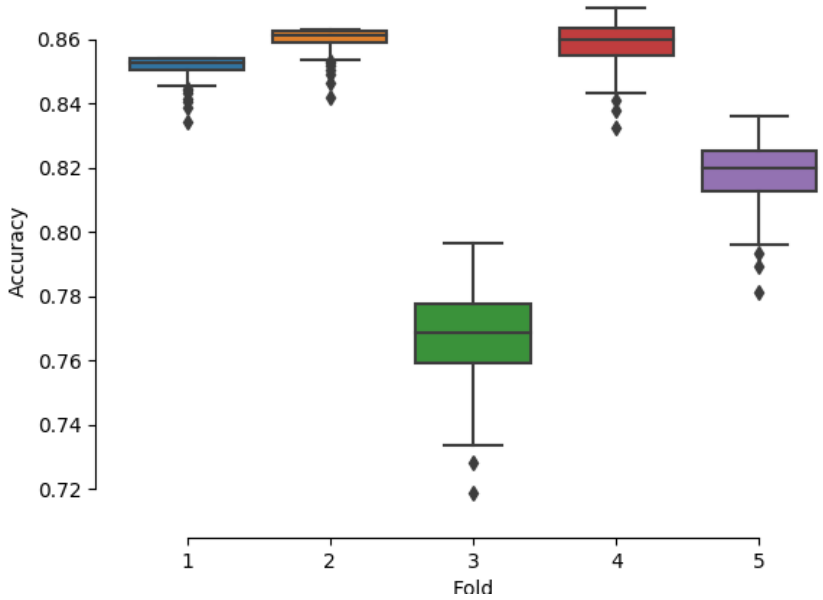


Figura 6: *Box plot* com os resultados do primeiro *round* no estimador força bruta

rede neural acabou aprendendo o caminho óbvio que era classificar qualquer *input* como café, com 100% de certeza, dessa forma a acurácia na fase de treinamento era de 100% e com uma função de perda baixa.

Acredita-se que para contornar o problema é necessário apresentar pelo menos mais uma classe durante a fase de treinamento. Estudos feitos da abordagem de cenário aberto demonstram que é uma técnica geralmente utilizada para problemas multi-classe (BENDALE; BOULT, 2015) (GE et al., 2017). Apesar disso, ainda existe uma alternativa que utiliza somente uma classe (*one-class classification*), mas para isso é necessário utilizar um *dataset* auxiliar para calcular a função de perda. (PERERA; PATEL, 2018)

Em relação à solução que envolve apresentar mais uma classe no treinamento, não foi possível utilizá-la, pois o *dataset* utilizado nesse trabalho não contém informações sobre a existência de outras classes. Já para a solução do *one-class*, esta não foi implementada, pois não encontrei um *dataset* razoável para utilizar como auxiliar. Logo, por essas razões, não foi possível obter resultados relevantes para essa abordagem.

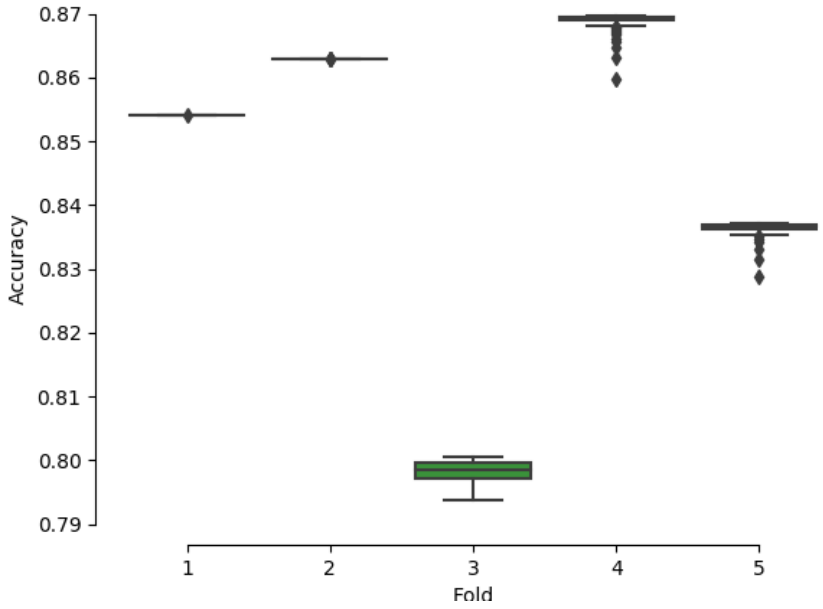


Figura 7: *Box plot* com os resultados do segundo *round* no estimador força bruta

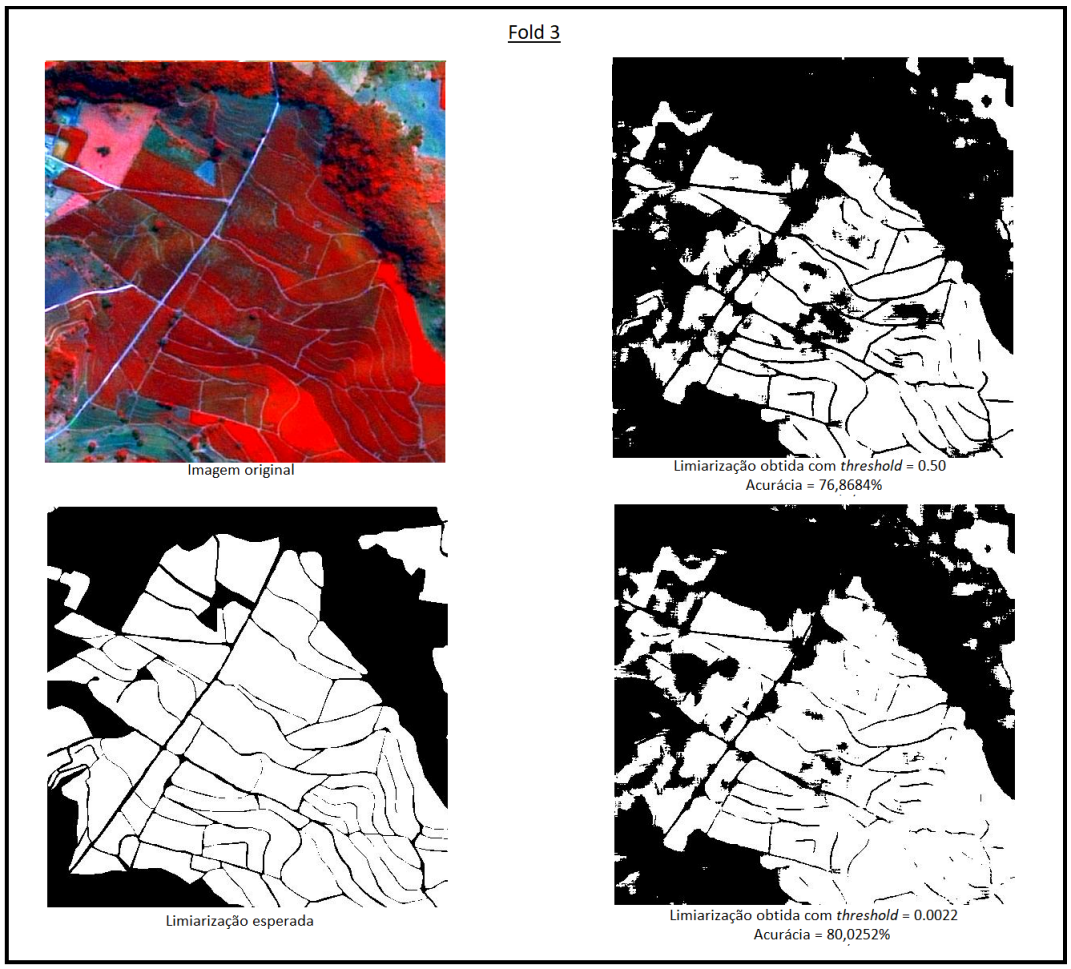


Figura 8: Resultado obtido antes e depois de estimar o *threshold*.

# 5 Conclusão

Esse documento apresentou uma abordagem para cenário aberto em um problema de limiarização de plantações de café utilizando imagens de sensoriamento remoto. Além disso, foi apresentada uma proposta de estimador do hiperparâmetro *threshold*, que discrimina as diferentes classes no classificador.

Em relação ao estimador do *threshold*, este demonstrou uma melhoria significativa dos resultados na abordagem para um cenário semi-aberto, atingindo uma média de melhoria da acurácia acima de 1%, atingindo uma média de acurácia de 83,26%.

Por fim, para os resultados utilizando a abordagem do cenário completamente aberto (*openset*), acredita-se que seja possível explorá-la em um futuro trabalho, de forma semelhante à feita em (PERERA; PATEL, 2018), ou inserindo uma nova classe no *dataset*. Dessas formas, a capacidade de generalização da rede neural treinada não deverá ser prejudicada, e talvez seja possível atingir bons resultados.

# Referências


BENDALE, A.; BOULT, T. E. Towards open set deep networks. *CoRR*, abs/1511.06233, 2015. Disponível em: <http://arxiv.org/abs/1511.06233>. Citado 2 vezes nas páginas 9 e 16.

BRONSHTEIN, A. *Data Science*. 2017. Disponível em: <https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6>. Citado na página 12.

CAMPBELL, J.; WYNNE, R. *Introduction to Remote Sensing*. Guilford Publications, 2011. ISBN 9781609181765. Disponível em: <https://books.google.com.br/books?id=zgQDZEya6foC>. Citado na página 7.

CHEN, X.-L. et al. Remote sensing image-based analysis of the relationship between urban heat island and land use/cover changes. *Remote Sensing of Environment*, v. 104, n. 2, p. 133 – 146, 2006. ISSN 0034-4257. Thermal Remote Sensing of Urban Areas. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0034425706001787>. Citado na página 7.

DESHPANDE, A. *A Beginner's Guide To Understanding Convolutional Neural Networks Part 2*. 2016. Disponível em: <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/>. Citado 2 vezes nas páginas 11 e 12.

GE, Z. et al. Generative openmax for multi-class open set classification. *arXiv preprint arXiv:1707.07418*, 2017. Citado na página 16.

GIRSHICK, R. B. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, p. 580–587, 2014. Citado na página 9.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Disponível em: <http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>. Citado na página 9.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 2 vezes nas páginas 7 e 9.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 7.

NOGUEIRA, K. et al. Exploiting convnet diversity for flooding identification. *IEEE Geoscience and Remote Sensing Letters*, v. 15, n. 9, p. 1446–1450, Sept 2018. ISSN 1545-598X. Citado na página 7.

NOGUEIRA, K. et al. Learning to semantically segment high-resolution remote sensing images. In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2016. p. 3566–3571. Citado 6 vezes nas páginas 3, 7, 8, 9, 10 e 11.

ORR, G. B.; MüLLER, K.-R. (Ed.). *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. London, UK, UK: Springer-Verlag, 1998. ISBN 3-540-65311-2. Citado na página 9.

PENATTI, O. A. B.; NOGUEIRA, K.; SANTOS, J. A. dos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2015. p. 44–51. ISSN 2160-7516. Citado 2 vezes nas páginas 8 e 10.

PERERA, P.; PATEL, V. M. Learning deep features for one-class classification. *arXiv preprint arXiv:1801.05365*, 2018. Citado 2 vezes nas páginas 16 e 18.

SCHEIRER, W. J. et al. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 7, p. 1757–1772, July 2013. ISSN 0162-8828. Citado na página 7.

TENSORFLOW. 2018. Disponível em: <https://www.tensorflow.org/>. Citado na página 9.